

Texture Classification with Neural Networks

William Raveane and María Angélica González Arrieta

Universidad de Salamanca, Salamanca, España
{raveane, angelica}@usal.es

Abstract. Texture classification poses a well known difficulty within computer vision systems. This paper reviews a method for image segmentation based on the classification of textures using artificial neural networks. The supervised machine learning system developed here is able to recognize and distinguish among multiple feature regions within one or more photographs, where areas of interest are characterized by the various patterns of color and shape they exhibit. The use of an enhancement filter to reduce sensitivity to illumination and orientation changes in images is explored, as well as various post-processing techniques to improve the classification results based on context grouping. Various applications of the system are examined, including the geographical segmentation of satellite images and a brief overview of the model's performance when employed on a real time video stream.

1 Introduction

Texture classification is a specialized area within the field of pattern recognition. As such, other pattern classification techniques have successfully been applied in the past to this problem, such as statistical analysis [7], stochastic algorithms [4], geometric methods [1], and signal processing [8]. Each of these methods has its own advantages and they may be more or less appropriate depending on the particular scenario they are applied to.

Textures, in the sense portrayed in this paper, refer to patterns of colors and shapes formed by pixels in a digital image. Recognizing and distinguishing these textures tends to be a very complicated task, as small variations in scene illumination and view perspective can lead to drastic differences in the visual appearance of a texture, making it difficult for automated systems to successfully segregate them. However, the applications for such a system are many and diverse, making this an important research topic in computer vision. Some of the most typical uses of these texture analysis algorithms are the segmentation of aerial imagery, industrial surface inspection, biomedical image analysis, as well as the classification of textiles, minerals and even wood species.

In this paper, a different method is reviewed based on artificial neural networks. A simplified version of this method using a trivial neural network has already been proposed in the past [5]. This paper proposes a better implementation of this procedure involving additional steps dealing with image processing, neural network preparation and automated results correction. This allows for the successful application of the model on a much wider array of image types, and more importantly, to better generalize on new images that the system has not been trained with.

2 Classification Model

The proposed model is based on a supervised neural network. This section reviews each of the steps involved in its methodology, and then presents how the final system can be used to classify pre-defined texture classes.

2.1 Data Preparation

The application of a neural network to data that originates from photographic imagery renders the preparation step even more significant, as special care must be applied with many of the issues surrounding the handling of visual information in machine learning tasks.

It is customary to first submit images to an enhancement filter, for which there exist many options. Some of the best results observed are consistently obtained by a range of specialized processes such as the Retinex filter developed by NASA for the boosting of detail in satellite and aerial photography [6]. This filter enhances color information in the resulting image and normalizes the variations in illumination adaptively and locally. The neural network results are greatly improved when images are pre-processed in this fashion as it does not need to learn redundant lighting variations found within the shadows or highlights of the image, but can instead focus on the inherent image characteristics.

Alternatively, a histogram equalization process can also be applied independently on all three color channels in the image. This ensures that each data channel fed to the neural network is properly normalized and data is evenly distributed throughout the intensity level spectrum, a process akin to normalizing and re-distributing input data as done in most machine learning tasks.

Figure 1 compares both pre-processing methods and their resulting level histograms in each color channel.

To prepare the neural network system, several regions of interest in the input image are defined, each of which is formed by a number of pixels within a bounded area. A desired label is designated for each of these regions that will identify it as one of the target texture classes. This area of pixels should ideally form an adequate data sampling region including as much variety of the target texture as possible – as the pixels in these regions will become the training data for the neural network, and their labels the ideal output data. It is not necessary for a region of interest to outline a single continuous area, and in fact, it is often desirable to extract multiple regions

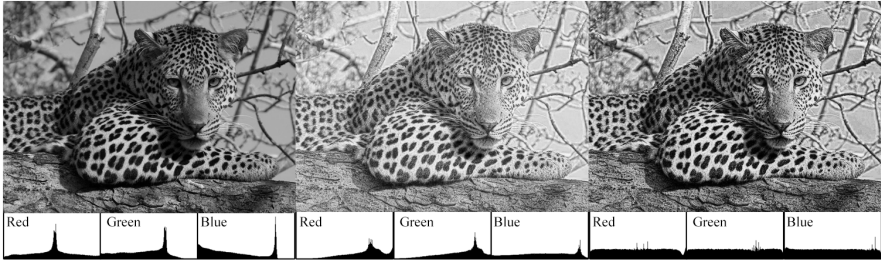


Fig. 1 Comparison of the original sample image (left), the result of the Retinex filter applied (middle), and the result of histogram equalization (right), along with their respective RGB histograms (bottom)

