Evaluation of Open-Source OCR Libraries for Scene Text Recognition in the Presence of Fisheye Distortion

María Flores[a], David Valiente[b], Marcos Alfaro[c], Marc Fabregat-Jaén[d] and Luis Payá[e]
Institute for Engineering Research (I3E), Miguel Hernandez University,
Avenida de la Universidad, s/, 03202, Elche, Alicante, Spain
{*m.flores, dvaliente, malfaro, mfabregat, lpaya*}*@umh.es*

Keywords: Scene Text Recognition, Fisheye Distortion, Optical Character Recognition.

Abstract: Due to the rich and precise semantic information that text provides, scene text recognition is relevant in a wide range of vision-based applications. In recent years, the use of vision systems that combine a camera and a fisheye lens is common in a variety of applications. The addition of a fisheye lens has the great advantage of capturing a wider field of view, but this causes a great deal of distortion, making certain tasks challenging. In many applications, such as localization or mapping for a mobile robot, the algorithms work directly with fisheye images (i.e. distortion is not corrected). For this reason, the principal objective of this work is to study the effectiveness of some OCR (Optical Character Recognition) open-source libraries applied to images with fisheye distortion. Since no scene text dataset of this kind of image has been found, this work also generates a synthetic image dataset. A fisheye model which varies some parameters is applied to standard images of a benchmark scene text dataset to generate the proposed dataset.

1 INTRODUCTION

Over the years, the use of cameras to acquire information about the environment has grown notably. This is due to the huge amount of information about the environment that can be extracted from an image. There are different vision system configurations, but cameras with fisheye lenses are receiving increased attention (Yang et al., 2023; Flores et al., 2024) because they can capture a wider field of view in a single image.

The rich semantic information that text provides is hugely beneficial in a wide range of vision-based applications. In the same way as for humans, this high-level information helps achieve a better analysis and understanding of the environment. As a result, text detection and recognition have attracted a great deal of attention in recent years. For instance, Yamanaka et al. (2022) propose a method that detects text and arrows on surrounding signage in an equirectangular image captured by a 360-degree camera. The aim is to help blind people determine the correct direction to a destination when they navigate through an unfamiliar public building. Regarding autonomous navigation, Case et al. (2011) present a system to generate a map for a robot that navigates in an office environment, considering that much critical information about a location is included in signs and placards posted on walls. Then, this map collects semantic labels about room numbers, lists of office occupants, or the name of a room or hall.

Optical Character Recognition (OCR) involves recognizing and converting the text that appears in an image into a string-readable format.

The objective of this work is threefold. First, this work aims to evaluate the effectiveness of some open-source OCR tools in the presence of fisheye distortion. To the best of our knowledge, all available scene text datasets are composed of images that comply with pinhole projection, but none are composed of fisheye images. Then, this work also aims to generate a synthetic wide-angle image dataset by applying transformations to the conventional images of a benchmark image dataset for this task. Finally, this work intends to compare two open-source OCR tools using the benchmark (standard images) and the generated (fisheye images) dataset.

The remainder of this paper is structured as follows. In Section 2 and 3, some related works on text

[a] https://orcid.org/0000-0003-1117-0868
[b] https://orcid.org/0000-0002-2245-0542
[c] https://orcid.org/0009-0008-8213-557X
[d] https://orcid.org/0009-0002-4327-0900
[e] https://orcid.org/0000-0002-3045-4316

Flores, M., Valiente, D., Alfaro, M., Fabregat-Jaén, M. and Payá, L.
Evaluation of Open-Source OCR Libraries for Scene Text Recognition in the Presence of Fisheye Distortion.
DOI: 10.5220/0012927600003822

In *Proceedings of the 21st International Conference on Informatics in Control, Automation and Robotics (ICINCO 2024) - Volume 2*, pages 133-140
ISBN: 978-989-758-717-7; ISSN: 2184-2809

recognition and available datasets for this task are outlined respectively. In addition, the problem that this work addresses is clearly stated in both cases. Section 4 presents some OCR tools, with more emphasis on those used in this work. Section 5 describes the transformations that have been applied to generate fisheye images from a standard image. Section 6 is focused on the experimental part, describing the database used and the quality measurement for the evaluation. The results obtained from the experiments are presented and discussed in Section 7. Finally, Section 8 presents the conclusions and future works.

2 SCENE TEXT RECOGNITION

Scene Text Recognition (STR) is a computer vision task that aims to transcribe text that appears in an image captured by a camera in an environment (i.e. scene text) into a sequence of digital characters that encode high-level semantics, which is often essential to fully understand the scene (Du et al., 2022). STR involves two fundamental tasks. Firstly, the text within natural scene images is identified and localized, and its position is often defined by a bounding box. This first task is known as text detection. Secondly, the image regions containing text are converted into machine-readable strings (Lin et al., 2020). This is known as text recognition.

Challenges in STR. In contrast to the recognition of text printed in documents, STR is an arduous task. This complexity can be caused by effects either related to the scenario (e.g. non-uniform illumination, hazy effect, noise, distortion, partial occlusion or background clutter), related to the text (e.g. different sizes, diverse fonts, geometric distortion, color, orientation of the text, languages) or related to the camera (e.g. low resolution and motion blurring) (Gupta and Jalal, 2022; Naosekpam and Sahu, 2022).

