Driver License Field Detection using Real-Time Deep Networks

Chun-Ming Tsai[1*], Jun-Wei Hsieh[2], Ming-Ching Chang[3], and Yu-Chen Lin[1]

[1] Department of Computer Science, University of Taipei, Taiwan
[2] College of Artificial Intelligence and Green Energy, National Chiao Tung University, Taiwan
[3] Department of Computer Science, University at Albany, State University of New York, USA
cmtsai2009@gmail.com, jwhsieh@nctu.edu.tw, mchang2@albany.edu, sky81915@hotmail.com

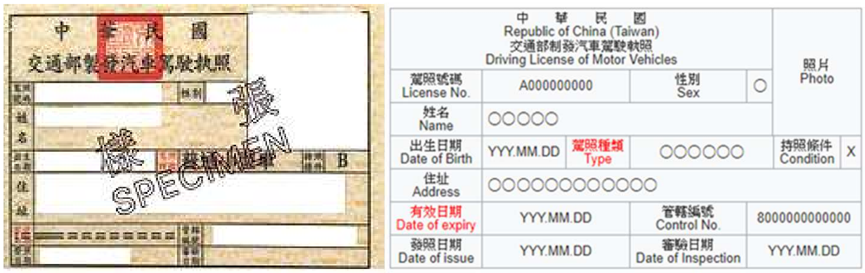
Abstract. We present an automatic system for real-time visual detection and recognition of multiple driver's license fields using an effective deep YOLOv3 detection network. Driver licenses are essential Photo IDs frequently checked by law enforcement and insurers. Automatic detection and recognition of multiple fields from the license can replace manual key-in and significantly improve workflow. In this paper, we developed an Intelligent Driving License Reading System (IDLRS) addressing the following challenging problems: (1) varying fields and contents from multiple types and versions of driver licenses, (2) varying capturing angles and illuminations from a mobile camera, (3) fast processing for real-world applications. To retain high detection accuracy and versatility, we propose to directly detect multiple field contents in a single shot by adopting and fine-tuning the recent YOLOv3-608 detector, which can detect 11 fields from the new Taiwan driver license with accuracy of 97.5%. Our approach does not rely on text detection or OCR and outperforms them when tested with large viewing angles. To further examine such capability, we perform evaluations in 4 large tilting view configurations (top, bottom, left, right), and achieve accuracies of 93.3%, 90.2%, 97.5%, 94.3%, respectively.

Keywords: Field Detection, Driver License, OCR, Deep Learning, YOLOv3.

1. Introduction

Driver licenses are essential Photo IDs frequently checked by law enforcement, government, and insurers. Automatic detection and recognition of multiple fields from the license can replace manual key-in and significantly improve the workflow in the real-world usage scenario. Currently, Taiwan law enforcement uses the M-Police Mobile Computer System App to query license numbers and legality and routine checks. This process and workflow is hard to automate *e.g.* for integration with next-generation smart city infrastructure. When reading from a mobile camera, field texts and numbers can appear to be very small. The lighting condition can be extremely dark or bright. Other circumstances such as dynamic vigilance and alerts need to be constantly maintained by police officers, thus such automatic Intelligent Driver License Reading System (IDLRS) must be effective and easy enough to use for practitioners. The use case of IDLAS includes photo ID check, age check for liquor purchase, package receiver check, insurance claim at a car accident, etc.

Problem Setup. Fig. 1 shows an example of the new Taiwan driver license, which we aim to recognize and process. In Fig. 1(a), the front face consists of a table of fixed template containing 11 fields: “駕照號碼”, “性別”, “姓名”, “出生日期”, “駕照種類”, “持照條件”, “住址”, “有效日期”, “管轄編號”, “發照日期”, and “審驗日期.” Fig. 1(b) shows the English translation of the same information: *License No., Sex, Name, Date of Birth, Type, Condition, Address, Date of Expiry, Control No., Date of Issue, and Date of Inspection.* In this paper, we focus on detecting (identifying and localizing) all 11 fields, such that the results can be directly used for field content recognition, *e.g.* by OCR.



