

Intelligent Grading and Motion Prediction Dynamics of Tobacco

Tawanda Mushiri

Department of Mechanical Engineering
University of Johannesburg
P.O Box APK 524
Johannesburg
South Africa

tawandanda.mushiri@gmail.com, tawandamushiri123@hotmail.com

Napoleon Sithole

Department of Mechanical Engineering
University of Zimbabwe
P.O Box MP167
Mt Pleasant
Harare
Zimbabwe

sitholenapoleon@gmail.com

Charles Mbohwa

Faculty of Engineering and the Built Environment
University of Johannesburg
P.O Box APK 524
Johannesburg
South Africa
cmbohwa@uj.ac.za

Abstract

One of the most important yet ancient technique in the tobacco industry, is the grading of flue-cured tobacco. Factors to be considered include overall color, blemish, damage, texture, leaf length and ripeness. It is therefore imperative that leaves from different plant positions (different groups) i.e. primings, lugs, cutters, leaf, and tips must be kept separate because they have important different chemical characteristics. The efforts of this research project are directed towards restoring the objectivity in the grading of flue-cured tobacco in Zimbabwe, by means of using Machine Vision and Artificial Intelligence during the grading process. MATLAB software was used to implement elements of AI specifically, Computer Vision System and Machine Learning. An artificial grading expert system was designed by means of supervised machine learning particularly Support Vector Machines. This was achieved by training a data sample of flue-cured tobacco according to human expert knowledge of tobacco grading. A test subject is then tried and the system is evaluated and it was found that the system is 93.1% efficient which showed that automating the grading system in Zimbabwe is not only feasible but it is also more economic.

Keywords

Artificial intelligence (AI), Machine learning (ML), Machine Vision (MV), Support Vector Machines (SVM).

1. Introduction

The classification of flue-cured tobacco leaves is a rigorous task, and the complexity and variations of tobacco leaves means that most of the grading is manually operated(Zhang and Zhang, 2011), (Chincholkar et al.). Flue-cured tobacco must be graded by color, texture and size, and all this will be supported by a seasoned expert's perception towards the fine properties of the leaves(Patrick-Burns, 2014). Since the experts are only human, error often occur when they are tired(Shappell and Wiegmann, 2012), (Shappell and Wiegman, 2003). The results of the quality evaluation and classification heavily depends on factors such as, the expert's emotional state, illumination condition, human eyesight, etc(Jain et al., 2004). The classification very laborious, making it subjective and experientially based(Zhang and Zhang, 2011). The process error rate stability and overall efficiency is not satisfying(Tanaka et al., 1998). New and modern are solutions needed to improve this ancient technique of classifying flue-cured tobacco leaves(Livingston, 1989).

Most of the features used to evaluate the quality of tobacco leaves relate to human vision and include features such as color, texture, shape, extra features, etc(Zhang and Zhang, 2011). Because of developments in the areas of artificial intelligence, image processing, statistics and machine learning, and machine vision, the automated classification of flue cured tobacco leaves slowly becoming a reality(Devlin and Lorden, 2007), (Stern et al., 2014). Research on automating the classification of tobacco leaves is still going on and some work have been presented(Zhang and Zhang, 2011). Zhang X and Zhang F 2011 presented a transformation technique from RGB signals to the Munsell system for the color analysis of tobacco leaves. Zhang and Zhang 2011 proposed also a neural network based algorithm that extract the features of tobacco leaves. Classification and quality evaluation of tobacco leaves based on image processing and fuzzy comprehensive evaluation(Zhang and Zhang, 2011).

Emulating a human well-seasoned expert is nothing but a very challenging task yet a thought on an expert system based on image processing and Support Vector Machines (SVM) encourages an attempt to emulate a seasoned flue cured tobacco classification expert(Munson and Paulson, 2003), (Ehin, 2004). The argument of this paper is that the SVM expert system is very capable of classifying and evaluating the quality of tobacco leaves without much need of background computations.