Related Works. In view of the latter, STR has recently gained the attention of the computer vision community, and several methods have been proposed. There are several reviews and surveys about this task in the literature, such as (Chen et al., 2020; Naosekpam and Sahu, 2022; Long et al., 2021; Lin et al., 2020). For text detection, Textsnake (Long et al., 2018) follows a Fully Convolutional Network (FCN) model, which estimates the geometry attributes (potentially variable radius and orientation) of each overlapping disk centered at symmetric axes. These disks compose an ordered sequence which describes a text instance. The network architecture is composed of five stages of convolutions. The outputs of each stage (i.e. the feature maps) are fed to the feature merging network. FCENet (Zhu et al., 2021) is featured by modeling the text instances in the Fourier domain. The authors also proposed a novel Fourier Contour Embedding (FCE) method with the objective of representing arbitrary shaped text contours as compact signatures. The framework consists of a backbone, Feature Pyramid Networks (FPN) and a simple post-processing with the Inverse Fourier Transformation (IFT) and Non-Maximum Suppression (NMS).

For text recognition, some of the proposed methods are described next. Convolutional Recurrent Neural Network, (CRNN) (Shi et al., 2017) is an end-to-end trainable method, whose network architecture consists of convolutional layers, followed by recurrent layers and a transcription layer. SAR (Show, Attend and Read) (Li et al., 2019) is an approach that presents good results for regular and irregular text. This model is composed of a Residual neural network (ResNet) Convolutional Neural Network (CNN) (31-layer) for feature extraction, an LSTM based encoder-decoder framework and a 2-Dimensional attention module. RobustScanner (Yue et al., 2020) uses a CNN encoder to obtain the feature map which is then fed into a hybrid branch and a position enhancement branch. After that, the outputs of both branches are dynamically fused by the dynamically-fusing module at each time step.

Problem Statement. In many computer vision applications, images captured by an omnidirectional camera are used mainly due to their wide field of view. The drawback is that these images contain a lot of distortion, and as a consequence, recognizing text can be a challenge. The detection and recognition of curved and distorted text are more challenging than that of horizontal undistorted text. Taking into account the imaging projection of the wide-angle images, scene text, which is horizontal in the original scenario, can appear in the image curved or with other orientations depending on the image region where it was captured. This paper evaluates the robustness of some open-source OCR libraries in the presence of the distortion of fisheye images.

3 SCENE TEXT DATASETS

Related Works. A variety of publicly available benchmark datasets are available for English scene text detection and recognition techniques. Some of them are COCO-Text (Veit et al., 2016), Street View Text (SVT) (Hutchison et al., 2010), Street View Text Perspective (SVTP) (Phan et al., 2013) or ICDAR 2015 (Karatzas et al., 2015). In these datasets, the text usually appears horizontal or rotated but in a linear

(i.e. regular) arrangement. However, the text in the scene can be arranged in curved or other irregular arrangements. Considering this fact, Total-Text (Chng and Chan, 2017) and CTW1500 (Yuliang et al., 2017) datasets were proposed for curved text.

Problem Statement. Some of the mentioned datasets contain images with text that is curved or multi-oriented in the scene. However, all images have been captured with systems that follow a pinhole projection. Then, these images do not present distortion produced by the wide field of view, which is the subject of study of this paper. Then, we apply data augmentation to a benchmark dataset in order to generate distorted images with word annotations.

4 OCR LIBRARIES

Several open-source OCR libraries have been developed so far. The pioneer one was Tesseract toolbox, which Google released in 2006. One of the most recent ones is MMOCR (Multimedia Optical Character Recognition) (Kuang et al., 2021). It is an open-source toolbox with seven text detection approaches it contains, among others, Mask R-CNN (He et al., 2017), FCENet (Zhu et al., 2021) and TextSnake (Long et al., 2018)) and five text recognition algorithms (among which are CRNN (Shi et al., 2017), RobustScanner (Yue et al., 2020), SAR (Li et al., 2019)). The next subsections describe in detail EasyOCR and PaddleOCR, which are the OCR libraries that have been used in the evaluation section of the present work.

4.1 EasyOCR

EasyOCR (JaidedAI, 2020) is a a python-based PyTorch library for OCR created and maintained by Jaided AI. This library, which is implemented using PyTorch library, supports more than 80 languages (among them, English and Spanish). The EasyOCR framework consists of a detection stage and a recognition stage. The former uses CRAFT (Baek et al., 2019) (or other detection models) to find the regions of the image that contain text. Also its corresponding bounding boxes are extracted. The latter stage is based on CRNN (or other recognition models) and is composed mainly of three components:

- Feature Extraction. The useful features from the input image are extracted using a standard CNN without fully connected layers, i.e. ResNet or VGG.
- Sequence Labeling. The feature maps are fed to a Recurrent Neural Network (RNN), such as a Long-Short-Term Memory (LSTM), to interpret the sequential context. This component's output is a sequence of probabilities.
- Decoding. Finally, the sequence of probabilities are transformed into a text label sequence recognised using the Connectionist Temporal Classification (CTC) algorithm (Graves et al., 2006).

