Usage of Hough Transform for Expiry Date Extraction via Optical Character Recognition

Davide Scazzoli*, Giulia Bartezzaghi†, Deniz Uysal‡,
Maurizio Magarini*, Marco Melacini†, Marco Marcon*
*Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Italy
Email: {davide.scazzoli,maurizio.magarini,marco.marcon}@polimi.it
†Dipartimento di Ingegneria Gestionale, Politecnico di Milano, Italy Email: giulia.bartezzaghi@osservatori.net
‡Middle East Technical University, Ankara, Turkey. Email: uysal.deniz@metu.edu.tr

Abstract—**In this article we evaluate the impact of using two image pre-processing approaches with the objective of aiding an Optical Character Recognition (OCR) software in correctly retrieving an expiry date from an image of a product containing it. In particular, we analyze the impact of finding the rotation angle of an image using the Hough transform and the impact of image binarization using adaptive Gaussian threshold. We attempt to further increase OCR accuracy through a sliding window approach. Our results show that applying the Hough transform noticeably improves OCR performance with minimal impact on the execution time.**

I. Introduction

The growing number of people living in relative poverty are leading to an increase of food insecurity in Europe. Cities and regions are devising local policies to ensure food security for their inhabitants and promote resilient food systems. Such policies have led to a growth of food recovery initiatives [1]. At the same time, food waste is exerting a pressing challenge in the design of sustainable food systems [2]. Recent studies suggested the relevance of food waste induced emissions, water consumption, land use, and related economic and social impacts, representing the third emitter globally, with an estimated cost of about 940 billion USD [3]. Consequently, food waste is now at the core of several international policy agenda including the UN Sustainable Development Goals and the European plan for a Circular Economy. Within this context it becomes imperative to keep track of the expiration date of food products in order to identify and reduce waste.

New standards of barcode such as the GS1 DataBar, also known as EAN-128, introduce the possibility of including product validity and expiration date within the barcode [4], however, this requires the usage of an extended barcode format on the product. Many retail stores still use Barcodes including only the Global Trade Identification Number (GTIN). It becomes then necessary to acquire the expiry date of a product displayed on a shelf by means other than barcode reading.

The digitization of the expiry date brings many advantages to both donors and beneficiaries. The donors can testify to the validity of their donated material for legal reasons and the beneficiaries are able to know in advance the state of the donated food to better prepare for receiving it and avoid unnecessary waste. This can be achieved by Optical Character Recognition (OCR) software. In this paper we explore the impact of pre-processing on OCR performance.

The rest of the paper is organized as follows. Section II gives a survey of approaches to expiry date recognition present in the literature as well as similar problems. In Sec. III we introduce the Hough transform and other image pre-processing approaches we adopted. In Sec. IV we introduce the algorithms for rotation correction using Hough transform and sliding window approach to Optical Character Recognition (OCR) while, in Sec. V, we comment the results in terms of algorithm execution time and OCR accuracy. Finally, Sec. VII concludes the paper.

II. Related Works

Other works have already attempted the extraction of expiry dates from images of products. In [5] the authors use an approach based on Stretched

