TIRE TYPE RECOGNITION THROUGH TREADS PATTERN RECOGNITION AND DOT CODE OCR

Tasneem Wahdan, Gheith A. Abandah, Alia Seyam, Alaa Awwad
Computer Engineering Department
The University of Jordan
Amman 11942, Jordan
tasneem.90.wahdan@gmail.com, abandah@ju.edu.jo, alia.seyam@gmail.com, alaa.awwad.90@gmail.com

Islam Shdaifat
JoVision Image Processing Pioneers
Hamburg, Germany
i.shdaifat@jo-vision.com

ABSTRACT

Tire text and geometry detection are used in scanning tires in order to identify the tire type and classify them according to a code printed on the sidewall which is called "DOT Code" then checking the treads whether they are defected or not based on the pattern periodicity of them. Classification is done based on applying some image processing and pattern recognition techniques on both treads plus DOT code samples of the tire. Tire classification that works based on treads is more accurate than classification based on DOT code, but using them together results in a powerful and accurate classification process.

Keywords: Tire Type Recognition, Tire DOT Code, Tire Treads, OCR.

1 INTRODUCTION

Any tire consists of two main parts: treads, which come in contact with the road, and sidewalls that contain many information about the tire such as manufacturer name, radial, etc. The DOT code is a part of information printed on the sidewall. The DOT code as shown in Fig. 1 begins with the word "DOT" followed by codes that specify company, factory, mold, batch, and date of production. The last four digits represent the week and year when the tire was made. To manually classify different types of tires it takes a lot of time and effort. So, this paper introduces a solution that can automate the process of classification. Tire text classification has to be developed so that it can classify the tire according to its DOT code and/or treads shape.

1.1 Related Work

The research related to this problem is still in its early stages, hence, no mature solutions are available yet, and available knowledge in this field is not coherent enough to form strong basis. The main paper that was found in this field is titled "Pattern Recognition for Classification and Matching of Car Tires" [1]. The following is a description of its work. This paper is interested in whether a given tread pattern matches an existing tire; they saw that the only way at that moment is to compare it manually and search in a large library of tire tread patterns. So they work to automate this pattern matching using computer vision, image processing and pattern recognition. The first step is to pre-process the given image and convert the digital image of tread pattern to a two-dimensional fast Fourier transform (2D-FFT), and this provides information about the global structure of the tread pattern, from the 2D-FFT they use two methods to extract useful features. The first method which they refer to as PCA (Principle Components Axes), the second method is based on PSA (Power Spectral Analysis). And their prototype matching system uses *k*-nearest neighbor (*k*NN) classifier because this algorithm is simple to be implemented [1].

2 SAMPLES COLLECTION

The prototype machine shown in Fig. 5 was used for taking the samples of both tire treads and sidewall.

Figure 1: DOT code region on tire sidewall image

2.1 The Rotating Part of the Machine

This part is used to rotate the tire to take a full scan of the sidewall and treads of it. The speed of rotation is controllable using tuning button. In addition, this part is designed in a way where we could put tires with different diameters and still being stable while the tire is rotating.

2.2 The Linear Camera and the Computer

Basler linear camera shown in Fig. 2 is connected to the pc via fire wire then configured and checked if it is working well using Pylon SDK software [2]. Fig. 5 shows the entire machine used for taking the samples. The user interface shown in Fig. 6, which is built in c# programming language, was used to acquire tire samples. The camera stand is designed in a way that can be rotated to be able to take samples of both tire sides as shown in Fig. 3 and Fig. 4.

Figure 2: Basler linear camera used for taking samples of tires

2.3 Samples

The samples were taken many times in different conditions to decide the best situation for taking them.

- Taking treads samples in normal room illumination were good enough to be used directly, but these conditions were not good for taking the sidewall samples.
- We tried to change the conditions by using an illumination source, but the sidewall samples did not achieve to the result we wanted.
- After many trials on different tire types, we found that the change of the angle of the illumination source improves the quality of the samples.
- It has to be mentioned here that we were forced to divide each treads full image into smaller samples since we had only one tire for each type used.

