Feature Fusion for Image Classification

Khalid AlRebdi
Information Technology Department
College of Computer, Qassim University Buraydah, Saudi Arabia
Khalid.a.alrebdi@gmail.com

***Abstract*— Combining multiple features sets that include specific features like color, texture, and shape is a challenging task towards the improvement of the image classification process. This article covers the image-based classification area. We investigate the phenomenon of fusion of feature sets for the image classification problems. Our approach can be applied to all computer domains which are based on the features. We compare our approach with existing methods that extract features and apply it on other datasets. Our main objective is to get the optimist classification from fusing features of image classification. The objective is also to find whether a single good feature set is better than combining multiple features and whether the combination of many features increases performance.**

The feature fusion performs differently with each classifier and each dataset. From the experimental results, we note that the features reactions to fusion depend on their characteristics and we see that most of these features get benefit from fusion with the correct selection of the fusion rule.

Keywords— . Image classification, Features, machine learning

I. Introduction

Classifying an Image into known and recognizable objects is an important task in computer vision, it is a fundamental task towards image understanding and includes many applications, including object recognition, scene understanding, image tagging and recommendation [1].

Our work is to deal with the pixel-level classification which is a set of local features that includes color, texture, shape, etc. Each one of these properties plays a major part in classification. For instance, to discriminate football players from two teams, color information is crucial.

The feature combining process is known as gathering all of the important information from multiple features and their inclusion into a single one [2]. Adding more features will not always give you a better result of classification and that's why many practical cases prefer to analyze the different features separately, and later fuse together the results of the classifications.

We consider the fusion of features to get a classification which will hopefully be more reliable than using only one feature. In this Article, we are trying to analyze the relation of the classifiers with the feature set. Our objective is to find the idealist classification from fusing features of image classification.

II. Problem Specification and Motivation.

Image classification is fundamental to many computer vision applications. The image feature set plays an important role in image classification. In this article, the objective is to investigate the fusion of features for image classification. Fusion here means to combine two feature extraction methods. Given the image, the features are extracted and they are trained by the classifier. The feature calculation is based on different features approaches. for example, color features, texture features, and others. This thesis investigates the fusion of multiple features based on classifiers.

III. Study Scope

This article is related to feature extraction from images and then learning image classes from these features. Therefore, the article covers machine learning and computer vision. The tool used will be WEKA and MATLAB. The results of the article cover and can affect many applications domains where image processing and machine learning is needed.

IV. Datasets

For this article we used three datasets:

1) Faces dataset.
2) Birds dataset.
3) Butterflies dataset.

V. Classifiers

classifier model tries to draw some conclusion from observed values. Given one or more inputs a classification model will try to predict the value of one or more outcomes. Outcomes are labels that can be applied to a dataset.

There are two approaches to machine learning: supervised and unsupervised. In a supervised model, a training dataset is fed into the classification algorithm. That lets the model know what it is, for example, "authorized" transactions. Then the test data sample is compared with that to determine if there is a "fraudulent" transaction. This type of learning falls under "Classification".

Unsupervised models, on the other hand, are fed a dataset that is not labeled and looks for clusters of data points. It can be used to search data for similarities, detect patterns, or identify outliers within a dataset. A typical use case would be finding similar images. Unsupervised models can also be used to find “fraudulent” transactions by looking for anomalies within a dataset. This type of learning falls under “Clustering”.

There are several classification models. Classification models include logistic regression, decision tree, random forest, gradient-boosted tree, multilayer perceptron, one-vs-rest, and Naive Bayes [3].