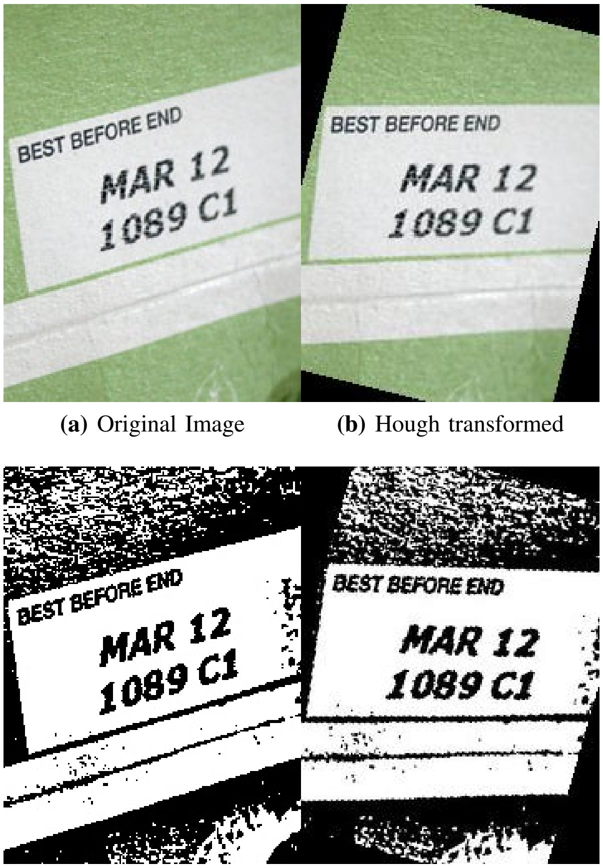
Gabor Feature Extraction and a multi-layer dense neural network for character recognition. The main difference with our work is that the OCR algorithm was designed specifically for expiry dates while we used a generic OCR and studied the impact of image pre-processing techniques to improve its performance. Another approach is shown in [6] where the authors focus on the aspect of localizing the expiry date from images of a product. The authors implement an approach where the image is directly passed through an OCR at 8 different angles and 3 magnification levels. Our approach differs since we use Hough transform to immediately find the rotation angle of text in the picture [7]. The problem of detecting the expiry dates from images shares many similarities with the problem of reading car licence plate from traffic cameras. One interesting approach to this problem is shown in [8] where the authors train a neural network on an artificially enlarged dataset but that work differs greatly from our approach as we take a general purpose OCR software and study the impact of image pre-processing algorithms on its performance. Usage of Hough transform to generally improve OCR performance outside of the specific problem of extracting expiry dates has already been implemented in [9], [10].

III. Image Pre-Processing and Hough Transform

In order to increase OCR accuracy and speed we have considered three methods of image pre-processing:

A. Image Binarization

We have implemented a Gaussian threshold binarization technique to attempt improving the results of the OCR, an example of which can be seen in Fig 1c,1d. This adaptive thresholding method has applied by using OpenCV library on Python [11], however results indicate that in many cases binarization lowers expiry date detection accuracy. By applying binarization on an image the effect of illumination on OCR is decreased and accuracy of OCR slightly increased, however, if the binarization approach fails at correctly identifying a threshold, the expiry date might be entirely obstructed. This is particularly critical for those cases where the expiry date is not printed in black on white color format.



(a) Original Image **(b)** Hough transformed

(c) Binarized Image **(d)** Binarized and Hough transformed image

Fig. 1. Effect of image preprocessing

B. Hough Transform

We have implemented the Hough transform [7] using the OpenCV Python library as shown in [12] to detect and correct rotation angle of text as shown in Fig. 1b. First a high pass Prewitt filter is applied to the image to detect edges, then the Hough transform is performed and the accumulation point with the highest concentration returns the estimated rotation angle of the picture. The transformation consists in transforming the relevant points (belonging to edges) from the (x, y) plane to corresponding sinusoids in the (ρ, θ) dual plane by using equation 1.

$$\rho = x \cos(\theta) + y \sin(\theta) \tag{1}$$

Once x and y are fixed the values of ρ and θ in Eq. (1) represent respectively the distance from the origin and the orientation of all the straight lines belonging to a pencil centered in (x, y). If we consider multiple collinear points in the (x, y) plane, they belong to the same straight line that is then shared among all the pencils associated to these points. This straight line, in the (ρ, θ) plane is then represented as the intersection of all the sinusoids (see eq.(1)) generated by every point of the straight line. In a discretized representation of the (ρ, θ) plane the intersections of different sinusoids are represented by accumulation points. We then search for local maxima of these accumulation cells: since we are interested in the rotation angle estimation we search, in the (ρ, θ) plane, for the horizontal row (i.e. constant θ) where we have the highest number of local maxima independently from the ρ value. The resulting value of θ: $\bar{\theta}$ is the orientation angle of most of edges in the image independently from their relative position. This is done on the assumption that the expiry date itself should follow generally straight lines so, by doing this transform, we are able to identify the dominant rotation angle of the text present in the picture. Once the rotation angle is identified the picture is then rotated to compensate for the rotation, this operation was shown to be very fast and thus has very limited impact on performance when compared to the OCR execution time. In some cases, however, the expiry date may be printed on round objects or at odd angles with respect to package lines. In these cases we could expect a failure of the Hough transform to correctly identify the rotation angle and a more accurate processing must be done accordingly (color text removal, large uniform region removal, etc.).