4.2 PaddleOCR

PaddleOCR (also knows as PP-OCR) is a practical open-source OCR toolbox based on PaddlePaddle with different versions: PP-OCR (Du et al., 2020), PP-OCRv2 (Du et al., 2021) and PP-OCRv3 (Li et al., 2022). The pipeline of the latter contains three main parts:

- Text Detection. In this part, Differentiable Binarization (DB), which is based on a simple segmentation network, is used. This detection model is trained using CML (Collaborative Mutual Learning) distillation.
- Detection Boxes Rectification. The followed step consists in transforming the text box into a horizontal rectangle one. In order to determine if a text box is reversed (i.e. text direction), a simple image classification model is employed. In the case that it is determined reversed, the text box is flipped.
- Text Recognition. This part is based on SVTR-LCNet, which is a lightweight text recognition network fusing Transformer-based network SVTR (Du et al., 2022) and lightweight CNN-based network PP-LCNet (Cui et al., 2021)

5 WIDE-ANGLE SYNTHETIC DATASET

As described in Section 3, the datasets for scene text recognition are typically composed of conventional images. Therefore, in the present work, a synthetic dataset has been generated from a public annotated dataset using fisheye projections to obtain these images with distortion as it can be seen in Figure 1.

A fisheye image can present more or less distortion depending mainly on the field of view; this distortion is more noticeable in the periphery than in the center of the image. Considering this, in the present work, a set of synthetic fisheye images is generated from a conventional image by varying the focal length value. Also, different 3D motion rigid transformations are applied so that the text appears in different regions of the fisheye image.

(a) Original (b) Synthetic fisheye

Figure 1: The original standard image and the synthetic fisheye image as result of applying the conversion from pinhole to fisheye.

5.1 Data Augmentation

Data augmentation has been applied to achieve a higher number of possible cases. Transformations related to the projection from standard image to fisheye (scale, field of view, and standard focal length) and rigid motion (translation and rotation) are performed.
Scale. This parameter establishes the size of the fisheye image. The generated fisheye images are squared, that is, the height and the width are equal, and their values are the minimum dimension of the original images, i.e. the minimum between the height $H_{original}$ and the width $W_{original}$, multiplied by the set scale value. The synthetic dataset has been generated using three values for the scale parameter: 1 (see 2a), 2 and 4 (see 2f). Table 1 shows the relation between the scale parameter and the dimension of the fisheye image generated.

Table 1: Values of the scale parameter and the dimensions of the images generated.

Original	S = 1	S = 2	S = 4
960x1280	1280x1280	2560x2560	5120x5120

Field of View. In this paper, the focal fisheye length in the equidistant projection has been established as the field of view of the virtual fisheye lens in radians divided by the maximum radius of the fisheye image, which is half of the height. The synthetic dataset was generated using three different values for the field of view: 180. 200 and 220 degrees. Figure 2b shows a synthetic fisheye image setting the field of view to 180 degrees and Figure 2c to 220 degrees.
Standard Focal Length. The effect produced by this parameter in the generated image is a zoom out which is more noticeable the higher this value is. The values are given by:

$$f_{std} = \alpha \cdot min(H_{original}, W_{original}) \quad (1)$$

where $H_{original}$ and $W_{original}$ are the height and width of the original dataset image, respectively, and α takes the following values: 0.6, 0.8 and 1.2. Figure 2d and Figure 2e show the result of setting this parameter to 0.6 and 1.2, respectively.

Translation. In order to simplify, the translation to generate different virtual points of view is coded as a movement to ”left” (see Figure 2h) or ”right” (see Figure 2g). It implies a translation along the positive/negative X-axis.
Rotation. A rotation around the vertical axis is also applied so that text appears in the area of most distortion. In Figure 2j, it can be seen that the text that initially appears in the center without rotation (see Figure 2i) now is on the right side. Notice that this rotation is only applied without translation.

6 EXPERIMENTS

The main objective of this paper is to evaluate the scene text recognition task in images with high distortion. Two open-source OCR libraries (EasyOCR and PaddleOCR) have been applied to recognize the text appearing in the images. In this way, the scene text recognition precision of these libraries with fisheye images is analyzed and compared.

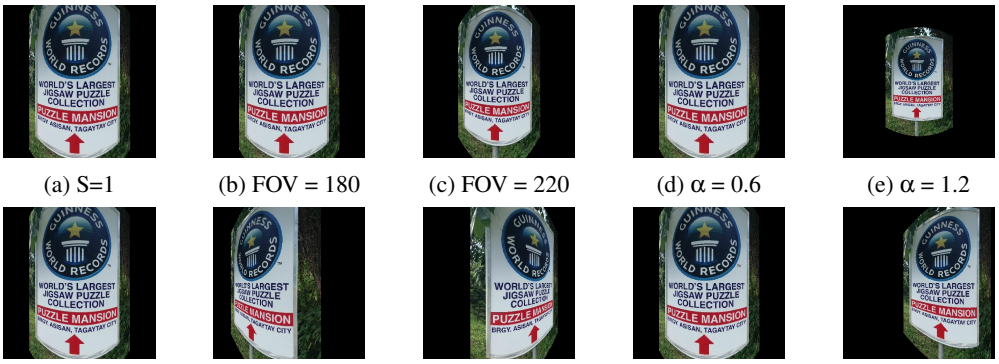
6.1 Dataset

The dataset used in this study is Total-Text (Chng and Chan, 2017). This dataset is composed of a total of 1555 images divided into a set of 1255 and another of 300, which are the training and testing, respectively. In this paper, only the testing set is used. One of the main features of this dataset is that the text in the images appears in different orientations, not only horizontally, as in most datasets. For each annotated word in the dataset, the type of orientation (horizontal, curved or multi oriented), is provided (i.e., the ground truth). Considering this, the results have been separated and analyzed according to this attribute, as described Section 7. After applying the data augmentation, each image of the Total-Text dataset generates 108 fisheye synthetic images. Thus, the total number of generated images in the dataset is 300*108.

