YOLO-WS: A Novel Method for Webpage Segmentation

Li Dai
daili@stu.xju.edu.cn
Xinjiang University
Urumqi, Xinjiang, China

Zunwang Ke
Xinjiang University
Urumqi, Xinjiang, China
kzwang@xju.edu.cn

Wushour Silamu
Xinjiang University
Urumqi, Xinjiang, China
wushour@xju.edu.cn

ABSTRACT

To address the limitations of traditional heuristic and machine learning-based webpage segmentation algorithms in feature extraction performance and efficiency, we propose a webpage segmentation method based on deep learning object detection. Specifically, we propose a webpage segmentation method named YOLO-WS based on the YOLOv5 model. We optimized and improved the YOLOv5 model's network structure, loss function, and post-processing for webpage segmentation tasks, and then use transfer learning to train YOLO-WS on the improved model. Experimental results show that YOLO-WS achieves good performance in web page segmentation tasks.

CCS CONCEPTS

• **Information systems** → **Data extraction and integration**; • **Applied computing** → *Document analysis.*

KEYWORDS

webpage segmentation, document layout analysis, object detection, deep learning

ACM Reference Format:
Li Dai, Zunwang Ke, and Wushour Silamu. 2023. YOLO-WS: A Novel Method for Webpage Segmentation. In *2023 4th International Conference on Computing, Networks and Internet of Things (CNIOT '23), May 26–28, 2023, Xiamen, China.* ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3603781.3603862

1 INTRODUCTION

With the development of the global internet industry, the internet has become the most important source of data and information in today's era. The internet is a huge container of data, and a vast amount of information on the internet is organized in the form of webpages. However, the purpose of designing and writing webpages is to enable human readers to read and access information more effectively, and the data organization methods of most webpages are not strictly compliant with semantic standards. Currently, most webpage segmentation algorithms mainly rely on traditional manual feature detection algorithms and machine learning algorithms. Applying deep learning-based object detection methods to webpage segmentation tasks has great potential. We propose to introduce object detection algorithms into webpage segmentation by transforming the webpage segmentation task into an object detection task. Based on the YOLOv5 model[4], we optimized it for webpage segmentation tasks from multiple aspects, such as network structure, loss function, post-processing method, and training method. For example, in the post-processing step, we applied a weighted fusion method to generate candidate boxes in the improved YOLOv5 algorithm. By replacing the NMS or Non-Maximum Suppression method with a weighted fusion method that can comprehensively consider the prediction results, this method can improve the performance of webpage segmentation. In addition, we introduced transfer learning to train the model by using a larger document layout analysis task dataset named DocLayNet[8] as the source domain to obtain source model weights and then transfer the weights to train the model on the Webis-WebSeg-20[5] dataset for webpage segmentation in the target domain. Finally, we obtained a well-performing webpage segmentation algorithm model, successfully introducing object detection algorithms into webpage segmentation tasks and achieving good results.

CNIOT '23, May 26–28, 2023, Xiamen, China

ACM ISBN 979-8-4007-0070-5/23/05…$15.00
https://doi.org/10.1145/3603781.3603862

2 YOLO-WS

2.1 Improved network structure for webpage segmentation

The YOLOv5 model uses the CSPDarknet53 network as the backbone network for feature extraction and combines PANet and SPP for feature fusion. It also uses three prediction networks with different sizes, namely 52x52, 26x26, and 13x13, to detect large, medium, and small objects respectively.

Attention module. To improve the performance of the YOLOv5 algorithm in webpage segmentation tasks, we added a coordinate attention[3] module to the output end of the C3 module in the backbone network of YOLOv5. This mechanism aims to improve the network's attention to the target content blocks of interest in the input webpage data and focus on useful information. The overview of the structure of Coordinate Attention is shown in Fig. 1.

Activation function optimization. Next, we improved the ability of the network to extract spatial semantic features in webpages by replacing the SiLU activation function with the FReLU[6] activation function in the convolution layers. The FReLU function extends the PReLU/ReLU function by adding a spatial condition to create a two-dimensional activation function, which is designed for visual processing tasks and can significantly improve the efficiency of such tasks. The formula for this activation function is shown in Eq. (1), where $T(x)$ is a context feature extractor.

$$y = \max(x, T(x)) \tag{1}$$

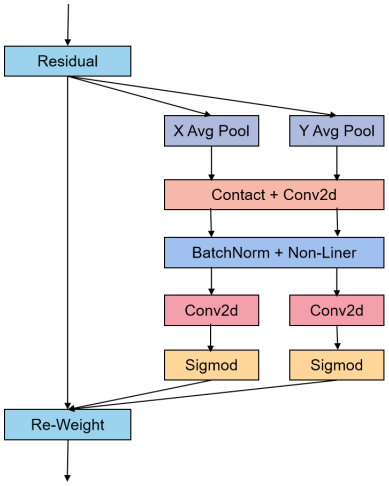

Figure 1: Overview of Coordinate Attention structure.