2. Materials and Methods

This section is intended to clearly outline all the software and hardware which will be used throughout. In order to acquire data on the objects, pictures of them were taken using a web cam. This required fixed lighting and camera angle. To facilitate this, a worktable is built that has an overhead frame where the lights and camera are installed. The camera is connected to a computer where the captured images are processed and analyzed. The image acquisition, processing and the SVM expert systems were programmed using Matlab.

2.1 Worktable

Dimension: 1.0 x 1.5 m

Height: 1 m from the floor to table surface.

1.3 m from the table surface to the table ceiling.

Initially, the lighting source was the sun as this removed the need to calculate deviations from the lighting intensity from the sun.

2.2 Computer

Specifications required: Pentium III Celeron 2.2 GHz, 512MB RAM, Windows XP are the recommended requirement for running Matlab.

Specifications used: Intel core i3 2.4 GHz, 8 GB RAM, Intel HD 4000 graphic card, Windows 10

2.3 Lights and illuminators for machine vision systems

The sensitivity of color cameras typically is less than ten times than monochrome cameras. Selection of the lens with suitable brightness according to lighting conditions is crucial. Light sources: LED, incandescent, fluorescent.

2.4 Camera

Model: Creative Live! Ultra or Intex USB webcam or Logitech USB webcam

Specifications: Wide angle lens, 1.3 Mega Pixels, 640 x 480 resolutions video at 30 fps

An economical webcam was purchased, wide-angle web camera for the project to intentionally introduce the problem of noise. The camera is positioned directly above the center of the worktable. The wide angle is necessary for capturing

the entire table surface. Although the camera captures video up to 30 fps, the limited processing speed of the lab computer only enables us to capture video at approximately 20 fps.

2.5 Matlab

There are many image acquisition softwares but Matlab's Image Acquisition Toolbox was used to write the preferred routine. This allowed adjustments to be made on images as seen fit. After the images are acquired successfully and ready for processing, Matlab's Image Processing Toolbox was used to perform linear filtering on them, a routine to handle the task was again written. Finally, when sufficient data from each image was captured, Matlab's the Statistics and Machine Learning Toolbox and Computer Vision Toolbox was used for feature extraction. All this will be discussed in further details in the next few sections.

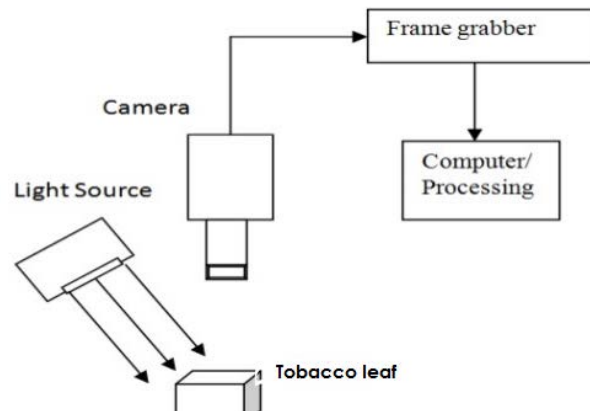


Figure 1. General setup of the hardware equipment.

3. Results and Discussions

Five groups of flue cured tobacco leaves were chosen and each group was allocated a simple group name which corresponded to the class and quality of the tobacco in that group. This however was done only for the purpose of the research hence the group names does not present standard auction floor tobacco classes. In each group, 25 images were captured using the camera and was loading into a folder with a corresponding group name. The images captured were then sequentially numbered from 1 to 25 by means of renaming the images in each folder.