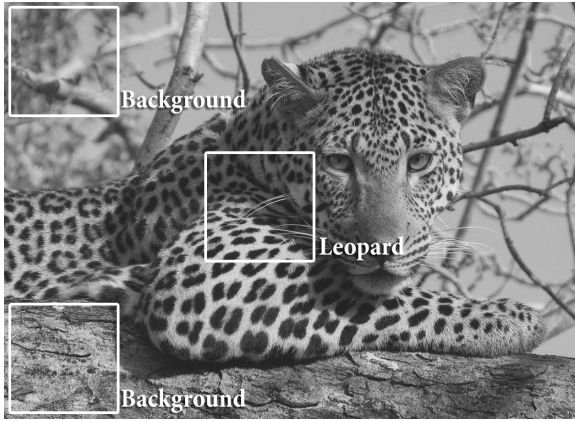


Fig. 2 Regions of interest defining the class labels for various textures in the input image

to provide a larger variation in the training data and in so doing, guarantee better generalization of the model. Figure 2 shows an example of the labeling process.

Once the image is ready for use and its target texture classes have been declared, the regions of interest extracted can be converted to data suitable for the neural network to train with. This is achieved by further subdividing these areas into a grid consisting of 5×5 pixels, thereby creating tiles of 25 pixels each. Figure 3 shows this process in detail.

The text labels assigned to each data sample must be converted to numerical values that the neural network can make sense of within a mathematical context. This is achieved by replacing each string label with an output vector of the same dimensionality as the total number of texture labels that have been assigned. Each output vector will then be an encoded representation that uniquely identifies each of the designated texture classes. With this end, a mapping mechanism is introduced that employs the following label replacement assignments:

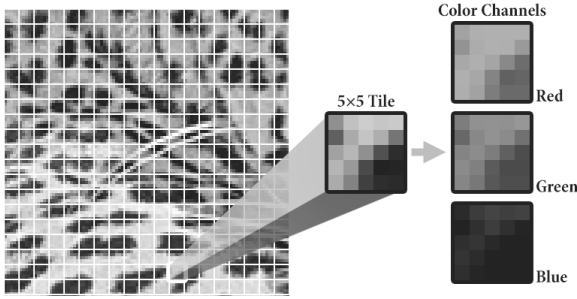


Fig. 3 Detail of a region of interest in the input image, subdivided into a 5×5 grid producing the sampling tiles for that particular texture; where for each tile, the collection of its 25 pixel values in each of the three color channels yields the \mathbb{R}^{75} input vector for a training sample

$$label_0 = \begin{bmatrix} +1 \\ -1 \\ -1 \\ \vdots \\ -1 \end{bmatrix} \quad label_1 = \begin{bmatrix} -1 \\ +1 \\ -1 \\ \vdots \\ -1 \end{bmatrix} \quad \dots \quad label_{n-1} = \begin{bmatrix} -1 \\ -1 \\ -1 \\ \vdots \\ +1 \end{bmatrix} \quad (1)$$

2.2 Texture Classification

The neural network used is a simple three layer feed forward network. Although it is one of the most basic configurations available, it yields very good classification results for the purpose at hand and in return it is capable of training within an acceptable amount of time. The neural network layer structure is arranged as follows:

1. The input layer, with one neuron for each attribute in the input sampling data. Having 75 input neurons, this layer accomodates an input sampling vector in \mathbb{R}^{75} consisting of the three color channel values for each of the 25 pixels in every sampling tile.
2. The hidden layer, with half as many neurons as in the input layer, which is a good rule of thumb to follow in configurations where the amount of samples considerably outnumbers the amount of attributes.
3. The output layer, with one neuron for each texture being classified. Each neuron in this layer corresponds to one of the class labels defined.

Once all of the training data has been laid out, the neural network is trained with it, using a learning algorithm such as Resilient-backpropagation (RPROP) [2], which in our tests has resulted as one of the most optimal methods when compared to other traditional algorithms such as simple Backpropagation.

Upon completion of the training process, the system can be used to classify the texture regions in an image. For this procedure, it is first necessary to subdivide the entire image into a similar 5×5 grid, and extract all of the individual tiles outlined by this grid. As before, each tile becomes a new sampling point that can be expressed

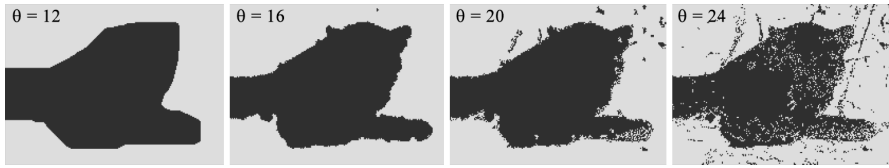


Fig. 4 The effect of varying θ in the final results, showing $\theta = 12, 16, 20$ and 24 respectively

as a vector in \mathbb{R}^{75} . The values for each sample tile in the testing image grid are sequentially fed to the trained neural network, and the output generated by each is then the predicted texture at that region. This process is repeated for every tile in the image, until all areas in the image have been classified.

