Journal of Information Hiding and Multimedia Signal Processing ©2023 ISSN 2073-4212
Ubiquitous International Volume **14**, Number **4**, December **2023**

Semantic segmentation network for horizontal scene text detection

Soufiane Naim

Laboratory of mathematics, computer science and applications
University Hassan II, Faculty of science and technology
BP 146, Mohammedia, 28806, Morocco
naimsoufiane@gmail.com

Noureddine Moumkine

Laboratory of mathematics, computer science and applications
University Hassan II, Faculty of science and technology
BP 146, Mohammedia, 28806, Morocco
noureddine.moumkine@fstm.ac.ma

Received May 2023; revised August 2023, accepted September 2023

Abstract. *Text detection in natural scene images is an active research field that is part of many day life applications. This task is far from being an easy one due to many factors such as text variability of size, color and shape, etc. In this article, we propose a new pipeline to deal with the detection of text that appears in a horizontal orientation. Our proposed method is based on a semantic segmentation architecture inspired from the UNet framework. In the encoder part, we use the Darknet network as a backbone which aims to extract global features inside the input image. In the decoder part, we use the transposed convolution in order to upsample the input feature maps. The final output of our model is a prediction map where each pixel that belong to every text area in an image is highlighted. This prediction map is combined with a graph algorithm in the post-processing step in order to predict the bounding box coordinates surrounding text locations. Our method has achieved a precision of 79.49 and a recall of 80.79 on the ICDAR 2013 dataset*

Keywords: deep learning, computer vision, scene text detection, convolutional neural network, Dice loss

1. **Introduction.** Scene text detection (STD) is an active research field which has many benefits that cover many domains such as tourism where detecting text is the first step before recognizing it and translating it into any language or autonomous driving where the detection and the recognition can help systems to make suggestions to the driver to find the right path. The main objective of STD is to localize text positions in natural scene images. This might be different from other traditional Optical Character Recognition (OCR) systems [29] which try to detect texts in document images that contain typed or handwritten text. In such documents, text generally appears in organized horizontal lines which make the detection an easy task. On the other hand, STD suffers from many challenging problems that make the detection a hard task. Firstly, texts in natural scene images appear in random positions with random sizes, fonts and colors. Secondly, the text can be found in different orientations (horizontal, vertical, or inclined). Finally, the background may appear in shapes that look like some characters which can lead to false detections.

The STD also differs from object detection systems as this one always tries to find objects represented with a compact shape and specific form. For example, a cat is represented with a specific shape, having a head, legs and a queue, which makes it different from a human. So, systems will easily distinguish between them. Unlike the STD where the system may find the text as a single character, word or even a sequence of words. This may conduct to partial detections (detect part of a word or a sequence not the whole one) that can be seen as false detections leading to drawback system performances.

2. **RELATED WORKS.** The scene text detection was the subject of many research papers. The early works were based on handcrafting strategies where we can distinguish between two main streams : Sliding window method (SW) [1][2] : in which a multi scale windows are used to scan the whole image. Each scanned region is classified as a text or a non-text region using a pre-trained classifier. The positive regions are then grouped to make final detections using some graph based methods like conditional random field. The connected components analysis (CCA) [3][4] : in this method, text candidate components are extracted (such as characters) based on color clustering or extreme region extraction. Then, text regions can be distinguished using manual rules or a pre-trained classifier. The most known (CCA) strategies are stroke width transform (SWT)[5] and maximally stable extremal region (MSER)[6]

In recent years, deep learning technics have helped to make a great improvement in text detection performances. Many research papers have made use of convolutional neural network (CNN) to obtain better results than handcrafting strategies. These networks are considered as a deep architecture composed of many layers where the main goal is to extract local and global features in an image; then, use them to identify text regions. Within these methods, there are strategies that consider text as an ordinary object[7][26][27]. So, they use general object detection models such as RCNN[8],YOLO[9],etc, in order to define a rectangular Bounding Box (BB) around the text. The BB detection is performed either by using a region proposal network combined with some defined anchors, or by using a direct regression through the prediction of the rectangular corners.