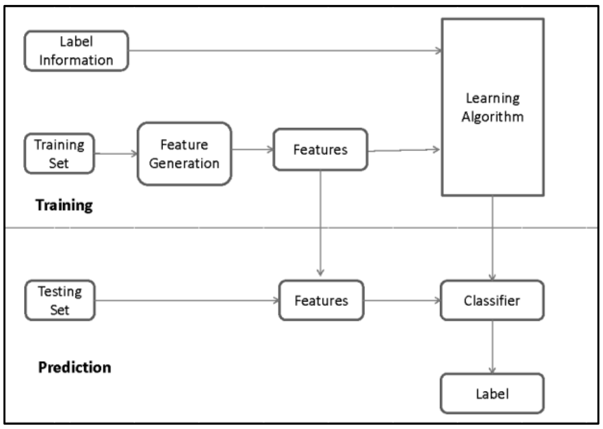


Figure 1: A General Process of Data Classification [4].

A. Bayesian Network

Bayesian networks are a type of Probabilistic Graphical Model that can be used to build models from data and/or expert opinion [5].
They can be used for a wide range of tasks including prediction, anomaly detection, diagnostics, automated insight, reasoning, time series prediction and decision making under uncertainty.

B. Naïve Bayesian

Naive Bayesian technique is based on the Bayesian theory and it is easy to build models that often outperform other methods of classification despite their simplicity. The Naive Bayesian algorithm predicts the measurement of patterns and relationships. Naive Bayes uses the concept of conditional probability and defines conditional probability is the probability of an event using prior knowledge.

C. Random Forest

An Algorithm is used for classification tasks and gives a wonderful result of the classification even if the information is not tuned and because of its flexibility is one of the most commonly used algorithms [6].
Random forest builds multiple decision trees and merges them together to get a more accurate and stable prediction. Also, it can be used for both classification and regression problems, which form most current machine learning systems.

VI. Features

A. Edge Histogram

An edge histogram in the image space represents the frequency and the directionality of the brightness changes in the image. It is a unique feature for images, which cannot be duplicated by a color histogram or the homogeneous texture features [7].

B. Color Layout

Color Layout is designed to capture the spatial distribution of color in an image [8]. The extraction process consists of two parts; a color selection representing the grid and a separate cosine conversion with quantization. Color is the basic quality of visual content, so it is possible to use colors to describe and represent an image. This specification suggests different ways to obtain these descriptors, and a specific color-naming tool is ColorLayout, which allows describing the color relationship between the sequences or the set of images. ColorLayout captures spatial mapping of representative colors on an overlapping grid on an area or image. The representation depends on the CT transactions. This is a very small descriptor for high efficiency in browsing and quick search applications. It can be applied to still images as well as to video clips.

C. PHOG

Color images provide stronger discriminating information than grayscale images and color-based image search can be very effective for face, object, scene, and texture image classification. Some desirable properties of the descriptors defined in different color spaces include relative stability over changes in photographic conditions such as varying illumination. Global color features such as the color histogram and local invariant features provide varying degrees of success against image variations such as rotation, viewpoint and lighting changes, clutter, and occlusions. The shape and local features also provide important cues for content-based image classification and retrieval. Local object shape and the spatial layout of the shape within an image can be described [9].

D. Simple Color Histogram

is the representation of the distribution of colors in an image. For digital images, a color histogram represents the number of pixels that have colors in each of a fixed list of color ranges that span the image's color space, the set of all possible colors. Used in photo processing, color schemes can be created for any type of color space, although this term is usually used for

3D spaces like RGB or HSV. For monochrome images, the density chart can be used instead. For multi-spectral images, where each pixel is represented by a random number of measurements (for example, outside the three measurements in RGB), the color graph is N-dimensional, with N being the number of measurements taken. Each measurement has its own wavelength range from the optical spectrum, and some may be outside the visible spectrum. If the set of possible color values is small enough, each of these colors can be placed on a scale by itself; the graph is just the number of pixels that each color contains. Often, space is divided into an appropriate number of ranges, often arranged as a normal grid, each of which contains many similar color values. The color graph can also be represented and displayed as a smooth function across the color space that approximates the number of pixels. Like other graphing patterns, a color scheme is a statistic that can be considered as a continuous distribution of color values [10].

VII. Experimental Results and Analysis

Results are beneficial, but it will be much more valuable when they are analyzed. In this part of the article, we will discuss the results of experiments applied to the four features.