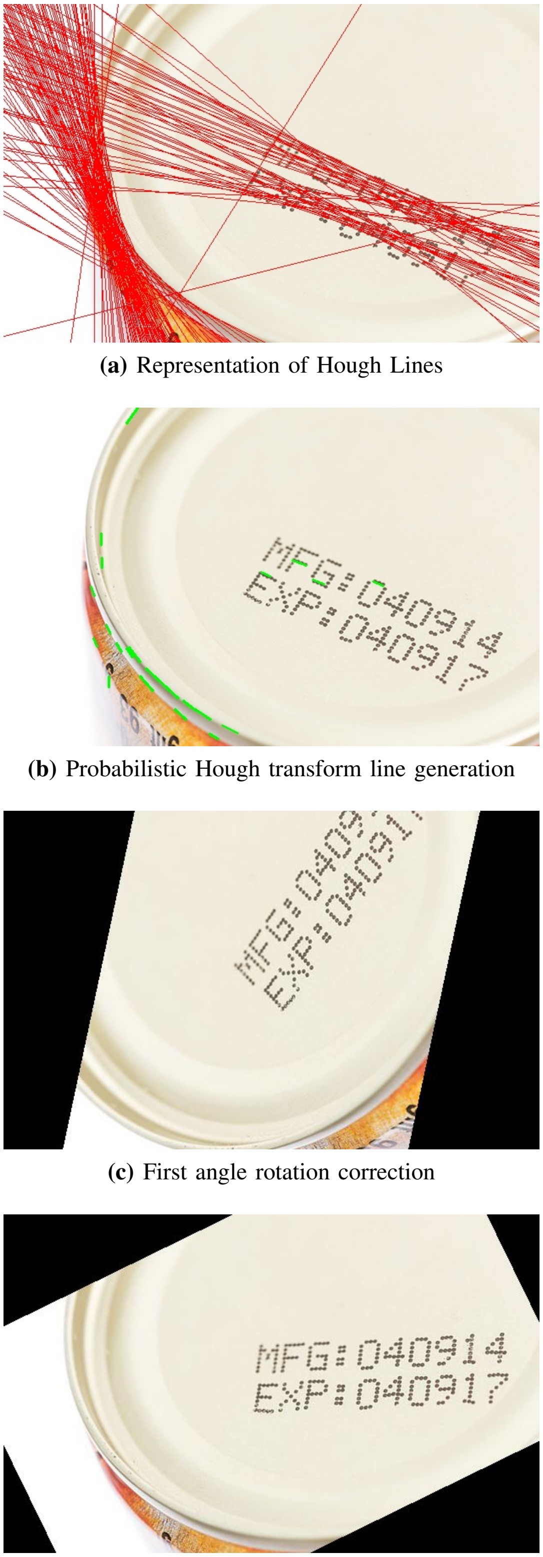
C. *Morphological processing*

In some cases the printed text of the expiry date could be blurred and the characters touching each other. Our approach to solve these cases was to use a morphological opening pre-processing [13] before we give the image as an input to the OCR.

IV. Algorithm Description

A. *Rotation correction*

Correction of the image rotation is performed with algorithm 1 which is similar to the approach of [10]. In the algorithm we first convert the image to grayscale and subsequently apply an high pass filter in order to detect edges. This image is then fed to the algorithm which generates the Hough Transform. When many edge points share the same value of ρ and θ (i.e. the associated sinusoids in the (ρ, θ) plane intersect in the same cell) then they



(a) Representation of Hough Lines

(b) Probabilistic Hough transform line generation

(c) First angle rotation correction

(d) Second angle rotation correction

Fig. 2. Rotation correction using the angles generated by algorithm 1.

Algorithm 1 skew angle identification

```
Convert image to grayscale
apply high pass filter
while Generated HoughLines < 50 do
   Apply Hough Transform
   reduce threshold
end while
for all lines in HoughLines do
   for all angles in VotingVector do
      if $|\theta - VotingAngle|$ < Precision then
         Increment AngleHits
      end if
   end for
end for
```

belong to the same straight line in the x, y space. If more points with these characteristics than the threshold chosen are found then a Hough line is generated.