中 華 民 國 Republic of China (Taiwan) 交通部制發汽車駕駛執照 Driving License of Motor Vehicles				照片 Photo	
駕照號碼 License No.	A000000000	性別 Sex	○		
姓名 Name	○○○○○				
出生日期 Date of Birth	YYY.MM.DD	駕照種類 Type	○○○○○○	持照條件 Condition	X
住址 Address	○○○○○○○○○○○○				
有效日期 Date of expiry	YYY.MM.DD	管轄編號 Control No.	800000000000		
發照日期 Date of issue	YYY.MM.DD	審驗日期 Date of Inspection	YYY.MM.DD		

(a) (b)

Fig. 1. Overview of driver license field detection and content recognition. In this example, the new Taiwan driver license consists of fixed templates of tables containing multiple fields in the front. (a) The front face of the new driving license in Taiwan. (b) The field names and field contents in the driving license table.

To ensure the real-world use for law enforcement, insurers, and practitioners, such automatic license reader much be able to detect and identify multiple license fields from a mobile camera in real time. Detailed functions of this system pipeline include: (i) capturing a driver license image, (ii) detecting field contents, (iii) classifying and recognizing field contents. Challenges in this pipeline include: (1) varying fields and contents from multiple types and versions of driver licenses, (2) varying capturing angles and illuminations from a mobile camera, (3) fast processing for real-world applications.

Recent Deep Learning (DL) methods have shown extraordinary performance in many computer vision tasks, including object detection [2-4]. The main reason behind this superior performance is that the deep network automatically learns discriminative features directly from the training data. To retain high detection accuracy and versatility, we propose to directly detect multiple field contents in a single shot by adopting and fine-tuning a widely-used deep object detection network, namely the You Only Look Once (YOLO) v3-608 detector [4]. YOLOv3 operates based on making prediction from performing regressions (rather than classifications of) anchor boxes in the network, which is both fast and accurate. Detection is performed at multiple scales in one shot (that can make predictions from 10x more number of bounding boxes compared to YOLO v2), thus it is good at detecting small objects in a large field-of-view.

To the best of our knowledge, our work is the first on adopting the cutting-edge DL visual object detector toward automatic license reader. The training of the proposed YOLOv3 pipeline is performed based on a much smaller driver license dataset, when compared to common use of YOLOv3 in standard datasets such as COCO or PASCAL VOC [5].

In this paper, we only focus on the detection of all 11 types of license fields as shown in Fig. 1. Since the table format of all fields in the driver license table are fixed (while the contents of the fields can vary case-by-case), our proposed approach of first detection all fixed fields is more effective than performing scene text recognition or OCR on a character-by-character level. We focus more on improving the accuracy and versatility in handling large viewing angles and luminance changes when designing and evaluation of our framework.

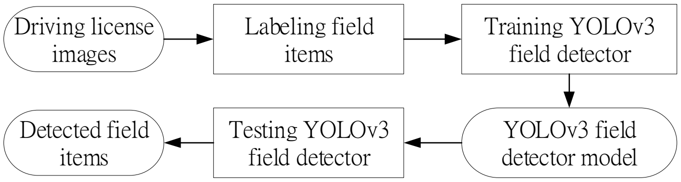

Driving license images
Labeling field items
Training YOLOv3 field detector
YOLOv3 field detector model
Testing YOLOv3 field detector
Detected field items

Fig. 2. Pipeline of works performed in this study.

Fig. 2 shows a flowchart of works performed in this study, which includes the following steps: First, 11 field items are labeled from images of the front face of driving licenses. Second, YOLOv3 is used to train field detector from the collected images. Third, the YOLOv3 field detection results are tested. Finally, we confirm that the 11 fields in the front face driving license were accurately detected.

The rest of this paper is organized as follows. Section 2 describes the related works in visual document processing from images, particularly on table field detection and content understanding. Section 3 describes the design and training of the proposed driver license field detector based on YOLOv3. Experimental results and discussion are shown in Section 4, and conclusion is given in Section 5.

2. Related Works

As mobile devices with cameras are becoming ubiquitous, mobile digital cameras and smart edge devices have several advantages over flatbed scanners for document processing in terms of portability, fast response, and non-contact requirements [6]. Smart edge devices can nowadays perform real-time, online processing of images and data, as well as connect to cloud services more securely.