Other methods use semantic segmentation with the aim being to predict whether each pixel of the image belongs to a text area. such methods are generally based on a fully connected network [10] so as to produce a score map consisting of probabilities of each pixel to be in a text area. One of the advantages of these methods is that they do not need to define anchors to make detections; Zhou and Yao[11] were inspired by the U-Net[12] architecture to define the EAST model which consists of an FCN that produces both a score map, which predicts the location of the text in the image, and the coordinates of the rectangular BB around it, as well. The EAST model predicts efficiently text in horizontal and inclined orientation but has limitations detecting vertical and curved text. TextBoxes is another efficient semantic segmentation method. It can detect horizontal text in a scene image even if the background is complicated. The detection is made in a single forward pass. This method suffers from some limitations (filled to handle overexposed images and to detect text with large spacing characters)[28]

In this study, we leverage prior research to introduce a novel framework for identifying horizontally oriented text in natural scene images. Our approach involves:

- The development of a convolutional neural network architecture, drawing inspiration from the U-Net architecture. In the decoder part of our model, we incorporate the transposed convolutions to facilitate the enlargement of feature maps. To streamline subsequent processing, we advocate for reducing the dimensions of the output

segmentation map to one-fourth(1/4) of the original image size, thereby alleviating computational cost.

- In addition, we introduce a novel pipeline that takes advantage of the Suzuki and Abe's counter algorithm [25]. This pipeline is designed to manipulate the output segmentation map of our architecture to identify text locations and find bounding box coordinates to surround them.

3. OUR METHOD.

3.1. **NETWORK STRUCTURE.** To deal with text detection, we propose to focus on semantic segmentation methodology. The aim is to build a prediction map whose size is $\frac{1}{4}$ of the original image size. This map will allow us to make pixel-wise predictions and retrieve each pixel that belongs to a text area; then, it will be used in the pre-processing step to detect the coordinates of the BB. The reduction of the image size has been chosen in order to reduce the load and the computation time in the pre-processing step. Our model architecture is inspired from the U-Net, which is composed from two principal parts. The first one is the encoder part that serves the function of reducing and extracting features from the image. Then, its output will be the input of the decoder part, part two, whose objective will be to upsample the image and produce the prediction map. We use Darknet53[13] as the backbone of the encoder part. This architecture has been introduced in the Yolov3[13]. It consists of 5 levels of convolution layers which allow to extract the text features and reduce the size of the input image to 1/32 of the original size. We denote each feature map of the encoder part as fi. In the decoder part, we use transposed convolution to increase the spatial dimensions of the features maps. This kind of convolution is flexible and will allow us to add more parameters to better upsample the image. The decoder part is composed of 3 levels of features maps denoted as hi; each one of them doubles the size of the previous one. hi blocs will be combined with fi ones from the encoder part having corresponding sizes. We can represent them as following:

$$g_i = \begin{cases} TransposedConv(h_i) \;\; if \;\; i \leq 3 \\ Conv_{3,3}(h_i) \;\; otherwise \end{cases} \tag{1}$$

$$h_i = \begin{cases} f_i \;\; if \;\; i = 1 \\ Conv_{3,3}(Conv_{1,1}(Concatenate(f_i + g_i))) \;\; otherwise \end{cases} \tag{2}$$

Where h_i is a decoder feature map, g_i is its first layer. The operator "+" represents the concatenation of the encoder bloc fi and the decoder bloc g_i of the same size. In each decoder bloc, the feature map from the last stage is the input of the transposed convolution layer which doubles its size and reduces its depth. Then, we concatenate its output with the encoder bloc fi having the same size, next a 1×1 convolutional layer is applied to normalize the features maps along the channel dimension. At last, we find 3×conv3×3 layers that reduce the number of channels before to feed it to the output layer. The output layer is represented by a single convolutional layer that use the sigmoid function to generate the predictions

3.2. **Loss function.** In scene images, text generally occupies a very small amount of pixels. Hence, positive labels are less represented than negative ones, leading to an imbalanced representation. Using the classic Binary Cross Entropy loss[14] may not help us to get better results since it functions better in equal data distribution. That's why, we have