In brief, two datasets are used in this work: the original Total-Text (standard images) and the synthetic (fisheye images) dataset generated.

6.2 Evaluation Protocol

Levenshtein distance measures the similarity between two strings. In more detail, the Levenshtein distance determines the minimum number of single-character changes required to convert one word to the other. The changes can be to insert, delete, or substitute a character.



(a) S=1 (b) FOV = 180 (c) FOV = 220 (d) $\alpha = 0.6$ (e) $\alpha = 1.2$

(f) S=2 (g) Move right (h) Move left (i) Without rotation (j) With rotation

Figure 2: Synthetic fisheye images with data augmentation.

Table 2: Open-source OCR libraries.

Method	Version	Model	Github repository
Easy-OCR	1.7.1		JaidedAI/EasyOCR
PP-OCR (Du et al., 2020)	2.7.0.3	English ultra-lightweight PP-OCRv3	PaddlePaddle/PaddleOCR

In this paper, word recognition score is one minus the ratio between the Levenshtein distance and the number of word characters:

$$score_{word} = 1 - \frac{LevDist(word, word_{GT})}{len(word_{GT})} \quad (2)$$

where *word* is the string output by a text recognizer and $word_{GT}$ the ground truth string. The lower the distance (i.e. the fewer the number of changes), the lower the ratio and, therefore, the higher the score.

Two string sets were obtained for each image: the set of recognized words and ground truth words. The procedure followed to determine whether a ground truth word was found was to search for the most similar word in the set of recognized words. In this way, each ground truth word will have a score value associated with it; if it is impossible to find a similar word, it will have a zero score associated with it. These associated scores are used to obtain the set of True Positives (TP) and the set of False Negatives (FN). FN are ground truth words that have not been recognized, i.e. the score is lower than a threshold, whereas TP are ground truth words that have been generally recognized, i.e. the score is higher than a threshold. This threshold has been established with a score value equal to 0.65.

Sensitivity is used to perform the study, and some modifications were made. In this paper, the number of true positives in the general equations is replaced by the sum of scores of them, i.e. $TP = \sum_1^{N_{TP}} score_i$. If the ground truth word ($word_{GT}$) is correctly recognized, the Levenshtein distance is zero, and then the score is equal to one. Thus, the summation could be described as the number of positives weighted according to how similar they are, not just whether they have been recognized correctly or not.

Taking all the above into account, the Quality Measurement (QM) is given by:

$$QM = \frac{\sum_1^{N_{TP}} score_i}{\sum_1^{N_{TP}} score_i + N_{FN}} \quad (3)$$

where N_{FN} is the number of FN.

6.3 Methodology

A synthetic dataset has been created for the evaluation by applying the transformations and procedure described in Section 5.1. After that, the experiments were carried out with two datasets: (1) the original one, composed of standard images (i.e. Total-Text dataset) and (2) the synthetic dataset created from the previous one and composed of synthetic fisheye images. The main objective of this paper is to assess different scene text recognition approaches on these two datasets. Table 2 shows the setup of the open-source OCR libraries used during the experiments.

7 RESULTS AND ANALYSIS

The scores obtained using eq. (2) from all the images of the synthetic and the original dataset are divided according to the orientation of the text in the scene: horizontal (h), multi-oriented (m) or curved (c). The results of the synthetic images are also divided according to the parameter values of the data augmentation. The aim of that latter is to examine the influence of the data augmentation parameters on the effectiveness of the OCR tools, also taking into account the

Table 3: The QM values (eq. (3)) of the synthetic fisheye images. Marked in **bold** if the measure value is greater than or equal to the one obtained with the original dataset (i.e. with no distortion) (i.e. last row). The best result in terms of the library (EasyORC or PaddleOCR) is marked with a colored background .