We focus the survey of the detection and recognition of the tabular formats in document processing, as tables are the essential element of documents for presenting structural information. Recognition and analysis of tables have been an active area of research. We organize the survey of prior works into three main categories: (1) table detection, (2) table layout recognition, and (3) table content understanding [10]. Table detection focuses on the identification and localization of table boundaries in a document image. Table recognition looks for internal table structures such as rows, columns and table cells. Table understanding is on extracting semantic meanings of the individual table contents. In Section 2.4 we survey Convolutional Neural Network (CNN) based methods, as these end-to-end deep learning methods are closely related in terms of similarity and design.

2.1 Table Detection in Documents

Conventional table detection methods perform line detection, corner detection, or character detection on the given document images, where the orientation of the tables is usually assumed to be known. Seo *et al.* [6] proposed a junction-based table detection in camera-captured document images. Junctions of line intersections are first detected in order to locate the corners of cells and connectivity inference. This method is only applicable for tables with horizontal and vertical ruling lines.

The work of [7] determines which lines in the document are likely belong to the tables of interest. The deep-learning method of Gilani *et al.* [8] consist of a Region Proposal Network followed by a fully connected Faster-RCNN network for table detection. It performs well on multiple document types (research papers and magazines) with varying layouts, and out-performs the popular Google Tesseract OCR on the publicly available UNLV dataset.

2.2 Table Layout Recognition

The recognition of table structure is essential for extracting its contents [10]. Table layout recognition methods focus on identifying the table structures and determining sub-structures of the tables. Hassan and Baumgartner [9] investigated table structures in three categories: (1) tables with both horizontal and vertical ruling lines, (2) tables with only horizontal lines, and (3) tables without ruling lines. This method can recognize spanning rows, spanning columns, and multi-line rows. Their method is effective in converting a wide variety of tabular structures into HTML for information extraction.

In contrast, OCR based approaches take an alternative route, by relying on detecting and recognize characters directly (without recognizing the actual table structure) for layout extraction. The deep learning network pipeline in [10] recognizes table contents in heterogeneous document images, where the textual contents are classified into table or non-table elements, followed by a contextual post processing. However, the exact table layout is still unknown after content extraction.

2.3 Table Content Understanding

Table understanding has been studied actively since the blooming of Big Data with huge volumes of tabular data in Web and PDF format [12].

Göbel *et al.* [11] evaluated table understanding methods in terms of (1) table detection, (2) table structure recognition, and (3) functional analysis for PDF understanding. Implementations and evaluations of each task are provided. The ICDAR 2013 Table Competition [12, 13] focused on table location and table structure recognition, which attracted both academic and industrial participations. Results show that the best performing systems can achieve average accuracies in the range of 84% to 87%.

2.4 CNN-based Text Detection

Convolutional Neural Networks (CNNs) are widely used in scene text detection. The fully-convolutional regression network in [13] directly detects scene texts from natural images, where the network is trained using synthetic images containing texts in clutters overlaid with backgrounds. The Fully Convolutional Network (FCN) of Zhang *et al.* [14] detects texts from scene images, where text regions of individual characters are predicted and segmented out, and then the centroid of each character are estimated for the spatial understanding and recognition of the texts. This method can handle texts in multiple orientations, languages and fonts. Jaderberg *et al.* [15] localize and recognize scene texts based on an object-agnostic region proposal mechanism for detection and a CNN classification. Their model was trained solely on synthetic data without the need of manual labelling.

The Deep Matching Prior Network (DMPNet) [16] is CNN-based that can detect text with tighter quadrangle with F-score of 70.64%. A shared Monte-Carlo module computes the polygonal areas that localize texts with quadrangle fast and accurately. A *smooth Ln loss* is used to moderately adjust the predicted bounding box. The rotation-based framework in [17] is built upon a region-proposal-based architecture that can handle arbitrary-oriented texts in natural scenes effectively. The FCN model in [18] can detect and local a potentially large number of texts in the image view. Three strategies (based on the detection of boxes, corners, or left sides) are used to improve detection precision on a broad range of documents.

3 YOLOv3 Driving License Field Detector

We perform driver license understanding by directly detect the multiple types of fields using the YOLOv3 object detector. These detected fields can be used for field content recognition. In comparison to other document understanding approaches surveyed in Section 2, our approach bypasses the complex table layout recognition issues. Our approach is also more effective than OCR and scene-text-based methods, since we directly leverage the fixed template of the driver license table formats (which can vary between version but the change is rather rare).