Successively, a voting vector is formed with enough entries to guarantee the desired accuracy. For each Hough line the relative value of θ is evaluated against the angles in the voting vector and, if it matches, the corresponding entry is increased. With this approach the most likely θ candidate for rotation correction is the one that gives the largest number of Hough lines generated, however, as shown in fig. 2, this is not always the correct angle. The algorithm thus generates as many rotated images as the number of angles $\bar{\theta}$ shared by a high number of Hough lines (As a rule of thumb, the first five rotated images sharing the highest number of Hough lines can be considered). In the results shown in Sec. V all angles with more than one Hough lines were used for rotation correction. In our implementation we considered both full image Hough transform, shown in Fig. 2a, as well as the probabilistic Hough transform, shown in Fig. 2b, which only takes a random subset of points from a full image to generate the Hough lines, in both cases the rest of the algorithm remains unchanged.

B. Sliding Window Technique

The problem of acquiring an expiry date from an image is made difficult by several factors:

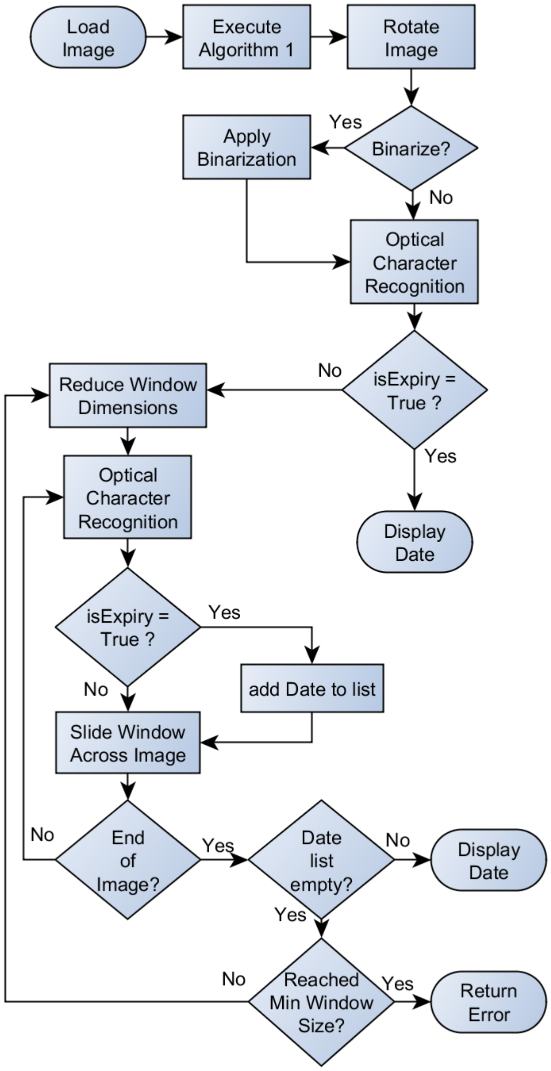


Fig. 3. Sliding Window Algorithm Scheme

- Expiry dates may be printed everywhere on a product, including sharp edges or curved surfaces like bottle caps or deformable plastic bags.
- Expiry dates do not have a single standard format, they may appear as only numeric or have months expressed in letters. In some cases it could be expressed as month/year while in others as day/month. Day month and year may have separator characters between fields or not.
- Expiry dates could be printed with many different fonts and colours.
- To the best of the authors knowledge there is not a repository containing expiration date images to use for testing algorithm or performing machine learning approaches.
- The expiry date is not the only text present on an item but is often surrounded by many kinds

of text, including preparation date which may give false positives.

Because of these limitations, many approaches that we considered could not be adopted.
The first solution we considered was developing an algorithm to locate the expiry date on an image similar to face recognition algorithms. Because of the limitations mentioned above the development of such algorithm would be extremely hard. Another option we considered was to have the user manually assist the algorithm by pointing in the general location of the expiry date. This option was also discarded as it requires user input. At the end we opted for a sliding window approach where a variable size window is passed along the image until the expiry date is correctly extracted.