			EasyOCR			PaddleOCR		
S	FOV	α	h	m	c	h	m	c
1	180	0.6	0.44	**0.50**	**0.50**	0.47	**0.50**	0.47
		0.8	0.44	**0.50**	**0.50**	0.47	**0.50**	**0.50**
		1,2	**0.50**	**0.50**	0.40	0.47	**0.40**	**0.50**
	200	0.6	**0.50**	**0.50**	**0.50**	**0.50**	**0.50**	0.45
		0.8	0.44	**0.50**	**0.50**	0.47	**0.40**	**0.50**
		1,2	**0.50**	0.42	0.40	**0.50**	**0.46**	**0.50**
	220	0.6	**0.50**	**0.50**	**0.50**	0.47	**0.50**	**0.50**
		0.8	**0.50**	0.40	**0.50**	0.47	**0.50**	**0.50**
		1,2	**0.50**	0.45	0.40	**0.50**	**0.50**	**0.50**
2	180	0.6	**0.50**	**0.50**	**0.50**	**0.50**	**0.50**	0.43
		0.8	0.44	**0.50**	0.45	0.47	**0.40**	**0.50**
		1,2	**0.50**	**0.50**	**0.50**	**0.50**	**0.50**	**0.50**
	200	0.6	**0.50**	**0.50**	**0.50**	**0.50**	**0.50**	**0.50**
		0.8	0.44	0.43	**0.50**	0.47	**0.45**	**0.50**
		1,2	**0.50**	**0.50**	**0.50**	**0.50**	**0.50**	**0.50**
	220	0.6	0.44	**0.50**	**0.50**	**0.50**	**0.50**	**0.50**
		0.8	0.44	0.43	**0.50**	0.47	**0.50**	**0.50**
		1,2	**0.50**	0.43	**0.50**	**0.50**	**0.50**	**0.50**
4	180	0.6	**0.50**	**0.50**	0.45	**0.50**	**0.50**	0.47
		0.8	0.44	0.43	0.45	0.47	**0.45**	**0.50**
		1,2	**0.50**	0.43	**0.50**	**0.50**	**0.50**	0.45
	200	0.6	0.47	**0.50**	**0.50**	**0.50**	**0.50**	0.43
		0.8	0.47	0.43	**0.50**	0.47	**0.45**	**0.50**
		1,2	0.44	**0.50**	**0.50**	**0.50**	**0.45**	**0.50**
	220	0.6	0.47	0.43	**0.50**	0.47	**0.50**	**0.50**
		0.8	0.44	**0.50**	0.45	0.47	**0.40**	**0.50**
		1,2	**0.50**	0.43	**0.50**	**0.50**	**0.42**	0.40
Standard			0.50	0.50	0.50	0.50	0.40	0.50

(a) No translation and rotation.

			EasyOCR			PaddleOCR		
S	FOV	α	h	m	c	h	m	c
1	180	0.6	**0.50**	0.45	**0.50**	**0.50**	**0.50**	**0.50**
		0.8	0.47	**0.50**	**0.50**	0.47	**0.46**	0.45
		1,2	0.44	**0.50**	**0.50**	0.47	**0.42**	**0.50**
	200	0.6	0.47	0.43	0.45	0.47	**0.45**	0.45
		0.8	0.44	0.43	0.45	0.47	**0.40**	**0.50**
		1,2	0.47	**0.50**	**0.50**	**0.50**	**0.46**	**0.50**
	220	0.6	**0.50**	0.43	**0.50**	0.47	**0.47**	0.45
		0.8	0.47	0.43	0.40	0.47	**0.42**	0.44
		1,2	0.42	**0.50**	0.40	**0.50**	**0.46**	**0.50**
2	180	0.6	**0.50**	**0.50**	0.45	0.47	**0.50**	**0.50**
		0.8	**0.50**	**0.50**	0.45	0.47	**0.50**	0.45
		1,2	**0.50**	0.40	**0.50**	0.47	**0.50**	**0.50**
	200	0.6	**0.50**	**0.50**	0.45	0.47	**0.50**	0.45
		0.8	**0.50**	**0.50**	**0.50**	0.47	**0.50**	**0.50**
		1,2	**0.50**	**0.50**	0.45	0.47	**0.50**	**0.50**
	220	0.6	**0.50**	**0.50**	0.45	0.47	**0.50**	0.45
		0.8	**0.50**	0.45	**0.50**	0.47	**0.45**	**0.50**
		1,2	**0.50**	**0.50**	0.45	0.47	**0.40**	0.45
4	180	0.6	**0.50**	**0.50**	**0.50**	0.47	**0.50**	**0.50**
		0.8	**0.50**	0.45	0.45	0.47	**0.50**	0.40
		1,2	**0.50**	**0.50**	0.45	0.47	**0.50**	**0.50**
	200	0.6	**0.50**	**0.50**	**0.50**	0.47	**0.50**	0.45
		0.8	**0.50**	**0.50**	**0.50**	0.47	**0.50**	**0.50**
		1,2	0.44	**0.50**	0.40	0.47	**0.50**	**0.50**
	220	0.6	**0.50**	0.43	0.45	0.47	**0.50**	0.45
		0.8	**0.50**	0.45	0.45	0.47	**0.50**	**0.50**
		1,2	0.44	**0.50**	0.40	0.47	**0.50**	**0.50**
Standard			0.50	0.50	0.50	0.50	0.40	0.50

(b) Motion left.