3.1 YOLO Object Detector

The You Only Look Once (YOLO) [19] is a popular single-shot object detection network that can achieve real-time performance. The first version (YOLOv1) is developed in 2015, with the idea that object detection is re-framed as a regression problem. The detection of multiple objects can be performed in a single network pass, where the bounding boxes of all detected objects and classification probabilities are produced. The base network is originally GoogLeNet, and later VGG based Darknet implementation is also available. The input image is first divided into a grid of cells, such that object bounding boxes and class probabilities in each cell can be directly predicted. The pipeline includes a post-processing, merging, and non-maximal-suppression, and yields the final prediction. The YOLO sub-sequel, YOLOv2 [20] outperforms YOLOv1 in both accuracy and speed. YOLOv3 [4] is more accurate than YOLOv2 but not faster. There are also fast versions such as the tiny YOLOv3 that runs extremely fast, however with reduced accuracy.

3.2 Driving License Field Labeling

We use LabelImg [21], a publicly available tool, to label the driver license fields. LabelImg is a graphical image annotation tool that can quickly mark object bounding boxes in images. Annotations are saved as XML files in PASCAL VOC format, a format used by ImageNet challenge [22]. We convert the LabelImg .xml file into the Darknet [23] format in the following:

```
<object-class> <center-x> <center-y> <width> <height>
```

The `<object-class>` is an integer representing object class (various driver license fields), ranging from 0 to 10. The `<center-x>` and `<center-y>` are the bounding box center, normalized (divided) by the image width and height, respectively. The `<width>` and `<height>` are bounding box dimensions normalized image width and height, respectively. So these entries are values between 0 and 1.

3.3 Training and Testing YOLOv3 Field Detector

Our YOLOv3-608 field detectors can be trained following the standard steps in [23, 24], including the adjustment of the configuration file (driverLicense-yolov3-608.cfg which stores important training parameters), with slight tuning and modification. For the 416x416 input image, the cell size in the YOLOv3 network is 32x32. Detailed steps are described in the following.

1. We randomly select 70% to 90% from our driver license data as training set in several experiment trials. The remaining is used as the testing set.
2. Configure the settings for 11 field classes and 3 yolo-layers.

$$\# \text{ filters} = (\# \text{ classes} + 5) \times 3 = 48,$$

since there are 3 convolutional layers before each yolo layer. The name file `field.names` should contain the field names of the 11 classes.

3. Data file `driverLicense.data` should contain:

```
classes = 11
train = data/driverLicenseTrain.txt
valid = data/driverLicenseTest.txt
backup = backup/
```

4. Start training:

```
./darknet detector train /path/to/driverLicense.data
/path/to/driverLicense-yolov3-608.cfg ./darknet53.conv.74
–map > /path/to/driverLicenseYolov3-608Train.log
```

5. Upon competition, resulting model `driverLicenseYolov3-608_final.weights` should be produced in the `backup/` directory.
6. We test the YOLOv3-608 field detector on the new Taiwan driver license test sets for performance evaluation.

4 Experimental Results and Discussion

All experiments were performed on a machine equipped with an Intel Xeon E3-1231 V3 @3.40GHz CPU and having 8GB of DDR5 memory on an NVIDIA Ge-Force 1070Ti GPU.

Dataset. The collected driver license data contains 617 licenses in total (Table 1). The number of the training samples is 556 and testing is 61. As aforementioned, we performed evaluation in five tilting viewing angles, including the no-tilt (up-right) angle. The number of data samples for the Positive (Pos) *i.e.* viewing from an up-right angle, Top-to-Bottom (TB), Bottom-to-Top (BT), Left-to-Right (LR), and Right-to-Left (RL) are 369, 20, 48, 79, 101, respectively.

Table 1. Driving license table with five capture angles.

