A Novel Approach for the Implementation of Classification Algorithms for Detecting Shadow and Non-Shadow Regions of an Image

Kamaljit Kaur
Asst. Professor
Lovely Professional University
Phagwara-Punjab

Heena Chawla
Student M.Tech
Lovely Professional University
Phagwara-Punjab

ABSTRACT

Many image analysis tasks require a classification procedure to identify the shadowed and non-shadowed areas of an image, so that once the shadowed areas are known, the clarity of the image can be improved. The main aim is to detect the anomalous regions of an image. There are numerous techniques of doing so. One of the best ways is to find hyper plane that can be identified to find the difference between the shadow and non shadow areas in the image by using classification algorithm. Thus this research will be to compare various classifiers on the basis of various parameters that which one proves the best in terms of identifying the boundary of shadow and non shadow areas in images. The classification algorithms used in this case would be Support Vector machine (SVM), K- Nearest neighbour (KNN) and Bayesian networks.

Keywords

Shadow, hyper plane, Support Vector Machine (SVM), K-Nearest Neighbor(KNN), Bayesian classifier, segmentation, seed points, scene analysis, true positive, true negative, false positive, negative, sensitivity, specificity, accuracy.

1. INTRODUCTION

In computer visualization algorithms, shadows in images have proved troublesome from a long time.

Because of the unlikable effects of shadows in the process of analyzing images, much attention has been paid to the area of shadow removal over the past decades and many improved the accuracy of many applications Such as traffic surveillance, face recognition and image segmentation.

1.1 Shadow

Shadows occur when direct light coming from a light source is partially or totally blocked. Shadows are mainly of two types: cast shadow and self shadow.

The self shadow is that region of an object that is not lightened up by direct light; the cast shadow is formed when light or an image is made to fall on a surface. It is mostly the dark area projected by an object on the background. Cast shadow can be further classified into umbra and penumbra region.

The umbra region is that part of a cast shadow where light coming directly from a light source is completely blocked; the penumbra region is that part of a cast shadow where direct light is blocked up to some extent. [3]

On a Technical basis, shadow detection methods can also be classified as property-based and physics –based shadow detection methods. Physics-based techniques need some prior knowledge, such as light and geometry, camera calibration, or indoor scenes. However, it is a complicated process to obtain the accurate model for a random scene because the complexity of environments and variation of light sources from time to time and from place to place. Hence, mostly physics-based techniques are used for particular applications, such as moving cast shadow detection and shadow detection in aerial images. Scene knowledge is utilized in Physics-based methods due to which these methods are only used in specific applications they are designed for. The algorithm may not work well when the application environments are different. Property-based techniques utilize shadow features for shadow detection. The most obvious and prominent feature of a shadow is that it darkens the surface it spreads on, and this feature is used by almost all methods.[11] Other features like edge, histograms, texture, geometry , color ratios, and gradient are also used widely. Sometimes, only one feature is not enough. For example, we have the feature that shadows usually have lower pixel values, but pixels with lower values may not necessarily be shadows.

It is somewhat simple to identify shadow and non-shadowed regions in sequential images, but it is quite complicated in a single image as the amount of information available is quite less.

Although clustering is widely used in shadow and non-shadow detection, classification algorithms can also be used. There are various classification algorithms which can be used here.

1.2 Classification

Classification is a kind of supervised learning, in which learning is acquired with the help of a teacher. Here we categorize a new object to a pre-defined class and then assign a label to it. It is done on the basis of a set of instances whose category membership is already known. This set is called training set. An algorithm that implements classification is known as a classifier. Here the classifiers used are Support Vector Machine (SVM), K-Nearest Neighbour algorithm (KNN) and Bayesian networks.

1.2.1 Support Vector Machine (SVM)

SVM mostly deals with binary classification. In binary classification, it classifies a given set of objects into two groups on the basis of possession of some property. It is also a linear classifier as it takes the decision of classification on the

basis of linear combination of characteristics. In SVM, given a set of training examples, which are assigned one of two categories, an SVM training algorithm builds a model that predicts the assignment of new examples to one category or the other. In SVM model, the examples are represented as points in space, such that the examples of different categories are divided by a clear gap, whose width should be as large as possible. New examples are then put into that same space and predicted to belong to a category based on which side of the gap they fall on. SVM model constructs a hyper plane which clearly divides the data into two classes. The margin of the hyper plane should be maximum.