			EasyOCR			PaddleOCR		
S	FOV	α	h	m	c	h	m	c
1	180	0.6	0.44	0.43	0.46	0.47	**0.50**	**0.50**
		0.8	**0.50**	0.43	0.40	**0.50**	**0.50**	**0.50**
		1,2	**0.50**	0.43	**0.50**	**0.50**	**0.44**	**0.50**
	200	0.6	**0.50**	0.43	0.40	**0.50**	**0.50**	0.43
		0.8	**0.50**	0.43	0.44	**0.50**	**0.43**	**0.50**
		1,2	**0.50**	0.46	0.44	**0.50**	**0.50**	0.44
	220	0.6	**0.50**	0.43	0.42	**0.50**	**0.50**	**0.50**
		0.8	**0.50**	0.43	0.46	**0.50**	**0.50**	**0.50**
		1,2	**0.50**	0.46	0.42	**0.50**	**0.50**	0.43
2	180	0.6	0.44	**0.50**	**0.50**	0.47	**0.50**	0.43
		0.8	**0.50**	0.43	**0.50**	**0.50**	**0.50**	**0.50**
		1,2	**0.50**	0.43	**0.50**	**0.50**	**0.42**	**0.50**
	200	0.6	**0.50**	0.43	**0.50**	0.47	**0.50**	0.43
		0.8	**0.50**	0.43	**0.50**	**0.50**	**0.50**	**0.50**
		1,2	**0.50**	**0.50**	**0.50**	**0.50**	**0.50**	**0.50**
	220	0.6	0.47	0.43	**0.50**	**0.50**	**0.50**	0.43
		0.8	**0.50**	0.43	**0.50**	**0.50**	**0.50**	**0.50**
		1,2	**0.50**	**0.50**	0.46	**0.50**	**0.50**	**0.50**
4	180	0.6	**0.50**	**0.50**	**0.50**	0.47	**0.50**	0.40
		0.8	0.44	**0.50**	**0.50**	**0.50**	**0.50**	**0.50**
		1,2	**0.50**	**0.50**	**0.50**	**0.50**	**0.42**	**0.50**
	200	0.6	0.44	0.43	**0.50**	0.47	**0.50**	**0.50**
		0.8	**0.50**	**0.50**	**0.50**	**0.50**	**0.50**	**0.50**
		1,2	**0.50**	0.43	0.40	**0.50**	**0.50**	0.43
	220	0.6	0.44	**0.50**	**0.50**	**0.50**	**0.50**	0.43
		0.8	**0.50**	**0.50**	**0.50**	**0.50**	**0.50**	**0.50**
		1,2	**0.50**	0.43	**0.50**	**0.50**	**0.42**	**0.50**
Standard			0.50	0.50	0.50	0.50	0.40	0.50

(c) Motion to the right.

			EasyOCR			PaddleOCR		
S	FOV	α	h	m	c	h	m	c
1	180	0.6	0.44	0.45	**0.50**	0.47	**0.50**	0.43
		0.8	0.44	0.43	**0.50**	0.47	**0.50**	**0.50**
		1,2	0.44	**0.50**	**0.50**	**0.50**	**0.50**	**0.50**
	200	0.6	0.44	0.40	**0.50**	**0.50**	**0.50**	**0.50**
		0.8	0.44	0.40	0.40	0.47	**0.50**	0.43
		1,2	**0.50**	**0.50**	0.45	**0.50**	**0.50**	**0.50**
	220	0.6	0.44	0.40	**0.50**	0.47	0.40	**0.50**
		0.8	0.44	0.40	0.40	0.47	**0.44**	**0.50**
		1,2	**0.50**	**0.50**	0.40	0.44	**0.50**	**0.50**
2	180	0.6	0.44	**0.50**	0.45	0.47	**0.43**	0.47
		0.8	0.44	0.43	**0.50**	**0.50**	0.40	**0.50**
		1,2	**0.50**	0.43	**0.50**	**0.50**	0.40	**0.50**
	200	0.6	0.44	0.43	**0.50**	0.44	0.40	**0.50**
		0.8	0.44	**0.50**	**0.50**	**0.50**	0.40	**0.50**
		1,2	0.44	0.43	**0.50**	**0.50**	0.40	0.45
	220	0.6	0.44	**0.50**	**0.50**	0.47	**0.45**	**0.50**
		0.8	0.44	0.43	**0.50**	**0.50**	**0.45**	**0.50**
		1,2	**0.50**	0.43	**0.50**	**0.50**	**0.42**	0.40
4	180	0.6	0.44	0.43	**0.50**	0.47	0.40	0.43
		0.8	0.44	0.43	**0.50**	**0.50**	0.40	**0.50**
		1,2	**0.50**	0.43	**0.50**	**0.50**	**0.42**	**0.50**
	200	0.6	0.44	0.43	**0.50**	0.44	**0.45**	**0.50**
		0.8	0.44	0.43	**0.50**	**0.50**	0.40	**0.50**
		1,2	**0.50**	0.43	**0.50**	**0.50**	0.40	**0.50**
	220	0.6	0.44	0.43	**0.50**	0.47	0.40	**0.50**
		0.8	**0.50**	0.43	**0.50**	0.44	**0.45**	**0.50**
		1,2	**0.50**	**0.50**	**0.50**	**0.50**	**0.42**	**0.50**
Standard			0.50	0.50	0.50	0.50	0.40	0.50

(d) Rotation

natural orientation of the text in the scene. Then, QM is calculated using eq. (3) for each group and all of them will be shown and analyzed in this section. The results are shown in four tables, depending on the extrinsic parameters: without motion (Table 3a), motion to left (Table 3b), motion to right (Table 3c) and rotation around the Y-axis (Table 3d). The last row shows the results for the original dataset (i.e. the one that contains undistorted images). The results related to the synthetic fisheye images are compared with this row. If the QM value is higher or equal, it is marked in bold. Also, the QM values obtained with both libraries in the same setting (row) are compared for the three orientations. If a library has a higher value than the other for a given orientation, the background of this cell is colored.

