Effects of Image Filters on Various Image Datasets

Didem Abidin
Manisa Celal Bayar University
Faculty of Engineering, Computer Engineering Department
Muradiye Campus, 45140, Yunusemre, Manisa, Turkey
+902362012110
didem.abidin@cbu.edu.tr

ABSTRACT
Image classification is a very common research area, on which researchers work with various classification techniques. The aim of this study is to apply different filters on four different datasets and evaluate their performances in image classification. The study was performed in WEKA environment with Random Forest algorithm and image filters are applied to the datasets one by one and as a combination. Filter combinations got better performance than applying single filter on data. Filter combinations got the worst result on artworks with a percentage of 83.42%. However they were very successful on classifying the images in natural images dataset with a performance of 99.76%.

CCS Concepts
• Computing methodologies ~ Supervised learning by classification.

Keywords
Image filters; WEKA; machine learning; image classification.

1. INTRODUCTION
Machine learning (ML) is the study of extracting information from a pile of data. With machine learning techniques, it is possible to learn from raw data the information which does not seem to be there at all. The data to be mined can be textual, numeric or image. ML algorithms look for patterns within data by training a certain percentage of it [1]. Classifying is one of the main goals of data mining and ML algorithms mainly serve as good classifiers on data. They make the classification based on the trained data before.

Images can also be mined as textual and numeric data and these type of data need to be preprocessed before applying any classification techniques.

Image classification is a very common research area, on which researchers work with various classification techniques. Image classification refers to the labelling of images into one of a number of predefined categories [2]. Image classification can be applied to areas like face recognition [3], [4], image and face recognition on social networks [5], handwriting detection [6] and identifying visual brands and logos [7].

ICCTA 2019, April 16–17, 2019, Istanbul, Turkey

ACM ISBN 978-1-4503-7181-0/19/04…$15.00
https://doi.org/10.1145/3323933.3324056

Image classification can be considered as a study of content based image retrieval (CBIR), which is indeed a problem of searching images in a large dataset [8]. The images in a given dataset belong to certain classes and the classification algorithms try to detect the classes of the images correctly. Many different techniques are applied on image datasets for classification like support vector machines (SVM) [9], decision trees (DT) [10], or artificial neural networks (ANN) [11].

As one of the necessary steps of image classification, images must be preprocessed before the classification algorithms are executed. This preprocess step contains filtering for images. Several image filters can be applied on images, either one filter at a time or more than one filters as a combination. The applied filters add new numeric data to each instance, which will later (on classification phase) help the classification algorithms to construct the data model more accurately.

The aim of this study is to apply different filters on four different datasets and evaluate their performances when run with the same classification algorithm. These datasets contain images from different domains. In some of the images, colors of the images are important, where in some, shapes are the discriminating features. For this reason, applying filters with different properties affects the performance results for image classification.

The study was performed in WEKA environment [12] which is a tool for data mining tasks. It contains many machine learning algorithms for classification and accepts data in various formats like .arff and .csv. In this study, the datasets are in .arff files which is explained in the following section.

The layout of the paper is as follows: Section 2 explains the datasets used in the study with preparation and preprocessing phases. In Section 3, the results with different image filters are presented. Section 4 discusses about the obtained results and gives some hints about future work.

2. MATERIAL AND METHOD
2.1 Datasets
The datasets chosen have similar numbers of classes and the instance amounts are close to each other. The datasets are obtained from Kaggle [13], which is an online platform with datasets and some competitions based on machine learning. All image files are .JPG files, where their properties are explained below. Example images to the datasets are given in Figure 1, Figure 2, Figure 3 and Figure 4 respectively.

Figure 1. Samples from art images (D1).

Figure 2. Samples from art images (D2).

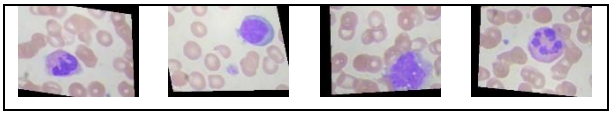

Figure 3. Samples from art images (D3).