The comparison between the ReLU function and its visualization is shown in Fig. 2. FReLU replaces the conditional part of the max function in ReLU with a two-dimensional funnel condition, which improves the insensitivity of the activation function to space and strengthens the ability of regular convolution to capture complex visual layouts, making the model capable of pixel-level modeling.

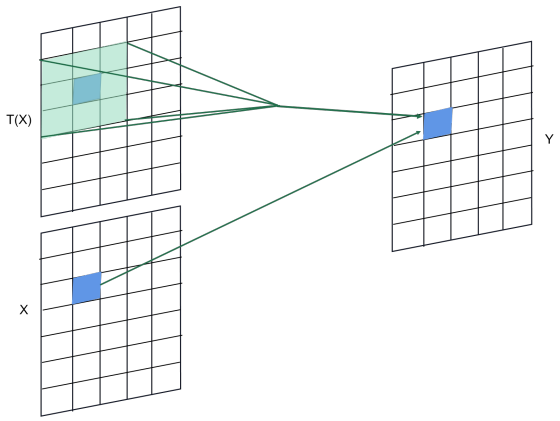

Figure 2: FReLU.

The network structure after the above improvements is shown in Fig. 3.

Optimized PANet. Analyzing the network structure of YOLOv5, it can be found that in order to increase the performance of its predictions, the network deepens the overall network layers, mainly reflected in the PANet and SPP modules of the feature fusion network. In this chapter, we improved the PANet by combining skip connections[10] to preserve the initial information in the output layer and prevent network degradation and reduce information loss, thereby improving the network performance. Fig. 4 shows a schematic diagram of the improvement of PANet by adding skip connections.

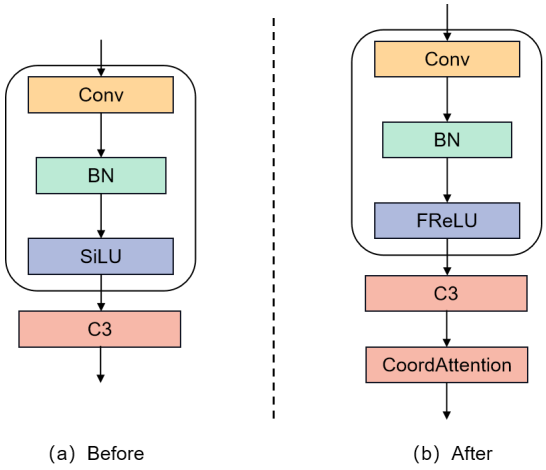

Figure 3: Comparison of network modules before and after improvement.

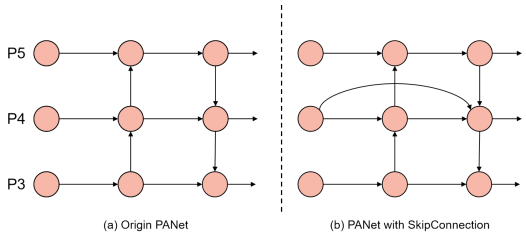

Figure 4: Comparison of network modules before and after improvement.

2.2 Improved loss function for webpage segmentation

To achieve better performance optimization, we replaced the CIoU loss function in the original YOLOv5 model with the EIoU[11] loss function in our proposed YOLO-WS model. The EIoU loss function consists of three components: width-height loss, overlap loss, and center distance loss. The overlap loss and center distance loss reuse the methods in CIoU, but for calculating the width-height loss, EIoU directly minimizes the difference between the width and height of the target box and the anchor box. This improvement makes the EIoU loss function converge faster than CIoU. Additionally, EIoU incorporates Focal Loss on top of CIoU to optimize the imbalance of samples in the bounding box regression task. By reducing the weight of anchor boxes with low intersection-over-union with the target box during regression, the Focal Loss focuses on high-quality anchor boxes. The formula for Focal Loss is shown in Eq. (2).

$$FL\left(p_t\right) = -\left(1-p_t\right)^{\gamma} \log\left(p_t\right) \tag{2}$$

$$p_t = \begin{cases} p, y = 1 \\ 1-p, \text{ otherwise} \end{cases} \tag{3}$$

If the constant γ in the formula is set to 0, the Focal Loss will degenerate to ordinary cross-entropy loss. The penalty term of EIoU is based on the penalty term of CIoU, the aspect ratio is separated

and the length and width of the anchor box and the target box are calculated separately. The calculation method of EIoU Loss is shown in Eq. (4), where w^c and h^c are the width and height of the minimum bounding rectangle of the predicted bounding box and the ground truth bounding box, and ρ^2 is the Euclidean distance between two points.