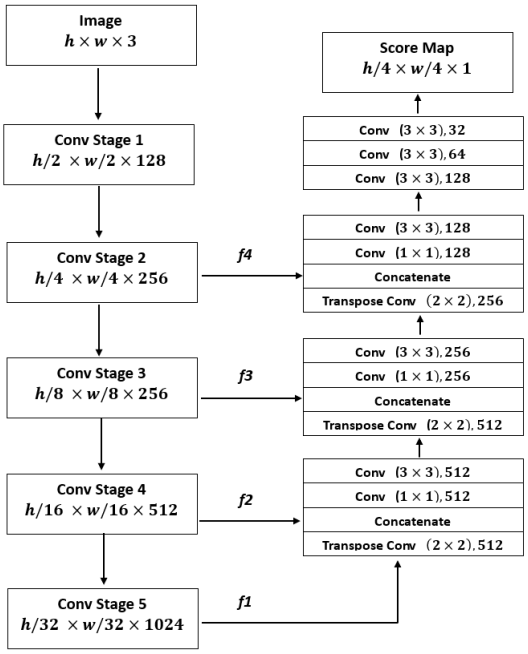

FIGURE 1. The proposed architecture consists of two parts. Feature extraction part (encoder) composed of Darknet53 as backbone and the upsampling part (decoder) which output a prediction map indicating locations of text within the image

chosen to use the Dice Loss[15] which works better in such cases. The dice loss helps to compute the similarity between the ground truth and the output predictions. The Dice Loss function helps to estimate the overlapping between pixels of the output predictions and the ground truth labels. Its value ranges from 0 to 1, where 1 represents a complete overlap of pixels. The dice loss formula is:

$$DSC = \frac{2 \mid X \cap Y \mid}{\mid X \mid + \mid Y \mid} \tag{3}$$

$$Dice\ Loss = 1 - DSC \tag{4}$$

where X and Y respectively represent the prediction label and the ground truth. DSC is the Dice Similarity Coefficient used to control the weight of the Dice loss in the overall loss function.

Minimizing the Dice Loss is equivalent to maximizing the Dice coefficient, which is a measure of similarity between the sets. This means that as the Dice Loss decreases, the overlap between the predicted and true segments increases, leading to a better-performing segmentation model. the DSC can be expressed in another way as:

$$DSC = \frac{2\,TP}{2\,TP + FP + FN} \tag{5}$$

Where TP, FP and FN respectively represent the amount of pixels classified as True Positive, False Positive and False Negative.

By analyzing this formula, we can list several benefits of utilizing the dice loss in comparison to other loss functions when dealing with image segmentation. First, it is more robust to class imbalance than other loss functions, such as cross-entropy loss. This is because the Dice loss takes into account both the number of pixels that are predicted to be foreground and the number of pixels that are actually foreground. Second, the Dice loss encourages the model to produce predictions that have clear boundaries between the foreground and background classes. This is because the Dice loss penalizes the model for predicting pixels that are both predicted to be foreground and are actually background.

3.3. **Post processing.** Since our output map is a matrix of probabilities where the values of pixels belonging to a text area are close to 1 while the others are close to 0, we transform it into a binary image by thresholding pixel's values. Then, we apply the Suzuki et Abe[25] counter algorithm in order to find pixels composing each text bloc boundary. Although this algorithm helps to find both of inner and outer boundaries, we just focus on the outer one since it will allow us to define the coordinates of the bounding boxes. The defined algorithm returns a list of all pixels coordinates defining the counter around a text $[(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)]$. We loop over this list of pixels to find $min(x_1, \ldots, x_n), max(x_1, \ldots, x_n), min(y_1, \ldots, y_n), max(y_1, \ldots, y_n)$ which helps us to define the top left pixel (x_t, y_t) of the bounding box together with its width (w) and height (h).

$$x_t = min(x_1, \ldots, x_n) \tag{6}$$

$$y_t = min(y_1, \ldots, y_n) \tag{7}$$

$$w = max(x_1, \ldots, x_n) - min(x_1, \ldots, x_n) \tag{8}$$

$$h = max(y_1, \ldots, y_n) - min(y_1, \ldots, y_n) \tag{9}$$

To refine the results of the last step, we use the non-maximum suppression (NMS) algorithm to delete any overlapping BB. We consider this step to be optional since the Suzuki et Abe algorithm always generates one box per text area. We choose to use it just to avoid any misrepresentation that could be resulting from our predictions map.