Figure 4. Samples from art images (D4).

- Art Images (D1): The dataset includes 7819 instances (.jpg files) from 5 different classes with a resolution of 96 dpi and about 320 x 240 pixels each. These classes are drawing (including watercolor drawings), engraving, iconography, painting and sculpture [14]. Some corrupted files are excluded and 7718 instances are used in this study.
- Natural Images (D2): This is a benchmark dataset created for another study [15]. It includes 6899 instances (.jpg files) of 8 classes with a resolution of 96 dpi and about 320 x 240 pixels. The classes are airplane, car, cat, dog, flower, fruit, motorbike and person [16].
- Blood Cells (D3): The dataset includes 9957 instances (.jpg files) from 4 different classes with a resolution of 96 dpi and about 320 x 240 pixels each. These classes are eosinophil, lymphocyte, monocyte and neutrophil [17].
- Kitchenware Images (D4): The dataset includes 5214 instances (.jpg files) from 4 different classes with a resolution of 96 dpi and about 800 x 800 pixels. These classes are chair, kitchen, knife and saucepan [18]. Some corrupted files are excluded and 5200 instances are used in the study.

2.2 Preprocessing Data

One of the preprocessing steps for the image data is detecting corrupted files (if any). These corrupted files cause WEKA not to be opened properly and when the folders of the images are checked, some files, which cannot be opened, are detected. These files are deleted from the folders manually. After cleaning out some corrupted files in the datasets, a small program implemented in Java is used to rename all files with consecutive numbers. This is done for simplicity and to obtain the class names of the instances to an .arff file automatically. A sample .arff file for one of the datasets is given in Figure 5.

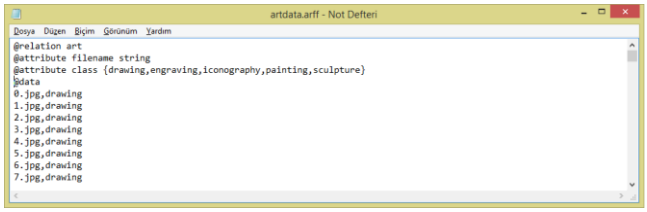

Figure 5. .arff file of art images (D1).

Since .arff file only contains data about the file names and classes of each instance, these data must be converted to some numerical format for prediction algorithms to be applied. In order to handle this conversion, image filters are applied to datasets. Filtering is transforming pixel intensity values to obtain some numeric data and this data reveals about the image characteristics [19]. It extracts features from image data and this data is written to the .arff file of the dataset. .arff file of each dataset is opened with WEKA and image filters, which can be found among unsupervised instance filters, are applied. The filters used are as follows:

- ColorLayoutFilter: This filter adds data about the spatial distribution of colors in an image [20]. The filter divides an image into 64 blocks and computes the average color for each block and then features are calculated from the averages [21].
- EdgeHistogramFilter: It is a very powerful filter especially on sketch-based images because it describes edge distribution with a histogram based on local edge distribution in an image [20]. It focuses on the edges of an image and takes shape information of the image into consideration for image indexing [22].
- BinaryPatternsPyramidFilter: It is used for extracting a pyramid of rotation-invariant local binary pattern histograms from images. A histogram of local binary patterns therefore encodes the larger scale patterns that occur across regions of images. These patterns are useful for texture and face recognition [23].
- FCTHFilter (Fuzzy Color and Texture Histogram): It encodes both color and texture information in one histogram. It is suitable for large image datasets [24]. This filter does the extraction of low level features that contain, in one histogram, color and texture information and an extension of these features so as to incorporate spatial information [25].
- SimpleColorHistogramFilter: It extracts color histogram features. It has three histograms for red, green and blue, each one having 32 bins. Each bin has the count of pixels that fall to that bin [26].
- PHOGFilter (Pyramid Histogram of Oriented Gradients): It encodes information about the orientation of intensity gradients across an image [27]. It consists of a histogram of orientation gradients over each image subregion at each resolution level. The distance between two PHOG image descriptors reflects the extent to which the images contain similar shapes and correspond in their spatial layout [28].