1.2.2 K-Nearest Neighbour (KNN)

In KNN, the learning is somewhat lethargic. It consists of two phases:

-Training phase- In this phase, we save the feature vectors and class labels of the training data in some storage.

-Classification phase- In this phase, the user defines a constant **k** with his own choice. Then a new input vector is given the label which is most repeated among the *k* training samples nearest to that query point.

1.2.3 Bayesian Classifier

A naive Bayes classifier is based on the concept of Bayes' theorem of probability. That is why it is sometimes called as probabilistic model. It is kind of an independent feature model, which means that Bayes classifier is based on the assumption that the presence (or absence) of a particular feature of a class is not linked to the presence (or absence) of any other feature, given the class variable. An advantage of the naive Bayes classifier is that a small amount of training data is enough for it to guesstimate the means and variances of the variables, which are the necessary parameters for classification. [12]

Thus individual classification algorithms can be used and their results can be visualized and compared.

2. PROPOSED APPROACH

Here, an approach is proposed which detects the shadows and non-shadow areas of a digital image using classification algorithms.

The methodology of this approach goes like this:

1) <u>Develop a representative dataset of natural images having shadow</u>- In this step, the images containing shadows are captured using a digital camera of good resolution.
   The following things should be kept in mind while capturing images:-
2) <u>Pre-processing of images:</u>
   This step consists of following steps:-
   -Resizing of images: All the images are to be resized to a particular size (256 by 256).
   -Heavy image compression: The heavy images are to be compressed to make the processing easier. This can be done using online compression tools available on the internet.
3) <u>Scene analysis:</u> The following steps are performed in this step:
   - The scene analysis is performed on the pre-processed images.
   - The main two tasks performed in scene analysis are:

i) Region-growing segmentation: This is also known as pixel based image segmentation. Here any initial seed point from a particular region of the image is selected. Then starting from the seed point, each pixel is analyzed and segmented as shadow or non-shadow pixel and is eventually assigned to that particular region. When the segmentation is completely performed, the image which we get as output clearly discriminates the shadow and non-shadow area and darkens the regions which contain shadow. Now this image, which we get after performing segmentation, is subjected to feature extraction.

ii) Feature extraction:

- The classification algorithms cannot be applied on the image which we are getting as output of scene analysis. We need to extract the features of entities of the image.
- Here the features of each pixel of the image are extracted and it generates a sheet containing the attributes of entities in that image and their dimensions including entropy, grey level, brightness etc.
- The classification algorithms cannot be applied on the image which we are getting as output of scene analysis
- We can define as many parameters as we like. The more the parameters chosen, the more accurate is the classification.

4) <u>Applying classification algorithms</u>:- The output that we get after performing scene analysis, acts as an input to the classification algorithms like Support Vector Machine (SVM), K- Nearest neighbour and Bayesian classifier.
   What these classifiers does, they use the features contained in the output sheet of scene analysis to assign a label to each and every pixel of image that whether that particular pixel belongs to a "shadowed" label or a "non-shadowed" label.

Then each of the classifiers is tested on the basis of various parameters like false negative, false positive, true positive, true negative, accuracy etc. Then conclusion is drawn that which is the best classifier.