As shown in Table 3a, PaddleOCR and EasyOCR output a QM value higher or equal to the case of undistorted text (last row) the same number of times when the text orientation the case of undistorted text

is horizontal (h). However, PaddleOCR returned a higher QM value in eight configurations more than EasyOCR. This also occurs when the orientation is curved, but the difference in this case is lower, PaddleOCR has a higher QM value only in one more configuration. In the case of multi-orientated (m) text, PaddleOCR improved or equaled the result of the standard images in a total of 27 configurations, while EasyOCR did it in 16. Additionally, PaddleOCR has improved the results of EasyOCR in four settings.

Considering now the results when the virtual fisheye camera is moved to the left (Table 3b), PaddleOCR performs better than EasyOCR when the natural orientation of the text is multi-oriented or curved. In these cases, PaddleOCR reached the QM value of standard images 2 and 16 times and improved it 25 times for multi-oriented text (i.e. the value is higher than 0.4). In the case of EasyOCR, the same QM value than using images without distortion is achieved 17 times for multi-oriented and 11 times for curved. By contrast, EasyOCR works better for horizontal text, achieving the same value than applied on standard images 18 times, unlike PaddleOCR, which achieves it only three times. By comparing both columns, EasyOCR has a higher QM value than PaddleOCR in 17 settings, whereas the opposite is met in 6 configurations.

After analyzing the results when the virtual fisheye camera is moved to the right (Table 3c), the conclusion is that PaddleOCR outperforms EasyOCR using standard images when the text is multi-oriented. In addition this library works better than EasyOCR independently of the orientation.

For the results when the virtual fisheye camera is rotated around the vertical axis (Table 3d), EasyOCR and PaddleOCR have similar behavior for curved text. However, PaddleOCR achieved a better or equal QM value as the obtained without distortion more times than EasyOCR when the orientation is multi-oriented or horizontal. On the one hand, if we analyze the number of cells colored in the second column (m) of each library, EasyOCR has outperformed PaddleOCR almost twice as often. On the other hand, if we analyze the number of cells colored in the first column (h) of each library, PaddleOCR has outperformed EasyOCR more than eight times, 17 using PaddleOCR against to 2 using EasyOCR.

8 CONCLUSION

In this paper, two open-source libraries for text recognition have been evaluated using fisheye images. Given that no database with this kind of image (highly distorted) has been found for this task, this dataset has been generated by converting standard images of a benchmark text scene dataset to fisheye for different projection parameter values.

After applying two well-recognized and publicly available OCR libraries, the results show that in most cases, these tools perform worse with distorted images. By comparing both libraries, EasyOCR and PaddleOCR, the latter one works better in a larger number of studied configurations, in terms of the QM used.

As possible future work, firstly, it is proposed to evaluate other libraries in this work. Secondly, the tools will be trained to recognize text in the presence of this type of distortion.

ACKNOWLEDGEMENTS

This work is part of the project TED2021-130901B-I00 funded by MCIN/AEI/10.13039/501100011033 and by the European Union "NextGenerationEU"/PRTR. The work is also part of the project PROMETEO/2021/075 funded by Generalitat Valenciana.

REFERENCES

Baek, Y., Lee, B., Han, D., Yun, S., and Lee, H. (2019). Character Region Awareness for Text Detection. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9357–9366, Long Beach, CA, USA. IEEE.

Case, C., Suresh, B., Coates, A., and Ng, A. Y. (2011). Autonomous sign reading for semantic mapping. In *2011 IEEE International Conference on Robotics and Automation*, pages 3297–3303. ISSN: 1050-4729.

Chen, X., Jin, L., Zhu, Y., Luo, C., and Wang, T. (2020). Text Recognition in the Wild: A Survey. arXiv:2005.03492 [cs].

Chng, C. K. and Chan, C. S. (2017). Total-Text: A Comprehensive Dataset for Scene Text Detection and Recognition. In *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, pages 935–942, Kyoto. IEEE.

Cui, C., Gao, T., Wei, S., Du, Y., Guo, R., Dong, S., Lu, B., Zhou, Y., Lv, X., Liu, Q., Hu, X., Yu, D., and Ma, Y. (2021). PP-LCNet: A Lightweight CPU Convolutional Neural Network. arXiv:2109.15099 [cs].

Du, Y., Chen, Z., Jia, C., Yin, X., Zheng, T., Li, C., Du, Y., and Jiang, Y.-G. (2022). SVTR: Scene Text Recognition with a Single Visual Model. arXiv:2205.00159 [cs].

Du, Y., Li, C., Guo, R., Cui, C., Liu, W., Zhou, J., Lu, B., Yang, Y., Liu, Q., Hu, X., Yu, D., and Ma, Y. (2021).

PP-OCRv2: Bag of Tricks for Ultra Lightweight OCR System. arXiv:2109.03144 [cs].

Du, Y., Li, C., Guo, R., Yin, X., Liu, W., Zhou, J., Bai, Y., Yu, Z., Yang, Y., Dang, Q., and Wang, H. (2020). PP-OCR: A Practical Ultra Lightweight OCR System. arXiv:2009.09941 [cs].

Flores, M., Valiente, D., Peidró, A., Reinoso, O., and Payá, L. (2024). Generating a full spherical view by modeling the relation between two fisheye images. *The Visual Computer*.