Table 1 gives the number of attributes that are generated with each filter. Filter numbers are for simplicity and are used in filter combinations.

Table 1. Number of attributes each filter generate

Filter No	Filter Name	# of Attributes
F1	ColorLayoutFilter	33
F2	EdgeHistogramFilter	80
F3	BinaryPatternsPyramidFilter	756
F4	FCTHFilter	192
F5	SimpleColorHistogramFilter	64
F6	PHOGFilter	630

When a filter is applied on the images, it adds its own numeric attributes to the dataset for every instance. These attributes help the classification algorithms to give more accurate results about the classes of the images. Fig. 2 shows the same .arff file given in Figure 6 after ColorLayoutFilter was applied.

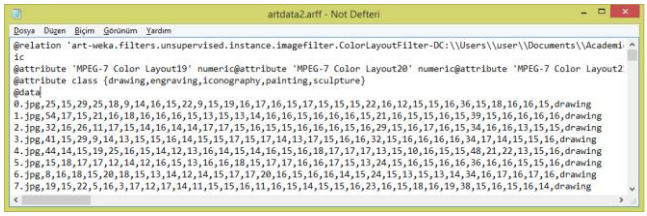

Figure 6. .arff file of art images after ColorLayoutFilter.

To obtain better results, the number of attributes for each instance can be increased. To do so, some of the filters are used together. Table 2 gives the filter combinations used together.

Table 2. Filter combinations

Filter #1	Filter # 2
F1	F2
F1	F4
F2	F5
F5	F1
F4	F5

These filter combinations are the best combinations for all 4 datasets. Other combinations have also been tested. However, no significant performance results has been got. The application order of filter combinations did not affect the performances in all filter pairs because with each filter WEKA generates and adds numeric data as attributes for an instance and every attribute of an instance are treated equally.

3. EXPERIMENTAL RESULTS

The chosen image filters were applied on four datasets explained in previous sections. After applying the filters mentioned above, machine learning algorithms are executed on data with additional attributes. In this study, Random Forest (RF) algorithm is used as being the best performing algorithm on image data. RF is a combination of tree predictors such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest [29]. When RF is used in image classification, each image is sent down every tree and the image is tested in internal nodes of the tree until it reaches the correct leaf. In WEKA environment, RF is executed with default parameters. The classification performances for all filters on datasets are given in Table 3.

Table 3. Classification performances of image filters

	Performance %			
Filter No	***D1***	***D2***	***D3***	***D4***
F1	78.08	77.34	78.88	76.13
F2	71.91	**84.36**	49.41	81.13
F3	72.17	80.01	43.52	73.13
F4	**79.49**	80.78	86.54	73.11
F5	78.34	71.79	**99.26**	68.36
F6	70.74	81.66	39.49	**82.07**

For art images dataset, applying FCTHFilter got the best classification performance. This dataset mostly includes drawings, paintings of people and filters identifying shapes did worse than other filters on this dataset. It was more successful in distinguishing iconography, painting and sculpture images than drawings and engravings.

EdgeHistogramFilter did the best for natural images dataset because the shapes of the classes in the dataset are more distinct. For this reason it had difficulties in classifying cat and dog images.

For blood cell images, although the images had very few colors, SimpleColorHistogramFilter got the best performance. This filter adds some attributes to image data in terms of three main colors (red, green and blue) and therefore it had a better performance with images of few colors. RF was equally successful in distinguishing the classes of blood cell images in this dataset.

For the kitchenware dataset PHOGFilter was the best performing filter because this filter deals with the orientations of subregions in images. Thus, RF can detect similar objects in a dataset with a better performance. It performed the best with chair images and the worst with knife images.

The combination of image filters had better classification performances than the results with only one filter applied on the datasets. The classification performances of filter combinations are given in Table 4 below.