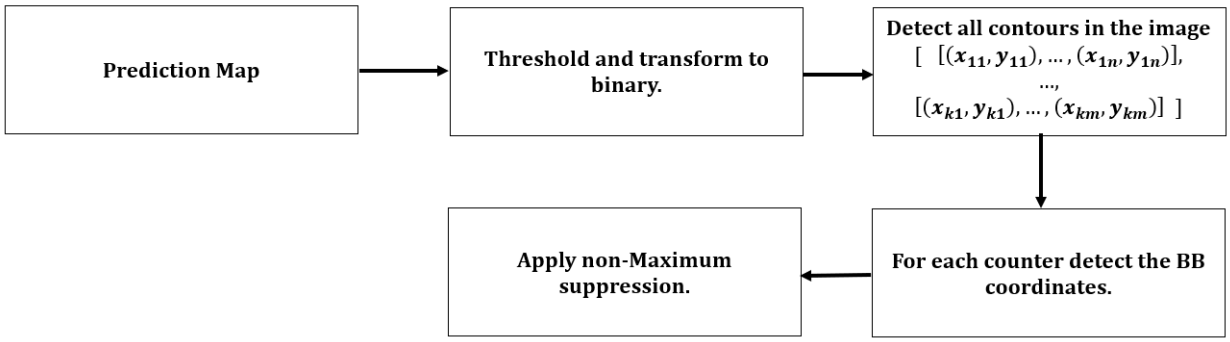

FIGURE 2. Post processing steps

4. Experimental Results.

4.1. **Benchmark Datasets.** In order to implement and test our model, we have used some scene text datasets. These ones have been the subject of the training and inference steps. Hereby, we make a brief presentation of them :

- **MLT 2017 Dataset** [16]: In 2017, the International Conference on Document Analysis and Recognition (ICDAR) introduced MLT dataset (Multi script text detection dataset) where the text in the images has many orientations. It has been provided from many languages: Arabic, Latin, Chinese, Japanese, Korean, Bangla, Symbols. We can note that some images do not contain any text. The proposed annotations are in the quadrangle format (x1,y1, x2,y2, x3,y3, x4,y4) which denote the four corners of the quadrangle surrounding the text. The dataset is composed of 7200 images in the training set and 1200 images in the validation set. This dataset is used to train our model. Even though we are interested only in the horizontal text, this dataset is very interesting since the languages that compose it are very diverse.
- **ICDAR2013 Dataset** [17]: This dataset is provided by the same ICDAR international conference in 2013. It is composed of horizontal English text and it is divided into 229 images for training and 233 images for test. We use this dataset at the inference step to test our model and verify its performance. We also use it to perform some fine tuning as we use the ICDAR 2017 dataset in the first training stage which contains text in multi-orientations.

4.2. **Training.** To better train our model, we have used the transfer learning in order to initialize the Darknet53 backbone kernels. Our purpose is to avoid wasting time by training the backbone layers from scratch. This initialization will also help us to take advantage of the previous training process performed on other dataset. The layers weights have been obtained by training the darknet53 backbone to detect objects other than text. So to get better results, we choose to fix its earlier layers and to train the rest.

During the training step, we have used the ICDAR 2017 dataset. The images to be fed to our model have a width and height of (512x512) and the batch size is set to 10. As an optimizer we have used Adam with a learning rate of 10**(-4) , beta1 and beta2 are respectively 0.9 and 0.999. As a GPU graphic card, we have used the Nvidia Tesla P100-PCIE with 16GB of RAM.

4.3. **Evaluation Protocol.** To evaluate our model, we calculate its precision, recall and F1-score:

$$Recall = \frac{TP}{TP + FN} \tag{10}$$

$$Precision = \frac{TP}{TP + FP} \tag{11}$$

$$F1 - score = \frac{2\, Recall \times Precision}{Recall + Precision} \tag{12}$$

Where TP,FP and FN represent respectively predictions that are evaluated as True positive, False positive and False negative. Precision serves to get information about the fraction of relevant boxes among all the predicted ones. Recall, on the other hand, gives the fraction of relevant boxes that have been predicted correctly among all the relevant ones. To classify a prediction as TP, FP, or FN, we use the IOU metrics. A prediction is seen as a True Prediction (TP) when the overlap between the predicted box and the

ground truth one is greater than a predefined threshold. Otherwise, it is considered as a False one (FP). When a ground truth box is not detected, it is considered as a False negative (FN).

$$IoU = \frac{AREA(G \cap P)}{AREA(G \cup P)} \tag{13}$$

Where G is the Ground Truth bounding box and P is the predicted one

4.4. **Results.** Table 1 shows the results of our model compared with the previous state of art papers tested on the ICDAR 2013 Dataset. It clearly shows how our model is competitive and how it has improved the detection performance. Hence, we contend that our model presents a robust and uncomplicated substitute for prevailing conventional frameworks that rely on segmentation masks. By employing a reduced segmentation map, our model requires less processing time for text detection in images. The integration of transposed convolutions assists our model in effectively upsampling feature maps, thereby enhancing its overall performance. Lastly, the incorporation of the Suzuki and Abe's counter algorithm in conjunction with non-maximum suppression mitigates issues of overlapping bounding boxes.