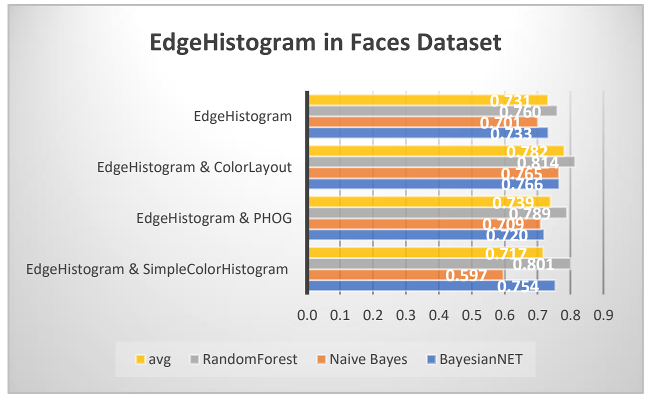
Our analysis will be based on the features. We'll compare the performance of features with each other and with the three classifiers in every dataset.

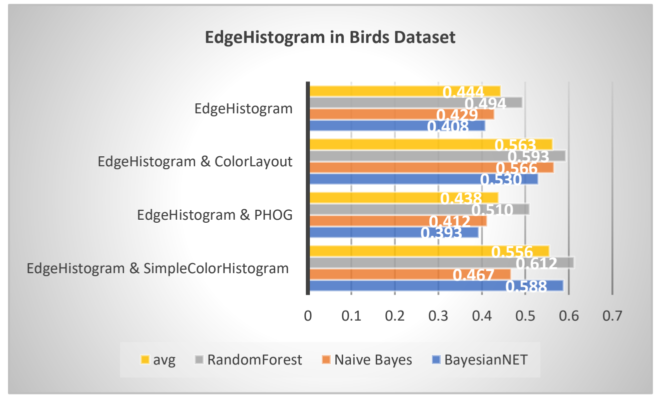
A. *Edge Histogram*

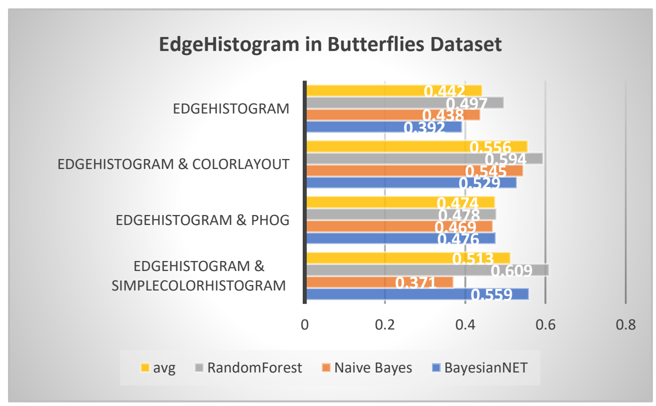
It seems that the EdgeHistogram usually gets a very good boost from fusion especially while applying the Random Forest classifier. However, it gives you a minor boost or decreases the performance while being fused with PHOG filter so the PHOG and EdgeHistogram are a bad combination. The EdgeHistogram is very effective when it is combined with ColorLayout that it gives you a better result than EdgeHistogram alone every time. For instance, a 35% increase in the F-measure in the Butterflies dataset using Bayesian Network.

The fusion of EdgeHistogram and SimpleColorHistogram looks like it depends on the classifier because it can give you surprisingly the optimist results like 42.6% increase in the Butterflies dataset using Bayesian Network and it can give the worst results as well such as 15% decrease in the same dataset using Naive Bayesian. With the random forest and the Bayesian Network classifiers, the EdgeHistogram gets its best fusion with SimpleColorHistogram two out of three datasets but with the Naive Bayesian this combination decreases performance and give the worst results two out of three datasets.

Therefore, with the right classifier and the correct fusion, the EdgeHistogram should get a great increase that shows how effective is the feature fusion.