The algorithm first attempts to read from the image, if it is unsuccessful it shrinks its searching window and starts sliding it across the image. After a certain number of iterations, in our case we fixed this to 10, if the date is still not found the algorithm gives up and returns an error. The full scheme of this algorithm we propose is shown in Fig. 3. after image preprocessing is performed and the OCR fails, the algorithm reduces the dimension of the window, in our case we shrunk it by a factor of 1/N where N is the number of iterations. This approach noticeably extends the amount of time taken to derive a result, however, it significantly increases the chances of correctly acquiring the expiry date. During the process of sliding the window, if an expiry date is found the algorithm ends the iteration before displaying the date. This choice in the design is motivated by the fact that packages sometimes contain two dates, preparation and expiry date, the algorithm must therefore terminate the iteration and return only the highest value of date to minimize the chance of false positives.

V. Results

We performed some initial tests on a small set of images containing 40 images taken from both the internet as well as with a smartphone. In Fig. 4 are reported the computation time required by the Hough transform based rotation correction algorithm to successfully correct an image rotation. While the algorithm generally takes less than a second to execute in some exceptions execution time exceeded 10 seconds, this happened for both probabilistic and full Hough Transforms. In our first implementations we saved to Hard Disk all the generated images, due to the presence of High Definition images with resolutions of 4000×3000 pixels this caused a bottleneck on execution times. We optimized the algorithm to only calculate the rotation and apply the rotation to the image stored in the program's allocated RAM and drastically reduced execution time for all types of images.

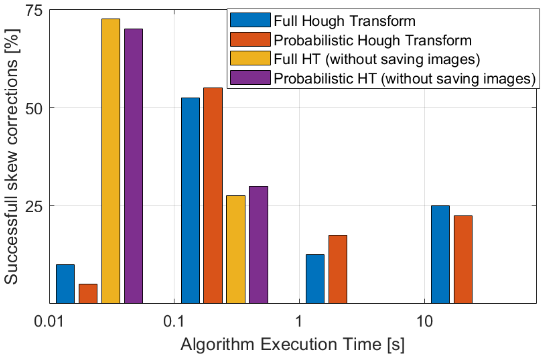
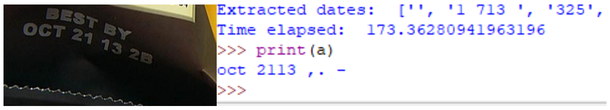


Fig. 4. Algorithm execution time histogram for probabilistic and full Hough Transform. The results show the case where rotation corrected images were generated and saved for analysis and the case of only rotation angle identification



```
Extracted dates:  ['', '1 713 ', '325',
Time elapsed:  173.36280941963196
>>> print(a)
oct 2113 ,. -
>>>
```

(a) Original Image **(b)** Extracted Date and Operation Time

Fig. 5. Example of date extraction error due to bad interpretation of the characters while using sliding window approach

Subsequently, we have studied the impact of our sliding window approach with 4 different combinations of preprocessing:

- No pre-processing
- Adaptive Gaussian Threshold Binarization
- Hough Transform
- Binarization and Hough Transform

We have further defined a correct extraction when the algorithm returns enough digits of the correct date to correctly identify the expiry in the short term, for example if only DD/MM are returned from a DD/MM/YYYY it is still considered a success. On the other hand, a result where only month and year is returned from a date containing the day will not be considered a success. Also cases where all the characters are correctly identified but

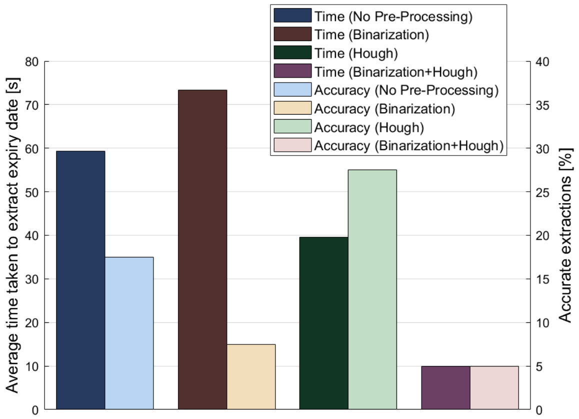


Fig. 6. Accuracy and conversion time results of our algorithm. Four cases are shown: no pre-processing, binarization only, Hough Transform only, Binarization and Hough transform

they are interpreted in a wrong manner, such as in Fig. 5 is still considered a failure.