Table 1. Results on ICDAR 2013 Dataset

Approach	Recall	Precision	F1-Score
Liang et al.[18]	68.0	76.0	72.0
Zhao et al.[19]	63.7	62.1	62.88
Mi et al.[20]	54.2	51.7	52.92
Liu et al.[21]	53.63	48.3	50.82
Shivakumara et al [22]	55.9	52.0	53.87
TextBoxes++ [23]	74.0	86	80
QuadBox[24]	70.0	90.0	79.0
Our Model	80.79	79.49	79.99

Fig.3 shows how our model is accurate. It is able to successfully detect text in a horizontal orientation even if it has a small size. On the other, we must underline some limitations since our model still makes some false predictions. We observe that in some cases, it hasn't detected the whole word. In some others, it finds some difficulties when the text is inclined

5. **Conclusion.** Scene text detection remains an important and difficult research domain. In our work we have tried to combine a fully connected architecture to produce a prediction score map then use it as an entry to a graph algorithm to generate the bounding boxes coordinates. The obtained results are promising and should be taken in consideration to improve the model and extend it to detect text in multi-orientation.

Acknowledgement. This work was supported by the Ministry of Higher Education, Scientific Research and Innovation, the Digital Development Agency (DDA) and the CNRST of Morocco (Alkhawarizmi/2020/01).

Conflict of Interest Statement. The authors declare that there is no conflict of interest

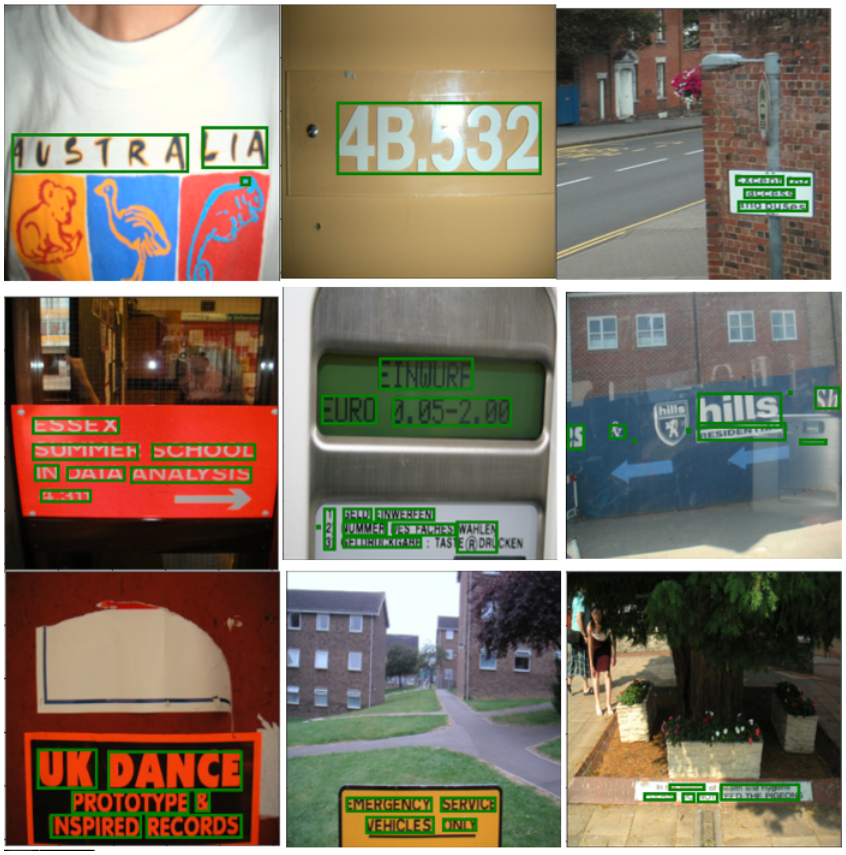

FIGURE 3. Samples of detections made by our model. The images are taken from ICDAR 2013 Dataset

6. **About Data Availability Statements.** The datasets analysed during the current study are available in the the International Conference on Document Analysis and Recognition (ICDAR) repository.