CA	Pos	TB	BT	LR	RL
Number	369	20	48	79	101

We use the Intersection over Union (IoU) and mAP metrics [26] to evaluate the YOLOv3-608 performance. The IoU metric measures the accuracy of an object detector in terms of detected bounding box overlaps with ground-truth box labeling. Mean average precision (mAP) calculates as the mean value of average precisions for each class, where the average precision is area-under-curve (AUC) of Precision-Recall (PR) curve for each threshold for each class. Specifically, the True Positive (TP), False Positive (FP), False Negative (FN), and True Negative (TN) are all calculated for each class. Precision = TP / (TP + FP) measures how accurate are the predictions. Recall = TP / (TP + FN) measures how well all the positives found and is defined as. Accuracy is calculated as (TP + TN) / (TP + TN + FP + FN). Finally, the F-score is calculated as 2 (precision x recall) / (precision + recall).

Table 2 shows the field detection results by using YOLOv3-608 field detector of our collected driving licenses. The accuracy (A), precision (P), recall (R), and F score (F) metrics for the positive (Pos) captured angle are 97.5, 99.7, 97.6, and 98.6, respectively. The accuracy, precision, recall, and F score metrics for from top to bottom (TB) captured angle are 93.3, 99.5, 93.4, and 96.4, respectively. The accuracy, precision, recall, and F score metrics for from bottom to top (BT) captured angle are 90.2, 100, 90.0, and 94.8, respectively. The accuracy, precision, recall, and F score metrics for from left to right (LR) captured angle are 97.5, 99.9, 97.6, and 98.7, respectively. The accuracy, precision, recall, and F score metrics for from right to left (RL) captured angle are 94.3, 99.8, 94.5, and 97.1, respectively.

Table 2. The field detection results of our collected driving licenses.

CA	TP	TN	FP	FN	A	P	R	F
Pos	4189	287	13	104	97.5	99.7	97.6	98.6
TB	211	11	1	15	93.3	99.5	93.4	96.4
BT	550	0	0	61	90.2	100	90.0	94.8
LR	990	21	1	23	97.5	99.9	97.6	98.7
RL	1184	16	3	69	94.3	99.8	94.5	97.1

Figure 3 shows the field detected results for the new Taiwan driver license for viewing from an up-right angle which is detected by our fine-tuned YOLOv3-608 field detector. The left and the right of the Fig. 3 are obtained from the Internet and the subject which simulates the police to see the driver license. As show in the left driver license of the Fig. 3, all fields were all correctly detected, except that the field “Condition” was incorrectly detected as “Control No.” and “Condition”. However, all fields in the right driver license were all correctly detected.

Figures 4-7 show the field detected results for the Taiwan driver licenses which are viewing from top to bottom angle, from bottom to top angle, from left to right, and from right to left, respectively. From the left driver license of these examples, most of the 11 fields in each driver license can be detected by our YOLOv3-608 field detector. However, the field “Condition” was incorrectly detected as “Control No.” in Figs. 4 and 7 and the field “Condition” cannot be detected in Figs. 5 and 6. However, the right real driver license of these examples, all fields were all correctly detected.

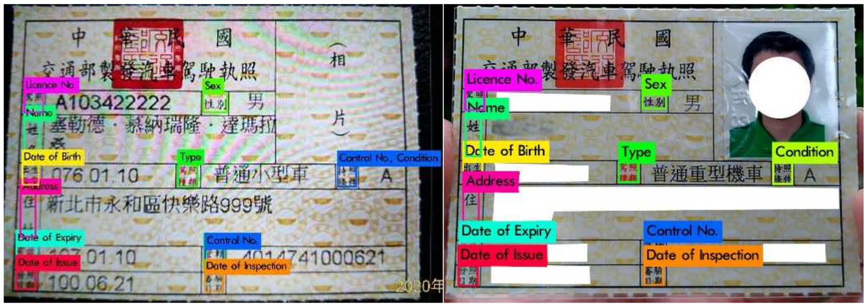


Fig. 3. Example of field detected result for viewing from an up-right angle.

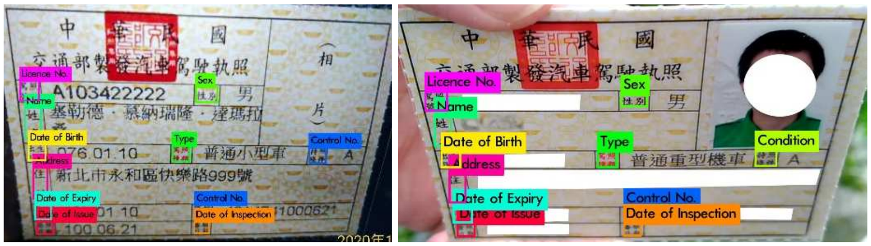


Fig. 4. Example of field detected result for viewing from top to bottom angle.