3. IMPLEMENTATION

The dataset that is collected for the implementation is the set of natural images captured through digital camera, keeping in mind the conditions mentioned before.
The tool selected for implementation is MATLAB version 7.12.0 (R2011a). Both scene analysis and classification algorithms can be implemented in MATLAB only as it deals best with the images. When SVM, KNN and Bayesian classifiers are applied on the sheet of scene analysis, then graphs are plotted for each of the classifiers for each parameter.
The following parameters are used:
False positive: A **false positive error** is also known as a "**false alarm**". It evaluates a condition as true, when it is actually false. The graph plotted against false positive parameter is as follows:
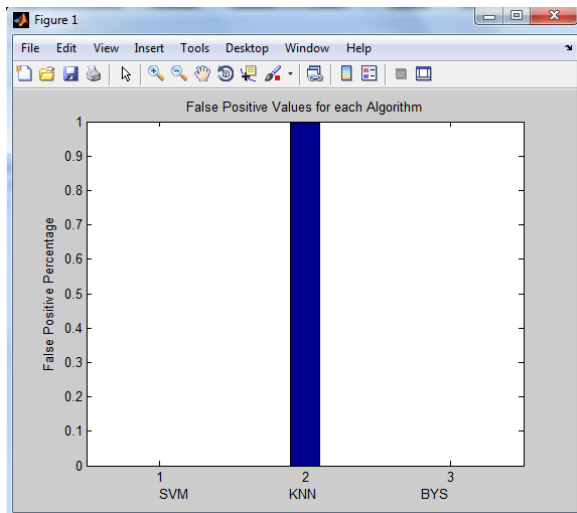
Fig 1. Representation of false positive for SVM, KNN and Bayesian

In case of false positive rate, KNN does not give a good result as it shows maximum number of false positives values.

False negative: A **false negative error** is where a test result declares the failure of a condition, while it actually was successful. The following graph was plotted against this parameter:
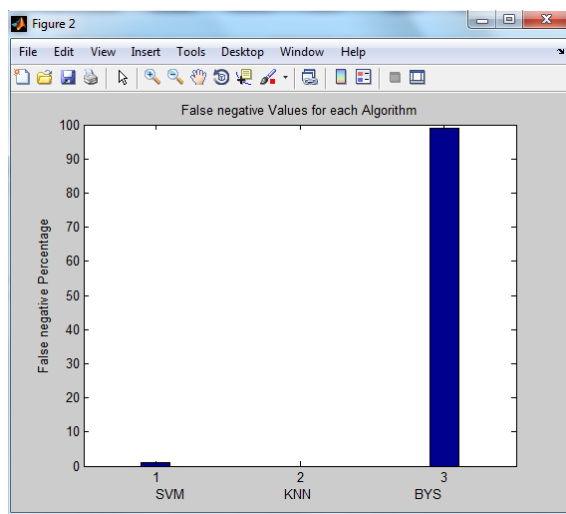

Fig 2. Representation of false negative for SVM, KNN and Bayesian

Thus in case of false negative parameter, the Bayesian classifier gives a poor performance as it gives the highest number of false negative values.

True positive: True positive means correctly identified. It means it correctly identifies the shadow and non-shadow region. The following graph was plotted against this parameter:
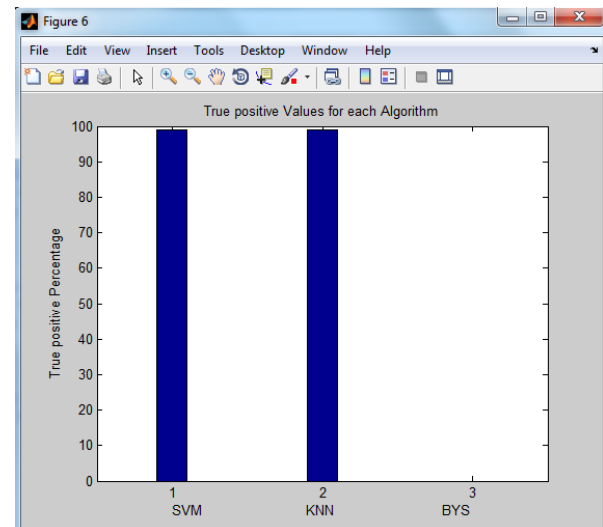

Fig 3. Representation of true positive for SVM, KNN and Bayesian

In true positive parameter, SVM and KNN prove to be equally good.