Table 4. Classification performances of image filters

	Performance %			
Filters	***D1***	***D2***	***D3***	***D4***
F1-F2	81.29	88.40	70.78	84.40
F1-F4	80.95	83.99	89.55	79.63
F2-F5	**83.42**	**99.76**	92.73	**84.60**
F5-F1	80.11	80.87	98.66	78.11
F4-F5	81.06	81.56	**99.60**	75.54

To measure the classification performance of RF algorithm for on the datasets, precision, recall and f-measure values must be examined. Precision can be defined as the fraction of retrieved data that are relevant to the query. On the other hand, recall is the fraction of the relevant data that are successfully retrieved [30]. The f-measure is a measure for a test's accuracy. It combines precision and recall by calculating their harmonic mean [31]. The formula of f-measure is given in (1).

Fmeasure = (2*precision*recall) / (precision+recall) (1)

Figure 7 shows the f-measure values for the best results of the 4 dataset. Confusion matrices for best performances of 4 datasets are given in Figure 8, Figure 9, Figure 10 and Figure 11 respectively.

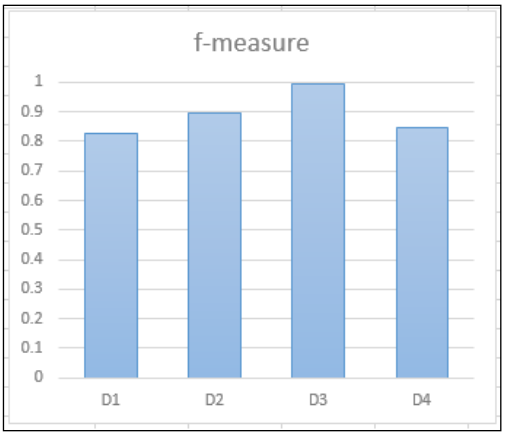

Figure 7. F-measure values for datasets for the best performing filter combinations.

```
   a    b    c    d    e   <-- classified as
 622  145  146   44  149 |    a = drawing
 165  476   57    6   53 |    b = engraving
  12   12 1972   17   64 |    c = iconography
  25    5  126 1800   86 |    d = painting
  40   11   85   31 1569 |    e = sculpture
```

Figure 8. Confusion matrices for art images.

```
   a   b   c   d   e   f   g   h   <-- classified as
 719   4   2   0   0   0   2   0 |   a = airplane
   0 944  11   3   7   2   1   0 |   b = car
   1  35 710 102  16   2   5  14 |   c = cat
   0  22 283 352  26   3   3  13 |   d = dog
   5   6  75  22 709  10   5  11 |   e = flower
   0   0   0   0   0 997   0   3 |   f = fruit
   0   1   0   1   4   0 782   0 |   g = motorbike
   0   0   5   1   0   0   0 980 |   h = person
```

Figure 9. Confusion matrices for natural images.

```
    a    b    c    d   <-- classified as
 2480    0    0   17 |    a = EOSINOPHIL
    4 2477    0    2 |    b = LYMPHOCYTE
    0    0 2476    2 |    c = MONOCYTE
   11    4    0 2484 |    d = NEUTROPHIL
```

Figure 10. Confusion matrices for blood cell images.

```
    a    b    c    d   <-- classified as
 1165   95   17   23 |    a = chair
   88 1089   27   96 |    b = kitchen
   98   83 1048   71 |    c = knife
   30  143   30 1097 |    d = saucepan
```

Figure 11. Confusion matrices for kitchenware images.

4. DISCUSSION

In this study, image filters were applied on four datasets and the classification performances were measured. Filters were applied to the datasets first one by one. Then combinations of two different filters were also applied to see whether the classification results were improving or not. Random Forest was chosen as the only machine learning algorithm. The reason for this was RF being the best performing algorithm on the mentioned datasets. Some others like J48 [1] was also tested but it did not get better results than RF. Best performance of J48 was for D3 dataset with SimpleColorHistogramFilter with 98.99%. For the same dataset, its best performance for FCTHFilter - SimpleColorHistogramFilter combination was 97.53%. For other datasets, applying one filter performed less than 73% and filter combinations performed less than 75% with J48.