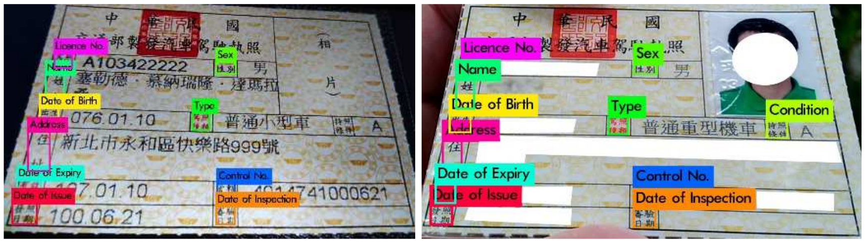


Fig. 5. Example of field detected result for viewing from bottom to top angle.

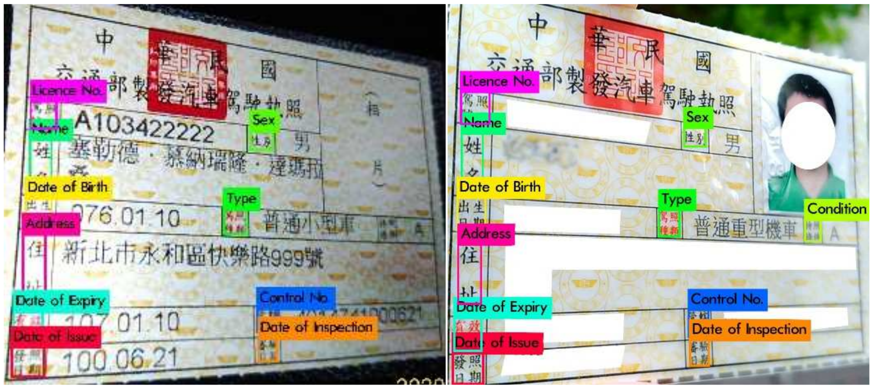


Fig. 6. Example of field detected result for viewing from left to right angle.

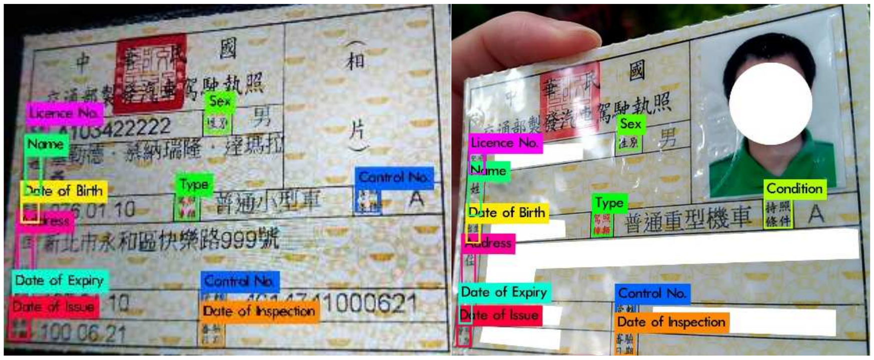


Fig. 7. Example of field detected result for viewing from right to left angle.

5 Conclusions and Future Works

This paper presented an automatic license field detection system based on the deep YOLOv3-608 networks, which are fine-tuned on a newly collected dataset of new Taiwan driver licenses. Detection accuracy of 97.5% is achieved in the test. Additional evaluations are performed on the YOLOv3-608 module with large viewing angles in 4 tilting configurations (top, bottom, left, right), and accuracies of 93.3, 90.2, 97.5, 94.3 are achieved, respectively.

Future works include the collection of a larger multi-national driver license dataset that can be used for model generalization. In addition, newer deep network backbones such as an improved YOLO variation can also be adapted. Finally, the developed field detection capability can be integrated with field content recognition for real-world usage evaluation and deployment.

Acknowledgements

This work is supported by the Ministry of Science and Technology, Taiwan, under Grants MOST 107-2221-E-845-005 – and MOST 108-2221-E-845 -003 -MY3. We thank Walter Slocombe for paper writing improvement.