For better results, a final post-processing procedure is applied to the resulting classified tiles. This consists of a simple nearest-neighbor search and replace process which seeks to correct any outlier tiles that usually represent misclassified data. Each tile is tested and compared against its neighbors up to a distance of 2 tiles apart, that is with the closest 24 neighboring tiles. A parameter θ is introduced as a threshold, where if the number of neighbors with a label different than that of the currently tested tile exceeds this parameter θ , then the current tile’s label is changed to match that of the surrounding majority. This is analogous to a smoothing procedure as is usually applied at the end of many image processing algorithms to alleviate noise introduced by the system. Figure 4 shows the result of varying θ in the classification results.

3 Results

A comparison of the results obtained on the sample image when three different algorithms are applied is shown in Fig. 5. The first of these is a well known image processing algorithm known as Chan-Vese segmentation [3]. Although this algorithm has been widely proven to be successful on images with a clear subject and background distinction, it is evident that in more complex images such as the one presented here, it fails to distinguish the textures, instead focusing on local variations such as the spots in the leopard’s skin, while being quite sensitive to shadows.

The following result shown is that of the non-optimized neural network solution proposed by Natarajan et al.[5]. The results in this case are much closer to what would be expected of a texture classifier. Although it is able to distinguish the sky portion of the background properly, it is not capable of correctly segregating the branch region of the background.

The final image shows the result of applying the system proposed in this paper, where the silhouette of the leopard is fully distinguishable from the background, with very little noise remaining in the final classification.

The neural network can also be applied to other images having similar texture characteristics. Figure 6 shows an example of a continuous video stream where only the first frame of the stream was used to train the neural network model. Yet, the

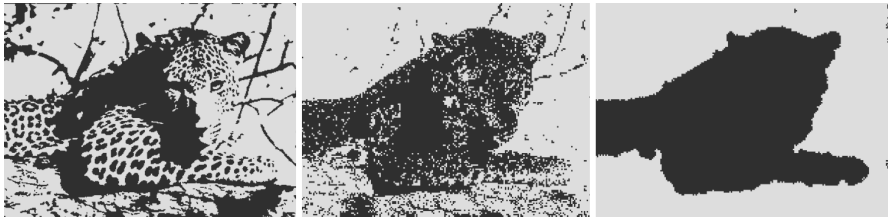


Fig. 5 Comparison of various systems, Chan-Vese segmentation [3] (left), a non-optimized neural network system [5] (middle), and the procedure proposed in this paper (right)

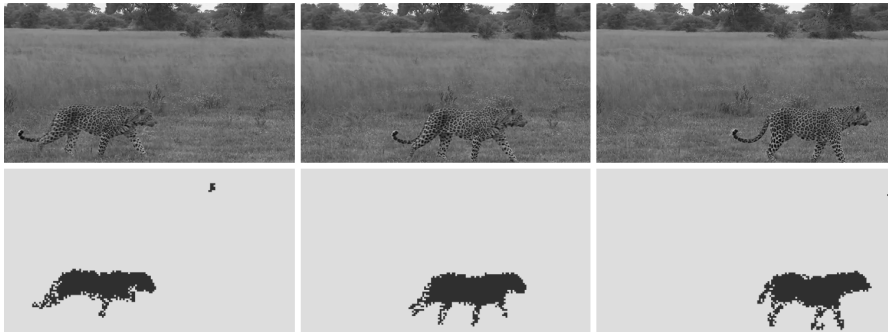


Fig. 6 Multiple frames of a video stream classified with a neural network model trained on a single frame of the video

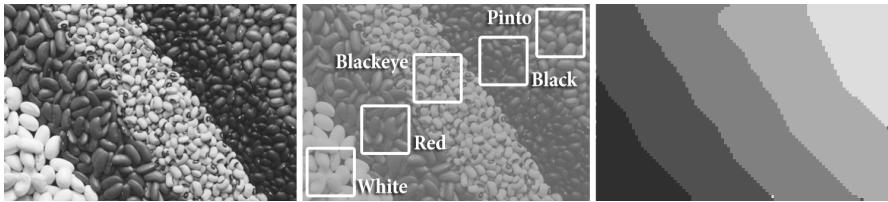


Fig. 7 A single image with multiple texture labels classified by the proposed system

system is able to generalize the classification results to the rest of the frames in the video, properly identifying the walking leopard and following it throughout the rest of the video sequence.