Since different features measure different properties of an image, the filters have different effects on datasets chosen. The combinations of ColorLayoutFilter and EdgeHistogramFilter did not get any remarkable results in all of the datasets. But when EdgeHistogramFilter was combined with SimpleColorHistogramFilter, it was successful on 3 of the datasets. The best performance for this combination was gathered for natural images dataset. The dataset contains 8 classes, which has very distinct instances of these classes.

For the blood cells data, EdgeHistogramFilter was unsuccessful. Since EdgeHistogramFilter focuses on shapes of the images and the cell shapes in the dataset are seriously similar to each other, the result was not surprising. Instead, the combination of FCTHFilter and SimpleColorHistogramFilter obtained the best result for this dataset, which is the second best classification performance.

As one of the obtained results, the application order of the filters on the datasets did not affect the classification performances because these filters do not make any change on the image itself but they only add some numeric values as the attributes of the instances in the .arff file.

As a future work to this study, same datasets will be tested on one of the deep learning platforms and convolutional neural networks will be studied on various datasets for different performance evaluations on different domains.

5. REFERENCES

[1] Witten, I.H. and Frank, E. 2005. *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann Press, San Francisco, USA.

[2] Kamavisdar, P., Saluja, S., and Agrawal, S. 2013. A Survey on Image Classification Approaches and Techniques. *International Journal of Advanced Research in Computer and Communication Engineering*, 2, 1 (Jan. 2013), 1005-1009.

[3] Beham, M. P. and Mansoor Roomi, S. M. 2013. A Review Of Face Recognition Methods. *International Journal of Pattern Recognition and Artificial Intelligence*. 27, 4, 1-35. DOI= 10.1142/S0218001413560053.

[4] Gandhe, S.T., Talele, K.T., and Keskar, A.G. 2007. Intelligent Face Recognition Techniques: A Comparative Study. In *Proceedings of the ICGST International Journal on Graphics, Vision and Image Processing* (Delaware, USA, April, 2007). GVIP'07, 7, 2, 53-60.

[5] More, K., Kadam, P., Jadhav, A., and Dalgade, D. 2015. Face Authentication Application for Social Networking Site. *International Journal of Computer Science and Mobile Computing*. 4, 3 (Mar. 2015), 430-433.

[6] Yadav, P. and Yadav, N. 2015. Handwriting Recognition System - A Review. *International Journal of Computer Applications (0975–8887)*, 114, 19 (Mar. 2015), 36-40.

[7] Alaei, A. and Delalandre, M. 2014. A Complete Logo Detection / Recognition System for Document Images. In *Proceedings of 11th IAPR International Workshop on Document Analysis Systems* (Tours, France, April, 2014). DAS'14). DOI= 10.1109/DAS.2014.79.

[8] Lew, M.S., Sebe, N., Djeraba, C., and Jain, R. 2006. Content-based multimedia information retrieval: State of the art and challenges, *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM)*. 2, 1 (Feb. 2006), 1–19.

[9] Cortes, C. and Vapnik, V. 1995. Support Vector Networks, *Machine Learning*. 20, 273-297.

[10] Rokach, L. and Maimon, O. 2010. Decision Trees. In *The Data Mining and Knowledge Discovery Handbook*, Springer US. DOI= 10.1007/978-0-387-09823-4.

[11] Caudill, M. 1989. Neural Network Primer: Part I, *AI Expert*, 1989.

[12] Frank, E., Hall, M. A., and Witten, I. H. 2016. The WEKA Workbench. Online Appendix for D*ata Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann, Fourth Edition, 2016

[13] https://www.kaggle.com/

[14] https://www.kaggle.com/thedownhill/art-images-drawings-painting-sculpture-engraving

[15] Roy, P., Ghosh, S., Bhattacharya, S., and Pal, U. 2018. Effects of Degradations on Deep Neural Network Architectures, arXiv:1807.10108v2, 1-8.