Fig. 7 and Fig. 8 show the final used samples of Nankang tire type.

Figure 3: Position of the camera while taking sidewall samples

Figure 4: Position of the camera while taking treads samples

Figure 5: The entire system used for taking samples which contains pc, linear camera and rotating machine

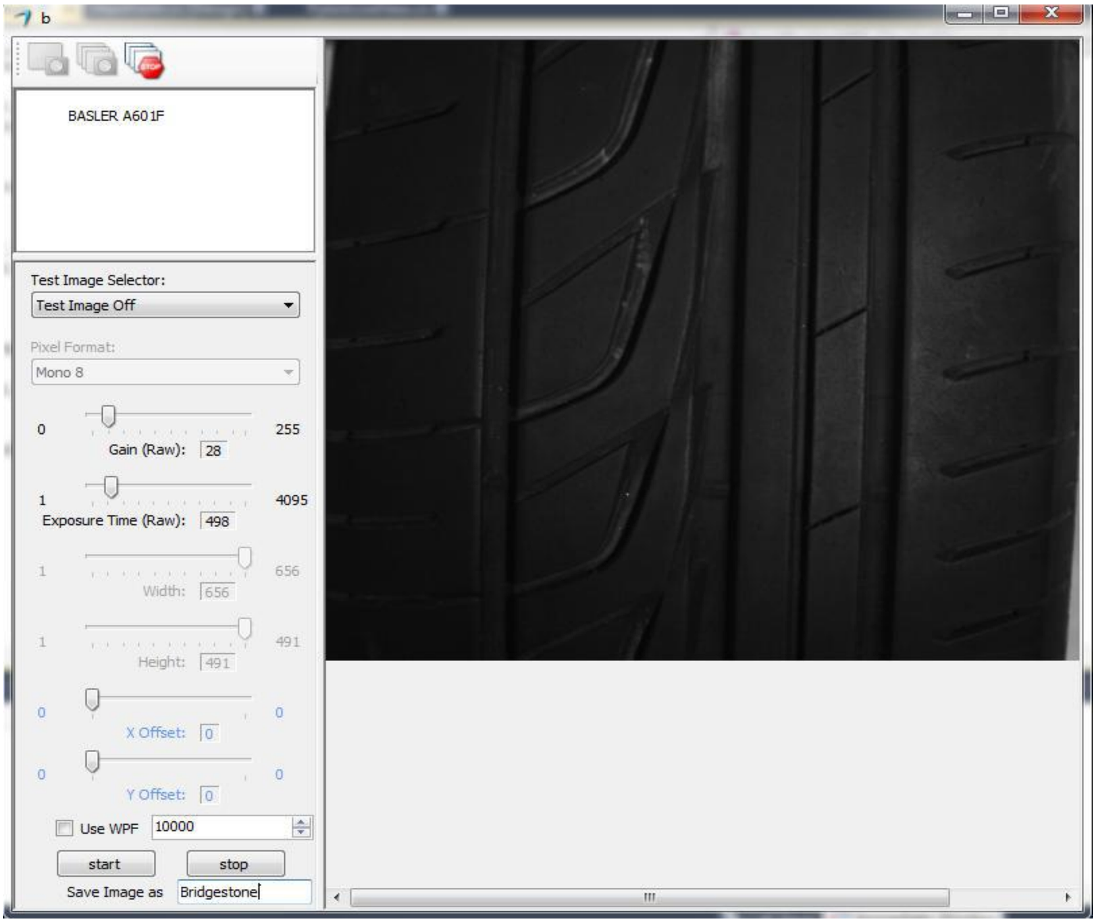


Figure 6: The user interface used for taking samples of tires

Figure 7: Sidewall image of Nankang tire type

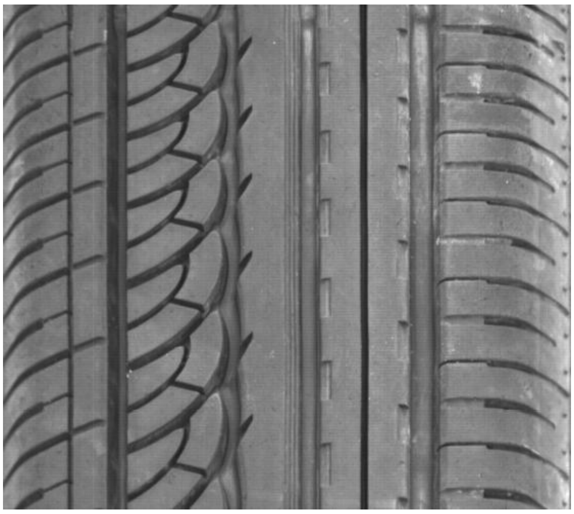


Figure 8: Treads image of Nankang tire type

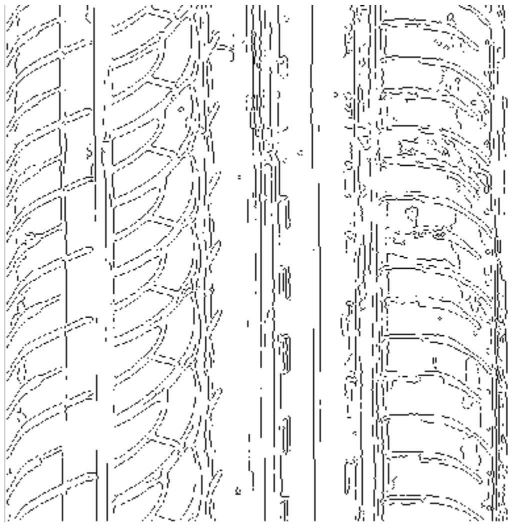


Figure 9: Treads image of Nankang tire type after applying canny edge detector

3 TIRE TYPE DETECTION

Firstly, the tire has to be classified according to its treads shape. The output of the classifier is essential for template matching and DOT Code recognition explained in next section.

3.1 Pre-Processing

Before feature extraction and classification, it was helpful to do smoothing which reduces the noise in the samples by applying average filter. Canny edge detector is then applied. The result of this edge detector is a binary image in which the black pixels closely approximate the true edges of the original image like the binary image shown in Fig. 9.

3.2 Feature Extraction

Many features can be extracted from treads binary image, but not all of them are discriminative. The following two types of features were collected but not all of them were used in classification process.

3.2.1 Discrete cosine transform (DCT)