References

1. Driving license in Taiwan, https://en.wikipedia.org/wiki/Driving_license_in_Taiwan.
2. Ren, S., He, K., Girshick, R. and Sun, J.: Faster R-CNN: Towards real-time object detection with region proposal networks. IEEE Trans. on PAMI, 39(6), 1137-1149 (2017).
3. Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.Y., and Berg, A.C.: SSD: Single Shot MultiBox Detector. In: arXiv preprint arXiv:1512.02325v5, 29 Dec. (2016).
4. Redmon, J. and Farhadi, A.: YOLOv3: An Incremental Improvement. In: arXiv preprint arXiv:1804.02767, April (2018).
5. The PASCAL VOC project, http://host.robots.ox.ac.uk/pascal/VOC/#bestpractice.
6. Seo, W., Koo, H. I., and Cho, N. I.: Junction-based table detection in camera-captured document images. IJDAR, 18(1), 47-57 (2015).
7. e Silva, A. C., Jorge, A., and Torgo, L.: Automatic Selection of Table Areas in Documents for Information Extraction. In: Pires F.M., Abreu S. (eds) Progress in Artificial Intelligence. EPIA 2003. Lecture Notes in Computer Science, vol. 2902. Springer, Berlin, Heidel-berg (2003).
8. Gilani, A., Qasim, S. R., Malik, I., and Shafait, F.: Table Detection Using Deep Learning. In: 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), pp. 771-776, Kyoto (2107).
9. Hassan, T. and Baumgartner, R.: Table Recognition and Understanding from PDF Files. In: Ninth International Conference on Document Analysis and Recognition (ICDAR 2007), pp. 1143-1147, Parana (2007).
10. Rashid, S. F., Akmal, A., Adnan, M., Aslam, A. A. and Dengel, A.: Table Recognition in Heterogeneous Documents Using Machine Learning. In: 14th IAPR International Conference on Document Analysis and Recognition (ICDAR), pp. 777-782, Kyoto (2017).
11. Göbel, M., Hassan, T., Oro, E. and Orsi, G.: A methodology for evaluating algorithms for table understanding in PDF documents. In: ACM Symposium on Document Engineering, pp. 45–48 (2012).
12. Göbel, M., Hassan, T., Oro, E. and Orsi, G.: ICDAR 2013 Table Competition. In: 12th International Conference on Document Analysis and Recognition, pp. 1449-1453. Washington, DC (2013).
13. Gupta, A., Vedaldi, A., Zisserman, A.: Synthetic data for text localization in natural images. In: IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas (2016).
14. Zhang, Z., Zhang, C., Shen, W., Yao, C., Liu, W., Bai, X.: Multioriented text detection with fully convolutional networks. In: IEEE Conference on Computer Vision and Pattern Recognition, Las Vegas (2016).
15. Jaderberg, M., Simonyan, K., Vedaldi, A., Zisserman, A.: Reading text in the wild with convolutional neural networks. Int J Comput Vis 116(1), 1–20 (2016).
16. Liu, Y. and Jin, L.: Deep matching prior network: toward tighter multioriented text detection. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2, p. 8, Honolulu (2017).

17. Ma, J., Shao, W., Ye, H., Wang, L., Wang, H., Zheng, Y., Xue, X.: Arbitrary-oriented scene text detection via rotation proposals. IEEE Transactions on Multimedia 20(11), 3111-3122 (2018).
18. Moysset, B., Kermorvant, C. and Wolf, C.: Learning to detect, localize and recognize many text objects in document images from few examples. IJDAR 21(3), 161-175 (2018).
19. Redmon, J., Divvala, S., Girshick, R. and Farhadi, A.: You Only Look Once: Unified, Real-Time Object Detection. In: arXiv preprint arXiv:1506.02640v5, May (2016).
20. Redmon, J. and Farhadi, A.: YOLO9000: Better, Faster, Stronger. In: arXiv preprint arXiv:1612.08242v1, Dec (2016).
21. Tzutalin. LabelImg. Git code, https://github.com/tzutalin/labelImg
22. ImageNet., http://www.image-net.org
23. Darknet: Open Source Neural Networks in C, https://pjreddie.com/darknet/
24. AlexeyAB/darknet, https://github.com/AlexeyAB/darknet#how-to-train-to-detect-your-custom-objects