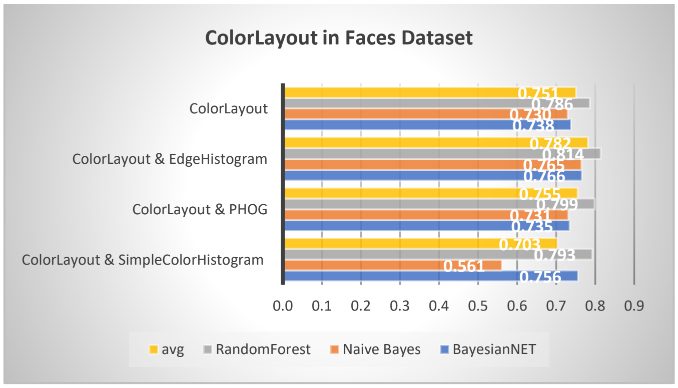
Figures 2-3-4. EdgeHistogram results in each Dataset.

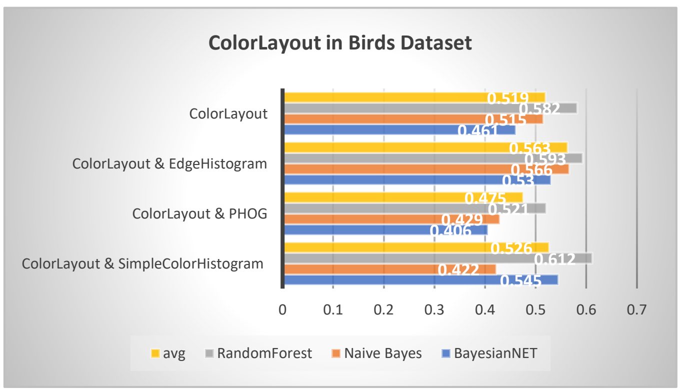
B. *Color Layout*

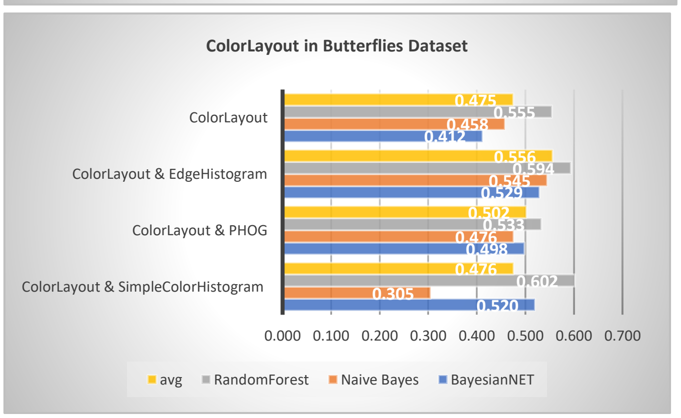
Unlike the EdgeHistogram the ColorLayout feature usually gets only a slight boost from the feature fusion and that may be due to its impressive performance on its own. Its best combination is with the EdgeHistogram always increasing the performance. On the other hand, The ColorLayout feature gives unstable results when it is fused with PHOG such as 20.8% Increase in the Butterflies dataset applied by Bayesian Network and 11.9% decrease in the Birds dataset applied by the same classifier and that might be due to the nature of PHOG. ColorLayout and SimpleColorHistogram seem like a decent combination when the right classifier is chosen. For instance, when we used the Bayesian Network classifier the

SimpleColorHistogram increased the F-measure of ColorLayout by 26.2% in the Butterflies dataset. However, when these two features are applied by Naive Bayesian the SimpleColorHistogram always reduces the F-measure of ColorLayout significantly such as 23.1% decrease in the Faces dataset.

Overall, ColorLayout is an efficient feature that does good on its own but with the feature fusion, it can get a considerable boost.