It is a process which transforms the image from spacial domain into frequency domain. DCT is used to extract thirty six near-centre coefficients that contain the maximum power, as features from each sample. Fig. 10 shows the Image in frequency domain.

3.2.2 Gradient

It is the first derivative. This technique is simply used to detect the angles of lines of tire patterns. First, masks are assigned using matrices that implement the desired line angles. Then, those masks are applied on the edge detected image so that we obtain four images with four different line angles: 0º (horizontal), 45º, 90º (vertical) and 135º. Firstly, we consider the number of black pixels in each image of those four new images as four different features [3]. Then, we found that taking the ratio of the number of black pixels over the total number of pixels in each image makes the four features scale-invariant ones.

Fig. 11 and Fig. 12 show horizontal and vertical edges of Nankang tire treads edge image.

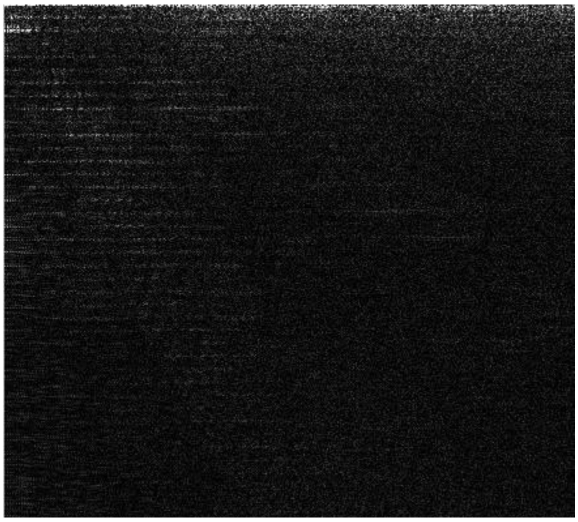


Figure 10: Discrete cosine transform of Nankang tire treads image

3.3 Feature Selection and Tire Type Classification

Four different classifiers are used to be able to achieve the best classification results and the minimum cost functions. We mention here that after analysis of the previous collected features, we found that the gradient features are the most discriminative, so we used them with the following classifiers: k-nearest neighbor, linear classifier, decision tree and neural network. Table 1 shows the average error rate of the four classifiers.