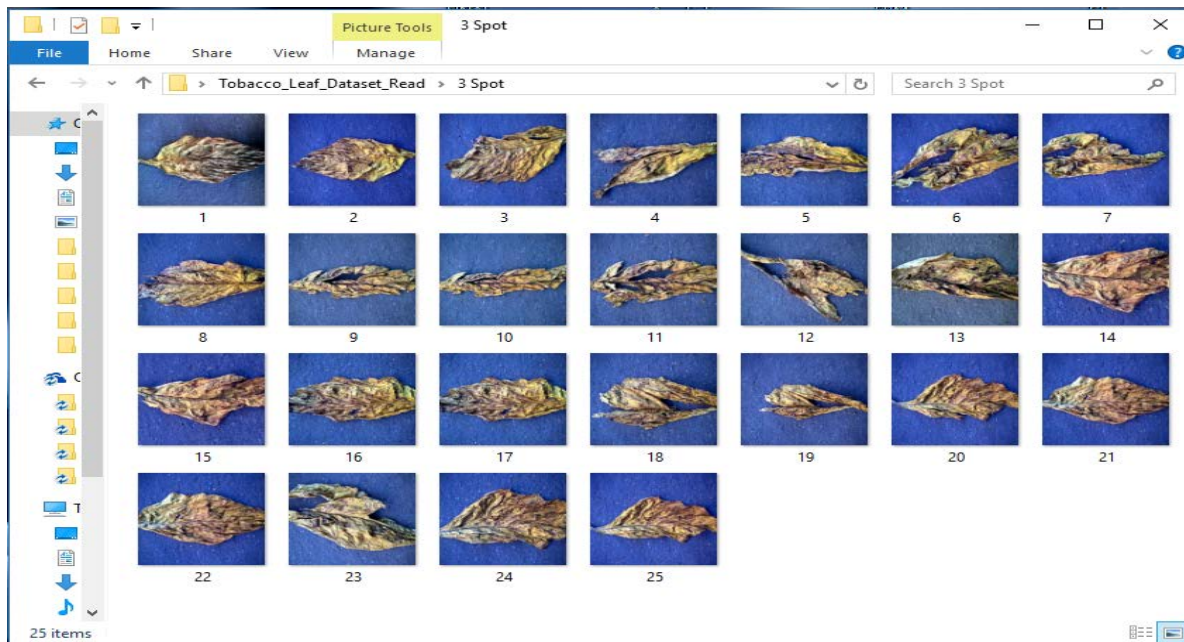


Figure 2. Sequential numbering of images in each folder.

3.1 Design workflow

The general workflow of the intended approach is shown in figure 3, the mathematical modelling of each stage is detailed below.

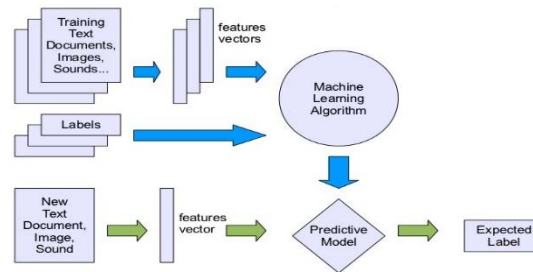


Figure 3. Design workflow layout.

3.2 Mathematical models of classification features.

There are over 190 features that can be extracted from an image but to distinctively separate images some features become more important than others. Mathematical models of some of the more important features are illustrated and these mathematical models were converted into an algorithm in Matlab.

3.2.1 HSV Histogram

The HSV stands for the Hue, Saturation, and Value based on the artists (Tint, Shade, and Tone). The coordinate system in a hexagon in Figure a view of the HSV color model. The Value represents intensity of a color, which is decoupled from the color information in the represented image. The hue and saturation components are intimately related to the way human eye perceives color resulting in image processing algorithms with physiological basis. As hue varies from 0 to 1.0, the corresponding colors vary from red, through yellow, green, cyan, blue, and magenta, back to red, so that there are actually red values both at 0 and 1.0. As saturation varies from 0 to 1.0, the corresponding colors (hues) vary from unsaturated (shades of gray) to fully saturated (no white component). As value, or brightness, varies from 0 to 1.0, the corresponding colors become increasingly brighter (Jeong, 2001).