[16] https://www.kaggle.com/prasunroy/natural-images/home

[17] https://www.kaggle.com/paultimothymooney/blood-cells

[18] https://www.kaggle.com/mbkinaci/chair-kitchen-knife-saucepan

[19] Voutsakis, E., Petrakis, E.G.M., and Milios, E. 2006. IntelliSearch: Intelligent Search for Images and Text on the Web. In *Proceedings of International Conference on Image Analysis and Recognition* (Povoa de Varzim, Portugal, September 18-20, 2006). ICIAR 2006, 697-708.

[20] Balasubramani, R. and Kannan, V. 2009. Efficient use of MPEG-7 Color Layout and Edge Histogram Descriptors in CBIR Systems. *Global Journal of Computer Science and Technology*. 9, 5 (2009), 157-163,

[21] ColorLayoutFilter, WEKA unsupervised instance image filter, URL: https://github.com/mmayo888/ImageFilter/blob/master/ImageFilter/src/weka/filters/unsupervised/instance/imagefilter/ColorLayoutFilter.java

[22] Shim S. O. and Choi, T. S. 2002. Edge color histogram for image retrieval, In *Proceedings of International Conference on Image Processing* (Rochester, NY, USA, September 22-25, 2002). IEEE, pp. 957-960. DOI= 10.1109/ICIP.2002.1037942

[23] BinaryPatternsPyramidFilter, WEKA unsupervised instance image filter, URL: https://github.com/mmayo888/ImageFilter/blob/master/ImageFilter/src/weka/filters/unsupervised/instance/imagefilter/BinaryPatternsPyramidFilter.java

[24] FCTHFilter, WEKA unsupervised instance image filter, URL: https://github.com/mmayo888/ImageFilter/blob/master/ImageFilter/src/weka/filters/unsupervised/instance/imagefilter/FCTHFilter.java

[25] Chatzichristofis, S.A. and Boutalis, Y.S. 2008. Fcth: Fuzzy color and texture histogram-a low level feature for accurate image retrieval, In *Proceedings of Ninth International Workshop on Image Analysis for Multimedia Interactive Services* (Klagenfurt, Austria, May 7-9, 2008). WIAMIS'08, IEEE, 191-196.

[26] SimpleColorHistogramFilter, WEKA unsupervised instance image filter, URL:https://github.com/mmayo888/ImageFilter/blob/master/ImageFilter/src/weka/filters/unsupervised/instance/imagefilter/SimpleColorHistogramFilter.java

[27] PHOGFilter, WEKA unsupervised instance image filter, URL: https://github.com/mmayo888/ImageFilter/blob/master/ImageFilter/src/weka/filters/unsupervised/instance/imagefilter/PHOGFilter.java

[28] Bosch, A., Zisserman, A. and Munoz, X. 2007. Representing shape with a spatial pyramid kernel, In *Proceedings of the ACM International Conference on Image and Video Retrieval* (Amsterdam, The Netherlands, July 9-11, 2007). ACM CIVR'07. 401-408.

[29] Breiman, L. 2001. Random Forests. *Machine Learning*. 5–32. DOI= 10.1023/A:1010933404324.

[30] Perry, J.W., Kent, A., and Berry, M.M. 1955. Machine literature searching X. Machine language; factors underlying its design and development. *American Documentation*. 6, 4, 242. DOI= 10.1002/asi.5090060411.

[31] Kılıçaslan, Y., Güner, E.S., and Yıldırım, S. 2009. Learning-based pronoun resolution for Turkish with a comparative evaluation, *Computer Speech & Language*. 23, 3, 311-331.

Authors' background

Your Name	Title*	Research Field	Personal website
Didem Abidin	Assistant professor	Machine learning, data mining	

***This form helps us to understand your paper better, the form itself will not be published.**

***Title can be chosen from: master student, Phd candidate, assistant professor, lecture, senior lecture, associate professor, full professor**