In Fig. 6 are shown the results of our tests. We used a small test set of 40 unique images. Among the approaches considered the binarization generally decreased performance. With the exception of the combination of Hough Transform and Binarization where convergence time is greatly decreased but this is likely due to the very small set of successful date extraction. From our tests the application of Hough Transform without binarization gives the best results in terms of accurate date extractions and algorithm execution time. The erosion method, not shown in the results, was also tested on the binarized images. The results showed that it further decreased the accuracy of the OCR so it was not included.

VI. Conclusion

In this paper we have presented several preprocessing methods to help general purpouse OCR software to recognize expiry dates, with a particular focus on probabilistic and full Hough Transforms. Our results show that the application of the full Hough transform can both increase the accuracy and reduce the average time needed to extract the expiry date from a picture at minimal computational cost when appropriate measures are taken. Further works will focus on the optimization of the algorithm for the detection of expiry dates and addressing the limitations caused by small labeled datasets currently available.

VII. Acknowledgements

This work was entirely funded from the project: Sistema Integrato di Valorizzazione delle Eccedenze nel Quartiere (SIVEQ, Integrated system for the valorization of food surplus in the neighborhood), winner of the POLISOCIAL Award 2017. This award aims to fosters scientific research with a high social impact and is funded with the tax-return funds donated to Politecnico di Milano. The authors would like to thank the anonymous reviewers for their useful suggestions.

References

[1] R. Sonnino, “Feeding the city: Towards a new research and planning agenda,” *International Planning Studies*, vol. 14, no. 4, pp. 425–435, 2009.

[2] F. HLPE, “Food losses and waste in the context of sustainable food systems,” *A Report by the High Level Panel of Experts on Food Security and Nutrition of the Committee on World Food Security. Available online: http://www. fao. org/3/a-i3901e. pdf (accessed on 2 October 2017)*, 2014.

[3] FAO, “Food wastage footprint-full-cost accounting-final report,” 2014.

[4] U. C. Council, “Gs1 databar family,” *Available from: Lawrenceville NJ: Uniform Code Council https://www. gs1. org/barcodes/databar. Accessed*, vol. 6, 2017.

[5] A. Zaafouri, M. Sayadi, and F. Fnaiech, “A vision approach for expiry date recognition using stretched gabor features.,” *Int. Arab J. Inf. Technol.*, vol. 12, no. 5, pp. 448–455, 2015.

[6] E. Peng, P. Peursum, and L. Li, “Product barcode and expiry date detection for the visually impaired using a smartphone,” in *2012 International Conference on Digital Image Computing Techniques and Applications (DICTA)*, IEEE, dec 2012.

[7] D. H. Ballard, “Generalizing the hough transform to detect arbitrary shapes,” *Pattern Recognition*, vol. 13, pp. 111–122, 1981.

[8] T. Björklund, A. Fiandrotti, M. Annarumma, G. Francini, and E. Magli, “Automatic license plate recognition with convolutional neural networks trained on synthetic data,” in *Multimedia Signal Processing (MMSP), 2017 IEEE 19th International Workshop on*, pp. 1–6, IEEE, 2017.

[9] S. Lucas, “Optical character recognition with hough transform based neural networks,” in *Hough Transforms, IEE Colloquium on*, pp. P7–1, IET, 1993.

[10] B. K. Shukla, G. Kumar, and A. Kumar, “An approach for skew detection using hough transform,” *International Journal of Computer Applications*, vol. 136, no. 9, pp. 20–23, 2016.

[11] “Adaptive image thresholding with python, a tutorial.” https://pythonprogramming.net/thresholding-image-analysis-python-opencv-tutorial/.

[12] “Hough lines with python, a tutorial.” https://docs.opencv.org/3.0-beta/doc/py_tutorials/py_imgproc/py_houghlines/py_houghlines.html.

[13] R. C. Gonzalez and R. E. Woods, *Digital Image Processing (3rd Edition)*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 2006.