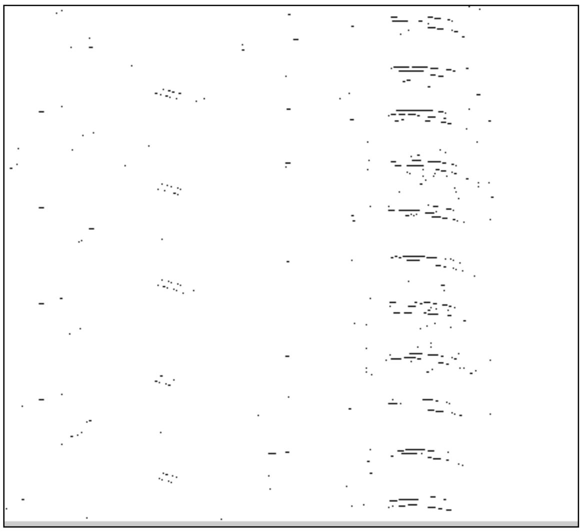


Figure 11: Horizontal edges of Nakang tire treads edge image

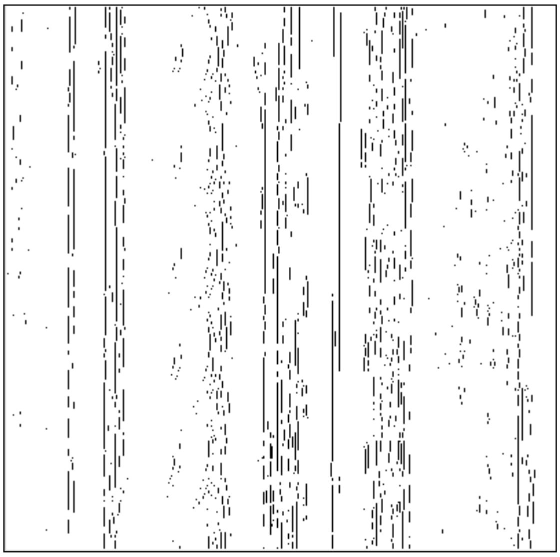


Figure 12: Vertical edges of nankang tire treads edge image

Table 1: The average error rate of the classifiers

Classifier	**Min error rate**	**Number of features used**
Decision Tree	0%	2
Neural Network	0%	4
Linear	0%	4
*k*NN	20%	4

4 DOT CODE RECOGNITION

This section explains the main steps used to recognize DOT code from tire sidewall image.

4.1 Template Matching

Template matching is used to detect the DOT code region in the sidewall image by scanning a template which contains DOT word, as the one shown in Fig. 13, and computing the correlation between the template and the part of the sidewall image that lies under the template. While computing the correlation, it is taken into consideration to find the maximum correlation and its corresponding location or index. The maximum correlation means the most similarity between the template image and the target image, in other words, the index of the maximum correlation found indicates the DOT code region. This method gives accurate results most of the time. However, it is slow because it computes the correlation for a large number of times since the sidewall image of the tire is very large. To reduce the delay of this method, we noticed that the DOT code is always within the middle range of the sidewall image, so we cut the sidewall image to a smaller size one that surely contains the DOT code region using this piece of code in Matlab:

```
[x y]=size(I);
I=I(floor(x/2):floor(x/2)+floor(x/4),:);
```

where I is the sidewall image. This formula means that the DOT code is always within the lower half and the lower half plus fourth of the length of the sidewall image. This formula came after testing and experimental trials. Fig. 14 shows a part of the sidewall image after applying the formula.