The system can be extended to classify multiple textures by defining additional labeled classes in the training set, as can be seen in the example results of Fig. 7. Here, multiple regions in the image are defined for each of the textures of interest, each one with its own unique class label.

A common application for texture classification is in the topographical study of aerial photography. The proposed system can be successfully applied to this field as

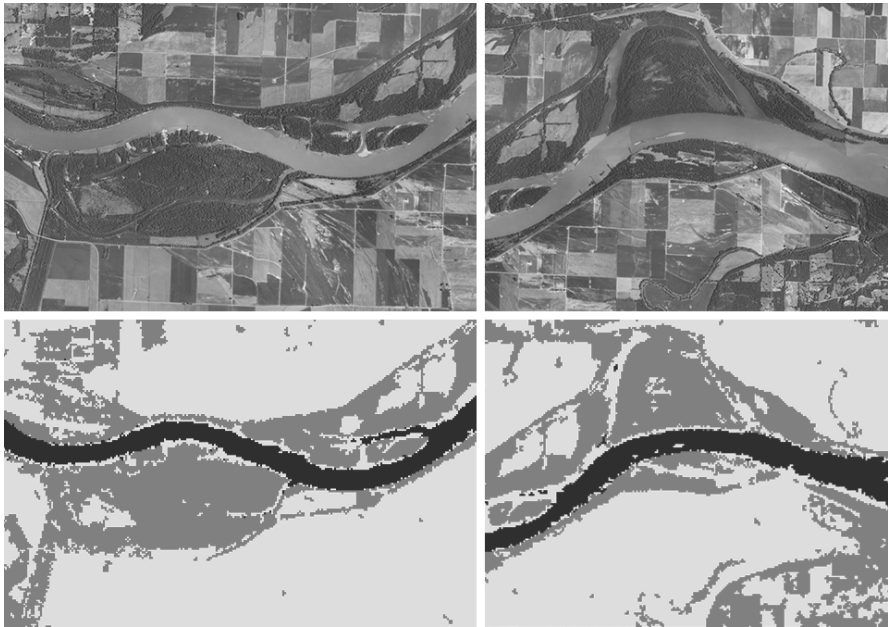


Fig. 8 An example of the system trained only on the left image for the classification of topographical terrain on satellite imagery of the Missouri River, where three types of surfaces are recognized by the model: croplands, woodlands, and the river surface

shown in Fig. 8. Here, the model is able to classify surface types by learning features in the various terrain types, even when such features may be complex and non-repeating such as in the segmentation of croplands shown in the example images.

4 Conclusions and Future Work

A new method was proposed for the classification of textures in an image through a supervised neural network model. The various steps involved in preparing and processing the image were described, and some results and applications of such a system were reviewed.

The current method gives reasonable results in a wide array of sample images, but improvements to the system’s robustness can always be improved upon. In particular, enhancements to the context representation of neighboring tiles for better classification consistency would be of great benefit to this technique. Additionally, this neural network model could be a stepping stone to building an unsupervised classification system where no textures or labels are defined beforehand – but instead regions would automatically be clustered together by their features in a manner not unlike current color segmentation algorithms, but with all the improvements described here.

References

1. Chen, Y.Q., Nixon, M.S., Thomas, D.W.: Statistical geometrical features for texture classification. *Pattern Recognition* 28(4), 537–552 (1995)
2. Encog Online Documentation: Resilient Propagation. Heaton Research, <http://www.heatonresearch.com/wiki/>
3. Getreuer, P.: Chan-Vese Segmentation. *Image Processing On Line* (2012)
4. Liu, L., Fieguth, P.: Texture Classification from Random Features. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 34(3), 574–586 (2012)
5. Natarajan, K., Subramanian, V.: Texture Classification, Using Neural Networks to Differentiate a Leopard from its Background. *Science Applications for NeuroDimension*, <http://www.nd.com/apps/science.html>
6. Rahman, Z.U., Jobson, D.J., Woodell, G.A.: Retinex processing for automatic image enhancement. *Journal of Electronic Imaging* 13(1), 100–110 (2004)
7. Varma, M., Zisserman, A.: A statistical approach to texture classification from single images. *International Journal of Computer Vision* 62(1), 61–81 (2005)
8. Yu, G., Slotine, J.J.: Fast Wavelet-Based Visual Classification. In: *Proc. IEEE International Conference on Pattern Recognition, Tampa* (2008)