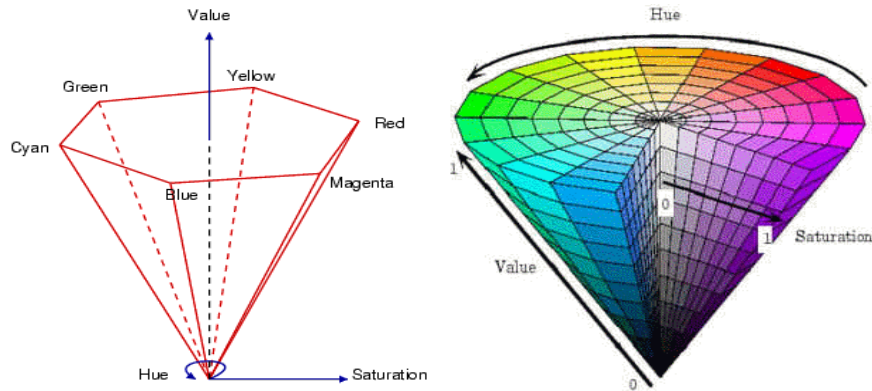


Figure 4. Left HSV coordinate system, right HSV color space

The HSV is developed from the RGB color space by the formulae below:

$$H = \cos^{-1} \left\{ \frac{\frac{1}{2}[(R-G) + (R-B)]}{\sqrt{(R-G)^2 + (R-B)(G-B)}} \right\}, \quad S = 1 - \frac{3}{R+G+B} [\min(R, G, B)] \quad \text{and} \quad V = \frac{1}{3} (R + G + B)$$

3.2.2 Color histogram definition

An image histogram refers to the probability mass function of the image intensities. This is extended for color images to capture the joint probabilities of the intensities of the three color channels. (Jeong, 2001) More formally, the color histogram is defined by,

$$h_{A,B,C}(a,b,c) = N \cdot \text{Prob}(A = a, B = b, C = c)$$

where A , B and C represent the three color channels (R,G,B or H,S,V) and N is the number of pixels in the image. Computationally, the color histogram is formed by discretizing the colors within an image and counting the number of pixels of each color (Jeong, 2001).

3.2.3 Color Moments.

Color moments are measures that can be used differentiate images based on their features of color. Once calculated, these moments provide a measurement for color similarity between images. These values of similarity can then be compared to the values of images indexed in a database for tasks like image retrieval (Keen, 2005). The basis of color moments lays in the assumption that the distribution of color in an image can be interpreted as a probability distribution. Probability distributions are characterized by a number of unique moments (e.g. Normal distributions are differentiated by their mean and variance). It therefore follows that if the color in an image follows a certain probability distribution, the moments of that distribution can then be used as features to identify that image based on color. (Keen, 2005). The color moment are then calculated as follows:

$$E_i = \sum_{j=1}^i \frac{1}{N} p_{ij}$$

Moment 1-Mean,

$$\text{Moment 2-Standard Deviation, } \sigma_i = \sqrt{\left(\frac{1}{N} \sum_{j=1}^i (p_{ij} - E_i)^2 \right)}$$

$$s_i = \sqrt[3]{\left(\frac{1}{N} \sum_{j=1}^i (p_{ij} - E_i)^3 \right)}$$

Moment 3-Skewness,

Skewness can be understood as a measure of the degree of asymmetry in the distribution (Keen, 2005)

3.2.4 Mean amplitude and energy.

The mean amplitude was extracted using the gabor filter shown below.

$$g(x, y; \lambda, \theta, \psi, \sigma, \gamma) = \exp \left(-\frac{x'^2 + \gamma^2 y'^2}{2\sigma^2} \right) \exp \left(i \left(2\pi \frac{x'}{\lambda} + \psi \right) \right)$$

In this equation, λ represents the wavelength of the sinusoidal factor, θ represents the orientation of the normal to the parallel stripes of a Gabor function ψ is the phase offset, σ is the sigma/standard deviation of the Gaussian envelope and γ is the spatial aspect ratio, and specifies the ellipticity of the support of the Gabor function.

3.2.5 Wavelet moments

Wavelet transform has get a wide range of applications in the field of signal processing. Wavelet analysis has good ability of local information analysis but the invariance, Combination multi-scale wavelet analysis and invariant moment is wavelet moment. It is able to get not only global characteristics of the image but also the local characteristics. It also have been widely used in the image target recognition.