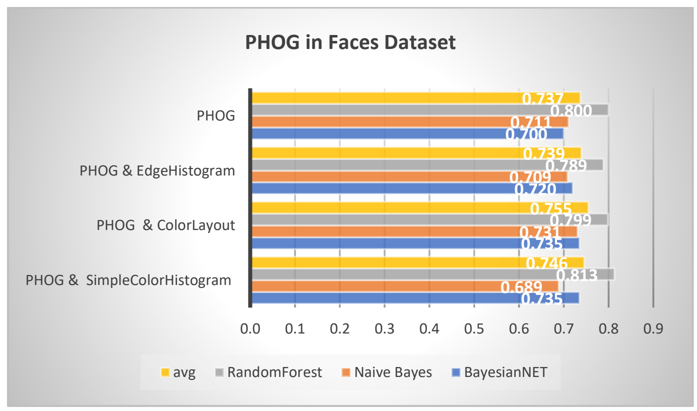
Figures 5-6-7: ColorLayout results in each dataset.

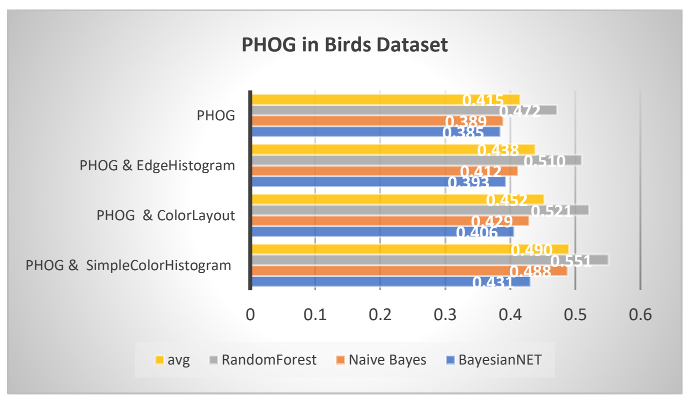
C. PHOG

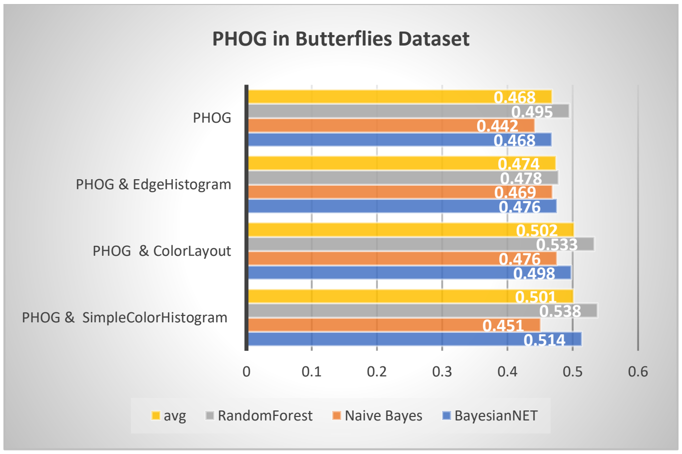
The PHOG seems like it gives different performance based on the datasets more than the features those who are fused with it. The performance is varying with each dataset. For example, when we fused the PHOG with the SimpleColorHistogram applied by Naive Bayesian we got a 24.4% boost in the Birds dataset and a 3.09% decrease in the Faces dataset and 2.03% increase the Butterflies dataset. The PHOG is the only feature that gives better results in the Butterflies dataset more than its results in the Birds dataset.

The PHOG doesn't regularly gets a great increase from fusion but sometimes it scores good results like the 10.3% increase when it was fused with ColorLayout in the Birds dataset using Random Forest. The PHOG best combination is with the SimpleColorHistogram. The performance of the two-features combined is very decent considering that the PHOG usually gets the least results between the four features. The Bayesian Network Always gets a boost from fusion in all three datasets so it is not useless to fuse features with PHOG.

In conclusion, the PHOG is an unstable and unpredictable feature that probably counts on the nature of data used more than the classifier.







Figures 8-9-10: PHOG results in each dataset.