Graves, A., Fernández, S., Gomez, F., and Schmidhuber, J. (2006). Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning - ICML '06*, pages 369–376, Pittsburgh, Pennsylvania. ACM Press.

Gupta, N. and Jalal, A. S. (2022). Traditional to transfer learning progression on scene text detection and recognition: a survey. *Artificial Intelligence Review*, 55(4):3457–3502.

He, K., Gkioxari, G., Dollár, P., and Girshick, R. (2017). Mask R-CNN. In *2017 IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988. ISSN: 2380-7504.

Hutchison, D., Kanade, T., Kittler, J., Kleinberg, J. M., Mattern, F., Mitchell, J. C., Naor, M., Nierstrasz, O., Pandu Rangan, C., Steffen, B., Sudan, M., Terzopoulos, D., Tygar, D., Vardi, M. Y., Weikum, G., Wang, K., and Belongie, S. (2010). Word Spotting in the Wild. In Daniilidis, K., Maragos, P., and Paragios, N., editors, *Computer Vision – ECCV 2010*, volume 6311, pages 591–604. Springer Berlin Heidelberg, Berlin, Heidelberg.

JaidedAI (2020). EasyOCR.

Karatzas, D., Gomez-Bigorda, L., Nicolaou, A., Ghosh, S., Bagdanov, A., Iwamura, M., Matas, J., Neumann, L., Chandrasekhar, V. R., Lu, S., Shafait, F., Uchida, S., and Valveny, E. (2015). ICDAR 2015 competition on Robust Reading. In *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, pages 1156–1160, Tunis, Tunisia. IEEE.

Kuang, Z., Sun, H., Li, Z., Yue, X., Lin, T. H., Chen, J., Wei, H., Zhu, Y., Gao, T., Zhang, W., Chen, K., Zhang, W., and Lin, D. (2021). MMOCR: A Comprehensive Toolbox for Text Detection, Recognition and Understanding. In *Proceedings of the 29th ACM International Conference on Multimedia*, pages 3791–3794, Virtual Event China. ACM.

Li, C., Liu, W., Guo, R., Yin, X., Jiang, K., Du, Y., Du, Y., Zhu, L., Lai, B., Hu, X., Yu, D., and Ma, Y. (2022). PP-OCRv3: More Attempts for the Improvement of Ultra Lightweight OCR System. arXiv:2206.03001 [cs].

Li, H., Wang, P., Shen, C., and Zhang, G. (2019). Show, Attend and Read: A Simple and Strong Baseline for Irregular Text Recognition. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):8610–8617.

Lin, H., Yang, P., and Zhang, F. (2020). Review of Scene Text Detection and Recognition. *Archives of Computational Methods in Engineering*, 27(2):433–454.

Long, S., He, X., and Yao, C. (2021). Scene Text Detection and Recognition: The Deep Learning Era. *International Journal of Computer Vision*, 129(1):161–184.

Long, S., Ruan, J., Zhang, W., He, X., Wu, W., and Yao, C. (2018). TextSnake: A Flexible Representation for Detecting Text of Arbitrary Shapes. pages 20–36.

Naosekpam, V. and Sahu, N. (2022). Text detection, recognition, and script identification in natural scene images: a Review. *International Journal of Multimedia Information Retrieval*, 11(3):291–314.

Phan, T. Q., Shivakumara, P., Tian, S., and Tan, C. L. (2013). Recognizing Text with Perspective Distortion in Natural Scenes. In *2013 IEEE International Conference on Computer Vision*, pages 569–576. ISSN: 2380-7504.

Shi, B., Bai, X., and Yao, C. (2017). An End-to-End Trainable Neural Network for Image-Based Sequence Recognition and Its Application to Scene Text Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(11):2298–2304.

Veit, A., Matera, T., Neumann, L., Matas, J., and Belongie, S. (2016). COCO-Text: Dataset and Benchmark for Text Detection and Recognition in Natural Images. arXiv:1601.07140 [cs].

Yamanaka, Y., Kayukawa, S., Takagi, H., Nagaoka, Y., Hiratsuka, Y., and Kurihara, S. (2022). One-shot wayfinding method for blind people via ocr and arrow analysis with a 360-degree smartphone camera. *Lecture Notes of the Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering, LNICST*, 419 LNICST:150–168.

Yang, L., Li, L., Xin, X., Sun, Y., Song, Q., and Wang, W. (2023). Large-Scale Person Detection and Localization using Overhead Fisheye Cameras.

Yue, X., Kuang, Z., Lin, C., Sun, H., and Zhang, W. (2020). RobustScanner: Dynamically Enhancing Positional Clues for Robust Text Recognition. In *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XIX*, pages 135–151, Berlin, Heidelberg. Springer-Verlag.

Yuliang, L., Lianwen, J., Shuaitao, Z., and Sheng, Z. (2017). Detecting Curve Text in the Wild: New Dataset and New Solution. arXiv:1712.02170 [cs].

Zhu, Y., Chen, J., Liang, L., Kuang, Z., Jin, L., and Zhang, W. (2021). Fourier Contour Embedding for Arbitrary-Shaped Text Detection. In *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3122–3130, Nashville, TN, USA. IEEE.