The wavelet transform is obtained from a gray scale image. Mathematically, it is obtained by taking the mean coefficient and standard coefficient and creating a feature vector based on these. The feature vector is extracted from the gray level concurrence matrix (GLCM).

$$C_{\Delta x, \Delta y}(i, j) = \sum_{p=1}^n \sum_{q=1}^m \begin{cases} 1, & \text{if } I(p, q) = i \text{ and } I(p + \Delta x, q + \Delta y) = j \\ 0, & \text{otherwise} \end{cases}$$

Where i and j are the image intensity values of the image, p and q are the spatial positions in the image I and the offset $(\Delta x, \Delta y)$ depends on the direction used θ and the distance at which the matrix is computed d . The 'value' of the image originally referred to the grayscale value of the specified pixel, but could be anything, from a binary on/off value to 32-bit color and beyond. Note that 32-bit color will yield a 256×256 co-occurrence matrix!

A gray level concurrence matrix (GLCM) has a maximal of 256×256 which is why all the images were resized. This GLCM is a pillar stone for extraction of most key features. After the resizing a contrast enhancement was done to all the images, because of the need to remove all the filtering done by the camera, in order to restore the image to its true sense.

3.3 Support Vector Machines

Support vector machines (SVMs, also support vector networks) are supervised learning models with associated learning algorithms that analyze data used for classification and regression analysis. Given a set of training examples, each marked for belonging to one of two categories, a SVM training algorithm builds a model that assigns new examples into one category or the other, making it a non-probabilistic binary linear classifier. A SVM model is a representation of the examples as points in space, mapped so that the examples of the separate categories are divided by a clear gap that is as wide as possible. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall on.

In addition to performing linear classification, SVMs can efficiently perform a non-linear classification using what is called the kernel trick, implicitly mapping their inputs into high-dimensional feature spaces. A support vector machine constructs a hyperplane or set of hyperplanes in a high- or infinite-dimensional space, which can be used for classification, regression, or other tasks. Intuitively, a good separation is achieved by the hyperplane that has the largest distance to the nearest training-data point of any class (so-called functional margin), since in general the larger the margin the lower the generalization error of the classifier. More information on **SVM** is found in the Matlab R2015 documentation, in the Statistical and Machine Learning toolbox.

Support vector machines in Matlab

A multi-class SVM was used the general steps when using a SVM is outlined below.

Using Support Vector Machines

As with any supervised learning model, you first train a support vector machine, and then cross validate the classifier. Use the trained machine to classify (predict) new data. In addition, to obtain satisfactory predictive accuracy, you can use various SVM kernel functions, and you must tune the parameters of the kernel functions.

1 Training a SVM Classifier

Train, and optionally cross validate, an SVM classifier using `fitcsvm`. The most common syntax is:
`SVMModel = multisvm(X,Y,'KernelFunction','rbf','Standardize',true,'ClassNames',{'negClass','posClass'});`

The inputs are:

X — Matrix of predictor data, where each row is one observation, and each column is one predictor.

Y — Array of class labels with each row corresponding to the value of the corresponding row in X . Y can be a character array, categorical, logical or numeric vector, or vector cell array of strings. Column vector with each row corresponding to the value of the corresponding row in X . Y can be a categorical or character array, logical or numeric vector, or cell array of strings.

2 Classifying New Data with an SVM Classifier

Classify new data using `predict`. The syntax for classifying new data using a trained SVM classifier (`SVMModel`) is:

```
[label,score] = predict (SVMModel,newX);
```

The resulting vector, `label`, represents the classification of each row in X . `score` is an n -by-2 matrix of soft scores. Each row corresponds to a row in X , which is a new observation. The first column contains the scores for the observations being classified in the negative class, and the second column contains the scores observations being classified in the positive class. (Mathworks, 2015).

3 Tuning an SVM Classifier

Try tuning parameters of your classifier according to this scheme:

- Pass the data to `fitsvm`, and set the name-value pair arguments 'KernelScale','auto'. Suppose that the trained SVM model is called `SVMModel`. The software uses a heuristic procedure to select the kernel scale. The heuristic procedure uses subsampling. Therefore, to reproduce results, set a random number seed using `rng` before training the classifier.
- Cross validate the classifier by passing it to `crossval`. By default, the software conducts 10-fold cross validation.
- Pass the cross-validated SVM model to `kFoldLoss` to estimate and retain the classification error.
- Retrain the SVM classifier, but adjust the 'KernelScale' and 'BoxConstraint' name-value pair arguments.