- **ICDAR 2013 :** `https://rrc.cvc.uab.es/?ch=2`
- **ICDAR 2017 :** `https://rrc.cvc.uab.es/?ch=8#`

References

[1] Wang, K., Babenko, B., & Belongie, S. (2011). End-to-end scene text recognition. In *2011 IEEE international conference on computer vision (ICCV)*, (pp. 1457–1464). IEEE.
[2] Coates, A., Carpenter, B., Case, C., Satheesh, S., Suresh, B., Wang, T., et al. (2011). Text detection and character recognition in scene images with unsupervised feature learning. In *2011 international conference on document analysis and recognition (ICDAR)* (pp.440–445). IEEE.
[3] H. Y. Darshan, K. Gopalkrishna and H. raju, "Text detection and recognition using camera based images," in *Proc.3rd Int. Conf. on Frontiers of Intellegent Computing (FICTA)*, Cham, Switzerland, vol. 14, pp. 573–581, 2014.
[4] Huang, Z. Lin, J. Yang, and J. Wang. Text localization in natural images using stroke feature transform and text covariance descriptors. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1241–1248, 2013.
[5] B. Epshtein, E. Ofek, and Y. Wexler, "Detecting text in natural scenes with stroke width transform," in *Computer Vision and Pattern Recognition (CVPR)*, 2010 IEEE Conference on. IEEE, 2010, pp. 2963–2970.

[6] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide-baseline stereo from maximally stable extremal regions," *Image & Vision Computing*, vol. 22, no. 10, pp. 761–767, 2004.
[7] Yu, W., Liu, Y., Hua, W., Jiang, D., Ren, B., & Bai, X. (2023). Turning a clip model into a scene text detector. *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. https://doi.org/10.1109/cvpr52729.2023.00674.
[8] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks." *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 39, no. 6, p. 1137, 2017.
[9] Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: Unified, real-time object detection. In: *IEEE conference on computer vision and pattern recognition*, pp 779–788
[10] Long J, Shelhamer E, Darrell T (2015) Fully convolutional networks for semantic segmentation. In: IEEE international conference on computer vision and pattern recognition, pp 3431–3440
[11] Zhou, X., Yao, C., Wen, H., Wang, Y., Zhou, S., He, W., & Liang, J. (2017). EAST: An efficient and accurate scene text detector. In *The IEEE conference on computer vision and pattern recognition (CVPR)*.
[12] Ronneberger O., Fischer P., Brox T. (2015) U-Net: Convolutional Networks for Biomedical Image Segmentation. In: Navab N., Hornegger J., Wells W., Frangi A. (eds) *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*. MICCAI 2015. Lecture Notes in Computer Science, vol 9351. Springer, Cham. *https* : $//doi.org/10.1007/978 - 3 - 319 - 24574 - 4_28$
[13] J. Redmon and A. Farhadi, "YOLOv3: An incremental improvement,"
2018, arXiv:1804.02767. [Online]. Available: http://arxiv.org/abs/1804.02767
[14] Ma Yi-de, Liu Qing, and Qian Zhi-Bai. Automated image segmentation using improved pcnn model based on cross-entropy. In *Proceedings of 2004 International Symposium on Intelligent Multimedia, Video and Speech Processing*, 2004., pages 743–746. IEEE, 2004.
[15] Carole H Sudre, Wenqi Li, Tom Vercauteren, Sebastien Ourselin, and M Jorge Cardoso. Generalised dice overlap as a deep learning loss function for highly unbalanced segmentations. In *Deep learning in medical image analysis and multimodal learning for clinical decision support*, pages 240–248. Springer, 2017
[16] N. Nayef, F. Yin, I. Bizid, H. Choi, Y. Feng, D. Karatzas, Z. Luo, U. Pal, C. Rigaud, J. Chazalon, W. Khlif, M. M. Luqman, J.-C. Burie, C.-L. Liu, and J.-M. Ogier, "Icdar2017 robust reading challenge on multi-lingual scene text detection and script identification - rrc-mlt," in *ICDAR*, 2017.
[17] Karatzas D, Shafait F, Uchida S, Iwamura M, Bigorda LG, Mestre SR, Mas J, Mota DF, Almazan JA, De Las Heras LP (2013) ICDAR 2013 robust reading competition. In: *12th international conference on document analysis and recognition*, pp 1484_1493
[18] G. Liang, P. Shivakumara, T. Lu, and C. L. Tan, "Multi-spectral fusion based approach for arbitrarily oriented scene text detection in video images," *IEEE Trans. Image Process.*, vol. 24, no. 11, pp. 4488-4501, Nov. 2015.
[19] X. Zhao, K.H. Lin, Y. Fu, Y. Hu, Y. Liu, and T.S. Huang, "Text from corners: a novel approach to detect text and caption in videos", *IEEE Transactions on Image Processing*, pp. 790–799, 2011.
[20] C. Mi, Y. Xu, H. Lu, X. Xue, "A novel video text extraction approach based on multiple frames", In *Proceedings of International Conference on Image and Signal Processing (ICISP)*, pp.678–682, 2005.
[21] C. Liu, C. Wang, and R. Dai, "Text Detection in Images Based on Unsupervised