D. Simple Color Histogram

It looks like SimpleColorHistogram works badly with Naive Bayesian that it only got the F-measure 0.23 in the Butterflies dataset. Giving that poor result it will surely get greater outcome when it is fused with any feature. However, the SimpleColorHistogram is an efficient feature that it scores outstanding results with other classifiers such as 0.543 in the Butterflies dataset.

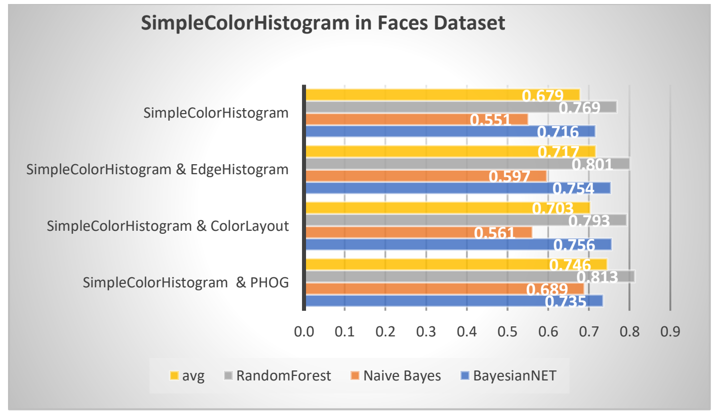
The fusion of features with SimpleColorHistogram seems very effective because the results of fusion are more likely to outperform the results without fusion such as the fusion with EdgeHistogram 32.4% increase in the Butterflies Dataset using the Bayesian network Algorithm.

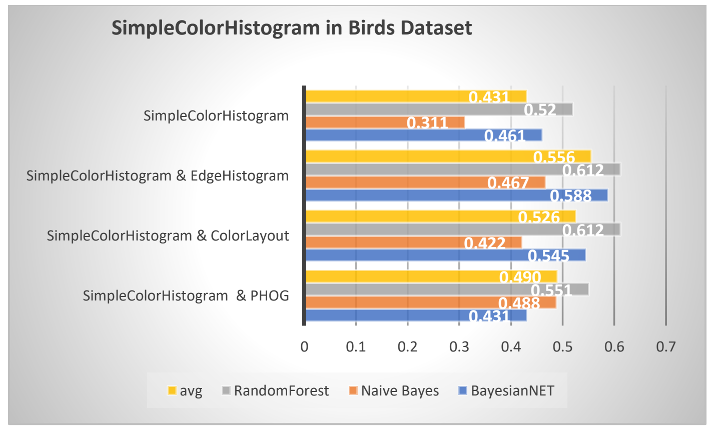
The best combination of SimpleColorHistogram is confidently the EdgeHistogram this feature gives SimpleColorHistogram the best performance in two datasets the Birds and the butterflies whereas the PHOG leads the Faces dataset. However, we say that the EdgeHistogram is better for SimpleColorHistogram than PHOG because it improves the F-Measure every time while the PHOG did get a 6.5% reduction in the Birds dataset when the fusion was applied by Bayesian Network Classifier.

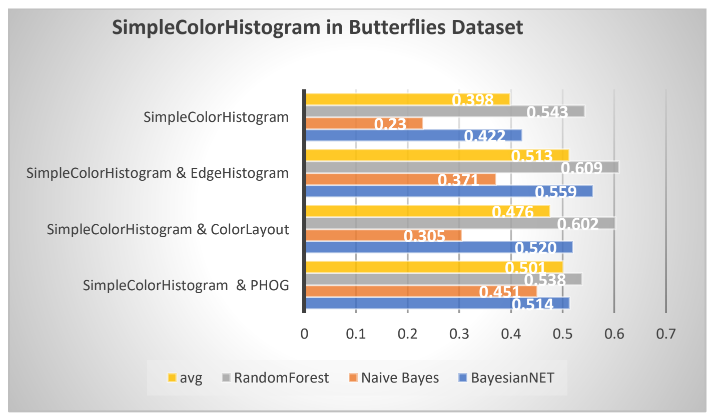
The ColorLayout feature is similar to EdgeHistogram that it always boosts the performance such as the 17% boost in the Birds dataset using Random Forest.