Figure 13: The template which contains 'DOT' word that is used to detect DOT code region from tire sidewall image

Figure 14: The result of applying the Matlab code explained above on the entire sidewall image

Since we had limited number of tire types, we thought to reduce the delay and maximize accuracy by using a unique template for each tire type. Choosing the right template depended on the output of the classifier we used before which classified the tire according to its treads. For example, Kumho tires' DOT code always starts with the two characters YO, so we used the template DOT YO that is shown below in Fig. 15 to detect and cut the DOT code region of Kumho tires.

Figure 15: The template used to detect DOT code region of Kumho tire type

Finally, the region of DOT Code is cut based on template matching, correlation, result. So, the output of template matching is an image as the one shown in Fig. 16.

Figure 16: Extracted DOT code region after template matching using template in Fig. 15

4.2 DOT Code Images Preparation for OCR

Most Optical Character Recognition (OCR) engines needs to have clear binary text images as inputs to give accurate OCR results. The most important challenge which it has to be mentioned here is the low intensity transition between the DOT code itself and the background, which is the tire. This makes binarization a very challenging process. After many trials and experimentations we found that using an illumination source makes a big difference in the quality of binarizaton. The following pre-processing steps where suitable to be applied on DOT Code grayscale images before binarization:

- Erode the grayscale DOT code image using 3×3 ball structuring element.
- Filter the eroded image using 5×5 Gaussian filter.

Fig. 17 illustrates applying the previous two steps on Federal tire DOT code. Otsu's method was used for binarization [4]. Fig. 18 shows the binary image accomplished from applying Otsu's method then erosion on the image in Fig. 17 (c).

4.3 OCR of Binary DOT Code Images

The DOT code of any tire contains only capital English letters and numbers. That means the OCR to be used has to have 36 classes which are 26 characters from A to Z plus 10 numbers from 0 to 9. According to this information, we decided to build our own OCR using binary characters' samples accomplished from DOT code images binarization process. We suggest doing segmentation based on vertical projection of DOT Code binary images [5]. Unfortunately, we did not have enough samples for all characters, so building OCR is a part of the future work.

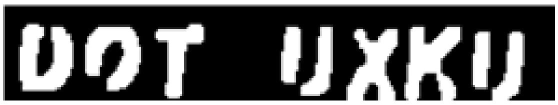
(a) Federal tire DOT code

(b) Image in (a) after erosion

(c) Image in (b) after filtering using gaussian filter

Figure 17: Preprocessing operations applied on DOT code grayscale images

(a) Thresholding result using Otsu's method



(b) Image in (a) after erosion

Figure 18: Thresholding result of DOT code grayscale Image

5 RESULTS AND CONCLUSIONS

A classification method was designed, and precisely described. The accuracy of classification that we achieved based on tire treads was 100%, but we were not able to complete the classification based on DOT code due to lack of characters samples. This high accuracy may be reduced if more tire types are added to the classifier.

We plan for future work to train the classifier on larger sets of tires samples. We will build our own OCR and try to improve classification result by finding more discriminative features. In addition, we will develop a smarter and faster template matching algorithm. Finally, we will try to make the project faster using parallelism techniques.

6 ACKNOWLEDGEMENT

We would like to thank JoVision Company for their support by providing needed equipment and valuable information and suggestions.

7 REFERENCES

[1] D. Colbry, D. Cherba, and J. Luchini: Pattern recognition for classification and matching of car tires, Tire science and Technology, Vol. 33, No. 1, pp. 2-18 (2005)

[2] Basler, Basler website, http://www.baslerweb.com, 21-12-2012, 10:00 pm.

[3] A. Suyyagh and G. Abandah: FPGA parallel recognition engine for handwritten Arabic words, Journal of Signal Processing Systems, pp. 1-8 (2013).

[4] Wikipedia, Otsu’s method, http://en.wikipedia.org/wiki/Otsu's_method, 21-12-2012, 10:00 pm.

[5] H. Bunke and P. Wany: Handbook of character recognition and document image analysis, World Scientific (1997).