3.4 Real interface

An interface was designed simplify the visualization to a laymen and figure 5 below shows the user interface of the expert system with all the algorithms running in background.



Figure 5. System user interface (UI)

The model was designed to determine its classification accuracy using 500 iteration. This is a fair number of iteration considering the size of the model database. Using this, 93% of accuracy was achieved from untrained images captured in sunlight illumination and about 90% accuracy of captured in an artificial illumination. The database used was relatively small which means a bigger database will the accuracy shots up as training data base gets bigger.

4. Recommendations and conclusion

Apart from the fact that much of the raw data used in the research was at farm level, pleasing results were obtained. All objectives were achieved and exceeded performance expectations. We therefore conclude that automating flue cure grading of tobacco is very feasible and if funded will impact our tobacco industry in a positive way.

Table 1. A desired table showing the expected tobacco features

Desired component	Explanation
Capture cured tobacco leaf data (Shape, Color and Texture features) systematically.	Systematically captured about 190 flue cured tobacco leaf features(Shape, Color and Texture) of each leaf and stored the data in a database file
Use machine vision technologies to grade cured tobacco leaves.	Successfully used digital imaging and machine learning to classify flue cured tobacco leaves
Achieve process efficiency greater than 76% for untrained data	A process efficiency of 93% was achieved

References

- Chincholkar, A. A., Kite, M. A. U. & Kowale, M. T. O. DIP and Neuro-Fuzzy Based Quality Categorization of Food: A Review.
- Devlin, K. & Lorden, G. 2007. *The numbers behind NUMB3RS: Solving crime with mathematics*, Penguin.
- Ehin, C. 2004. *Hidden assets: harnessing the power of informal networks*, Springer Science & Business Media.
- Jain, A. K., Ross, A. & Prabhakar, S. 2004. An introduction to biometric recognition. *IEEE Transactions on circuits and systems for video technology*, 14, 4-20.
- Livingston, L. J. 1989. Process for treating tobacco. Google Patents.
- Munson, D. R. & Paulson, D. S. 2003. Sharing the Collective Wisdom of Seasoned Practitioners: An Under Utilized Source of Practical Knowledge.
- Patrick-Burns, J. 2014. "The Only Tobacco Story Left in Town": The Interpretation of the Tobacco Industry in Durham, North Carolina. *NCSU Graduate Journal of History*, 2.
- Shappell, S. A. & Wiegman, D. A. 2003. A human error analysis of general aviation controlled flight into terrain accidents occurring between 1990-1998. DTIC Document.
- Shappell, S. A. & Wiegmann, D. A. 2012. *A human error approach to aviation accident analysis: The human factors analysis and classification system*, Ashgate Publishing, Ltd.
- Stern, N., Green, F., Boyd, R. & Finighan, R. 2014. Innovation, risk and government: perspectives and principles from the social sciences.
- Tanaka, K., Ikeda, T. & Wang, H. O. 1998. Fuzzy regulators and fuzzy observers: relaxed stability conditions and LMI-based designs. *IEEE Transactions on fuzzy systems*, 6, 250-265.
- Zhang, F. & Zhang, X. 2011. Classification and quality evaluation of tobacco leaves based on image processing and fuzzy comprehensive evaluation. *Sensors*, 11, 2369-2384.

Biography

Tawanda Mushiri is a PhD student at the University of Johannesburg in the field of fuzzy logic systems and maintenance, is a Lecturer at the University of Zimbabwe teaching Machine Dynamics, Solid Mechanics and Machine Design. His research activities and interests are in Artificial intelligence, Automation, Design and Maintenance engineering Contacted at tawanda.mushiri@gmail.com

Dennis M. Mupatsi is a Mechanical Engineering student at the University of Zimbabwe (2016). Contacted at gbmashana@gmail.com

Charles Mbohwa is currently a Full Professor of Sustainability Engineering and Engineering Management at the University of Johannesburg, South Africa. Contacted at cmbohwa@uj.ac.za