The greatest increase in this entire research was obtained by the SimpleColorHistogram and PHOG. The later feature has boosted the first by over than 96% while applying Naive Bayesian in the Butterflies dataset. However, this number can be understood because of the very poor performance by the unfused SimpleColorHistogram classified by Naive Bayesian.

In general, the SimpleColorHistogram got the most benefit from feature fusion with only two reductions out of 27 results.







Figures 11-12-13: SimpleColorHistogram results in each dataset.

VIII. Major Findings

- If properly adjusted (fusion rule selection), the fusion of features should increase classification performance.
- Fusion performance depends on the selection of classifiers used for fusion.
- Fusion performance is also affected by the rule selected for fusion.
- Fusion may reduce the performance if the fusion rule is not selected and investigated.
- Some features depend on the nature of images data more than the classifier or the feature that are fused with them.

• The disadvantage of fusion is the increased amount of processing time.

IX. Conclusion

Finally, it seems that feature fusion is very helpful in the image classification area that it increases the accuracy of classification, but the fusion rule must be properly selected.

Some classifiers do not work efficiently with some features and some features are bad with some images data so the right combination should be carefully selected in order to improve the image classification.

It is better to test a dataset that is hard to be classified than an easy one such as Faces dataset because the differences between the outcomes are barely noticeable.

Overall, the feature fusion is recommended to be used, but you need to know the nature of features and which feature combination can benefit you the most.

Acknowledgment. Firstly, we thank Allah for his graces, then are we are thankful to our supervisor Dr. Rehan Ullah Khan for his support, guidance, and continuous encouragement throughout the article.

REFERENCES

[1] Lingxi Xie, Qi Tian ,Bo Zhang. (2016). IEEE Transactions on Circuits and Systems for Video Technology. Simple Techniques Make Sense: Feature Pooling and Normalization for Image Classification. 26 (1), 1251 – 1264.

[2] Guyon, I. and Elisseeff, A. (2003) An Introduction to Variable and Feature Selection. The Journal of Machine Learning Research, 3, 1157-1182.

[3] Medium. (2018). Machine Learning: Classification Models – Fuzz – Medium. [online] Available at: https://medium.com/fuzz/machine-learning-classification-models-3040f71e2529 [Accessed 10 Nov. 2018].

[4] Anon, (2018). [image] Available at: https://www.semanticscholar.org/paper/Feature-Selection-for-Classification%3A-A-Review-Tang-Alelyani/310ea531640728702fce6c743c1dd680a23d2ef4 [Accessed 10 Nov. 2018].

[5] introduction, B. (2018). Introduction to Bayesian networks. [online] Bayesserver.com. Available at: https://www.bayesserver.com/docs/introduction/bayesian-networks [Accessed 10 Nov. 2018].

[6] NiklasDonges. (2018).the Random Forest Algorithm.Available: https://towardsdatascience.com/the-random-forest-algorithm-d457d499ffcd. Last accessed 12th Nov 2018.

[7] Anon, (2019). [ebook] Available at: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.182.5856&rep=rep1&type=pdf [Accessed 7 Apr. 2019].

[8]. Kasutani, E., Yamada, A. (2001),"The MPEG-7 color layout descriptor: a compact image feature description for high-speed image/video segment retrieval", International Conference in Image Processing, Proceedings, 1:674-677.

[9] Anon, (2019). [online] Available at: https://pdfs.semanticscholar.org/e819/a577c57c83a133a0a0e81180d14dc13b82e9.pdf [Accessed 1 May 2019].

[10] Anon, (2013). [ebook] Available at: https://pdfs.semanticscholar.org/279b/6bfa3edac97b0fb907483b4dabbc79635c4d.pdf [Accessed 1 May 2019].