True negative: It is equivalent to correct rejection. A true negative (TN) has occurred when both the predicted value and the actual value are negative. It means that the classifier truly indicates a non-shadow area as non-shadow. The following graph is plotted against this parameter:
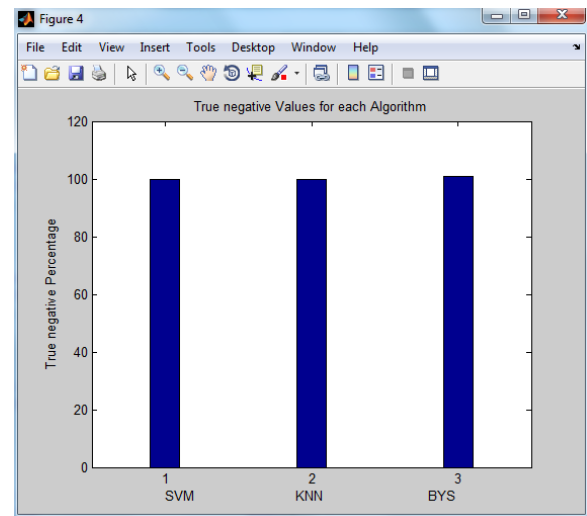

Fig 4. Representation of true negative for SVM, KNN and Bayesian

In case of true negative parameter, SVM, KNN and Bayesian, all the three classifiers give equal results. The Bayesian classifier shows it a bit better than the other two.

Sensitivity: Sensitivity measures the performance of a binary classification test. **It is also known as recall rate.** It measures the fraction of actual positives which are correctly recognized as such. The following graph is plotted against this parameter:
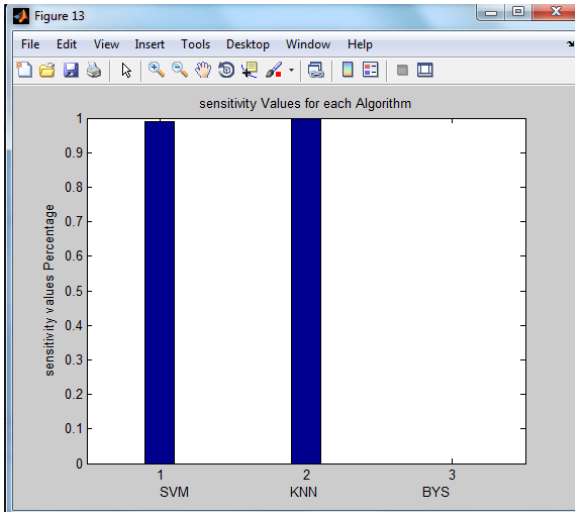
Fig 5. Representation of sensitivity for SVM, KNN and Bayesian

In this case, KNN gives the best result.

Specificity: Specificity measures the share of negatives which are correctly recognized (e.g. the percentage of healthy people who are correctly identified as not having the condition). The following graph is plotted against this parameter:
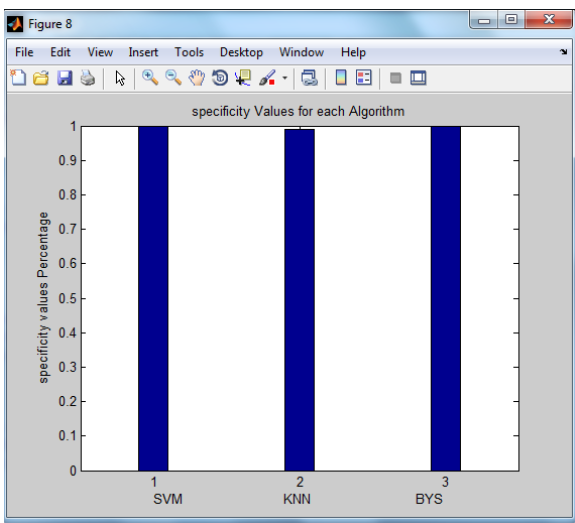

Fig 6. Representation of specificity for SVM, KNN and Bayesian

SVM and Bayesian prove to be equally good. The performance of KNN is also comparable to the other two.