Classification of Edge-Based Features" , In *Proceedings of International Conference on Document Analysis and Recognition (ICDAR)*, pp. 610-614, 2005.
[22] P. Shivakumara, R.P. Sreedhar, T.Q. Phan, S.Lu, and C.L.Tan, "Multi-oriented video scene text detection through Bayesian classification and boundary growing", *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 1227–1235, 2012.
[23] M. Liao, B. Shi, and X. Bai, "TextBoxes++: A single-shot oriented scene text detector," *IEEE Trans. Image Process.*, vol. 27, no. 8, pp. 3676_3690, Aug. 2018.
[24] P. Keserwani, A. Dhankhar, R. Saini and P. P. Roy, "Quadbox: Quadrilateral Bounding Box Based Scene Text Detection Using Vector Regression," in *IEEE Access*, vol. 9, pp. 36802-36818, 2021, doi: 10.1109/ACCESS.2021.3063030.
[25] Satoshi Suzuki and others. Topological structural analysis of digitized binary images by border following. *Computer Vision, Graphics, and Image Processing*, 30(1):32–46, 1985.
[26] Minghui Liao, Zhaoyi Wan, Cong Yao, Kai Chen, and Xi-ang Bai. Real-time scene text detection with differentiable binarization. In *AAAI*, pages 11474–11481, 2020.
[27] Ma J, Shao W, Ye H, Wang L, Wang H, Zheng Y, Xue X (2018) Arbitrary-Oriented Scene Text Detection via Rotation Proposals. *IEEE TRANSACTIONS ON MULTIMEDIA*, 20(11):3111–3122. https:// doi. org/ 10. 1109/ TMM. 2018. 28180 20. arxiv: 1703. 01086
[28] Liao M, Shi B, Bai X, Wang X, Liu W (2017) TextBoxes: A fast text detector with a single deep neural network. In: *31st AAAI Conference on Artificial Intelligence*, AAAI 2017, pp 4161–4167
[29] Najam, R.; Faizullah, S. Analysis of Recent Deep Learning Techniques for Arabic Handwritten-Text OCR and Post-OCR Correction. *Appl. Sci.* 2023, 13, 7568. https://doi.org/10.3390/app13137568

Photo and Bibliography.

PHD student at Hassan 2 University in Casablanca - FTSM. has got a master's degree at Abdou Chouaib Doukali University in El Jadida in 2008. currently working as assistant, associate computer science professor in preparatory classes at Ibn Tamiya Engineering Schools. interested by deep learning, computer vision for text detection and recognition. has already published a research paper with the title "LiteNet a novel approach for traffic sign classification using a lightweight architecture" in WITS 2020, Proceedings of the 6th International Conference on Wireless Technologies, Embedded, and Intelligent Systems. Email : naimsoufiane@gmail.com

Noureddine Moumkine received the Ph.D. degrees from the Faculty of Science Semlalia of Cadi Ayyad University of Marrakech, in 2006. He has been a research professor at Hassan II University in Casablanca – FSTM since (2007) His research activities focus mainly on the quality of service of NGN networks and artificial intelligence, in particular image classification and recognition. Noureddine Moumkine is a member of the Artificial Intelligence & Data Science research team of the Mathematical computer science and Applications laboratory at Hassan 2 University in Casablanca. Email : noureddine.moumkine@fstm.ac.ma