Accuracy: The **accuracy** of an algorithm means the degree of closeness of the measurements predicted by the algorithm to the actual (true) value.
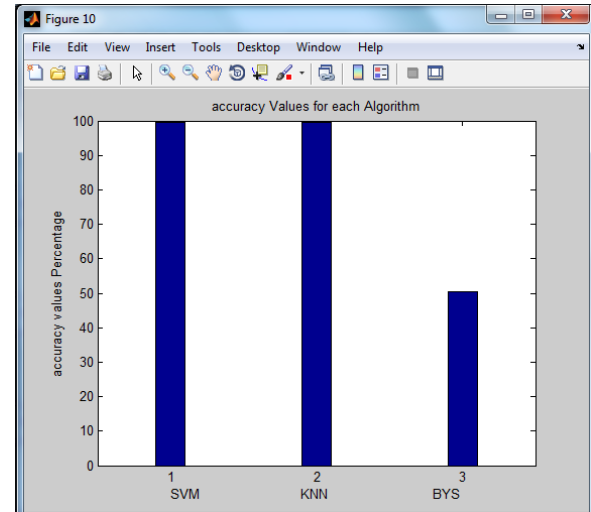

Fig 7. Representation of Accuracy for SVM, KNN and Bayesian

SVM and KNN are the most accurate in detecting shadow and non-shadow areas of an image. The accuracy of Bayesian network is not up to the mark.

Likewise, the graphs for positive predicted value and negative predicted value are also plotted.

4. CONCLUSION AND FUTURE WORK

Thus after analyzing all the classifiers on the basis of above mentioned parameters, it is concluded that KNN is the most suitable and Bayesian is the least suitable in detecting the shadowed and non-shadowed regions of an image, thereby improving the clarity of images for the accuracy of a number of practical applications like traffic surveillance, face recognition etc. There are numerous types of scene based images which include landscape; portrait etc. and many of these images might be having shadows which may lead to reducing of aesthetic values. This might require some more exploration in terms of segmentation and feature extraction to accurately detect shadow.

Thus in future, some more techniques of scene analysis may be tried which may include graph cut segmentation techniques etc. to actually increase the overall aesthetic value of an image.

5. REFERENCES

[1] Airton M. Polidorio, F. C. Flores, Nilton n. Imai, Antonio M., G. Tommaselli and Clélia Franco (2003) - "Automatic shadow segmentation in Aerial color images", XVI-Brazilian Symposium on Computer Graphics and Image Processing ,pp. 270-277.

[2] Andrea Prati, Ivana Mikc, Costantino Grana and Mohan M. Trivedi (2001) –"Shadow Detection Algorithms for Traffic Flow Analysis: a Comparative Study", IEEE Intelligent Transportation Systems Conference, pp. 340-345.

[3] Elena Salvador, Andrea Cavallaro, Touradj Ebrahimi(2001)- 'Shadow identification and Classification using invariant color models", ICASSP01, pp. 1545-1548.

[4] Li Xu, Feihu Qi and Renjie Jiang (2006)-"Shadow removal from a single Image", Intelligent System Design and application, pp.1049-1054.

[5] Mei Xiao, Chong-Zhao Han and Lei Zhang (2007)- "Moving Shadow Detection and Removal for Traffic Sequences" International Journal of Automation and Computing, pp.38-46.

[6] M.A. Egan -'Locating Clusters in Noisy Data: A Genetic Fuzzy c-means Clustering Algorithm", Fuzzy Information processing Society – NAFIPS, conference of the North American, pp.178- 182.

[7] Tai-Pang Wu and Chi-Keung Tang (2005)-"A Bayesian Approach For Shadow Extraction From a Single Image", Tenth IEEE International Conference on Computer, vol.1, pp.480-487.

[8] Yan Li, Tadashi Sasagawa, Peng Gong -A system of the shadow detection and shadow Removal for high resolution city aerial photo.

[9] Yan Li, Tadashi Sasagawa, Peng Gong -A system of the shadow detection and shadow Removal for high resolution city aerial photo.

[10] Yu-jiao XiaHou,Sheng-rong Gong "Adaptive Shadows Detection Algorithm Based on Gaussian Mixture Model",International Symposium on Information Science and Engineering, 2008,pp.116-120.

[11] Gonzalez, Woods, and Eddins(2009), Digital Image Processing Using MATLAB 2nd Ed.