$$\begin{aligned} L_{EIoU} &= L_{IoU} + L_{dis} + L_{asp} \\ &= 1 - IoU + \frac{\rho^2\left(b, b^{gt}\right)}{\left(w^c\right)^2 + \left(h^c\right)^2} \\ &+ \frac{\rho^2\left(w, w^{gt}\right)}{\left(w^c\right)^2} + \frac{\rho^2\left(h, h^{gt}\right)}{\left(h^c\right)^2} \end{aligned} \tag{4}$$

2.3 Improved post-processing method

In the post-processing step, the YOLOv5 defaults to using the NMS algorithm or Non-Maximum suppression algorithm to remove duplicate bounding boxes. The NMS algorithm iteratively compares the candidate box with the highest score to other candidate boxes based on the IoU and discards those with low confidence and high IoU with the highest score candidate box. Although NMS can effectively remove duplicate candidate boxes in practical applications, it has some drawbacks. For example, NMS directly discards candidate boxes with overlap greater than the overlap threshold, and if there is a target object in the overlapping area that is actually present but the candidate box is discarded, the detection effect may be reduced.

In order to optimize the selection of candidate boxes in the post-processing step, our YOLO-WS model introduced the weighted box fusion[9] method to handle duplicate candidate boxes with overlapping areas greater than the overlap threshold. The NMS method simply discards some candidate boxes and filters out the final candidate boxes from all candidate boxes, which cannot integrate all the predicted results of the candidate boxes well and still misses some predicted result information, leading to prediction bias. In contrast, the weighted box fusion algorithm does not discard any candidate boxes. In the weighted box fusion, the algorithm assigns a weight to each overlapping candidate box and carries the weight into the calculation. By calculating, the bounding regions of all candidate boxes are fused to obtain a new candidate box. This method fully considers the predicted regions of each candidate box and controls the contribution of each candidate box by assigning weights, which can obtain more accurate predicted regions.

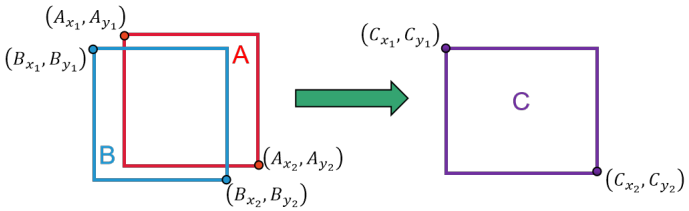

Figure 5: Weighted box fusion.

The process of weighted box fusion is shown in Fig. 5. The new candidate box C is obtained by calculating the weighted coordinates of candidate boxes A and B, and the weight of candidate box C is also calculated by averaging. The fused candidate box C can also continue to participate in subsequent fusion and form a chain of fusion.

The calculation of the region coordinates (C_{x_1}, C_{y_1}),(C_{x_2}, C_{y_2}) for the fused candidate box C is shown in Eq. (5), where (A_{x_1}, A_{y_1}),(A_{x_2}, A_{y_2}) are the upper left and lower right coordinates of candidate box A, (B_{x_1}, B_{y_1}),(B_{x_2}, B_{y_2}) are the upper left and lower right coordinates of candidate box B, and A_w and B_w are the confidence scores of candidate boxes A and B, respectively.

$$\begin{aligned} C_{x_1} &= \frac{A_{x_1} \times A_w + B_{x_1} \times B_w}{A_w + B_w} \\ C_{y_1} &= \frac{A_{y_1} \times A_w + B_{y_1} \times B_w}{A_w + B_w} \\ C_{x_2} &= \frac{A_{x_2} \times A_w + B_{x_2} \times B_w}{A_w + B_w} \\ C_{y_2} &= \frac{A_{y_2} \times A_w + B_{y_2} \times B_w}{A_w + B_w} \end{aligned} \tag{5}$$

The confidence score C_w of the fused candidate box C is calculated as shown in Eq. (6), which is the average of the confidence scores of candidate boxes A and B.

$$C_{y_2} = \frac{A_w + B_w}{2} \tag{6}$$

2.4 Transfer learning

Due to the relatively small scale of datasets for webpage segmentation tasks, such as Webis-WebSeg-20, transferring existing knowledge from similar auxiliary fields can have a positive effect on improving the effectiveness and training speed of webpage segmentation tasks. DocLayNet dataset is a purely visual dataset for document layout analysis, consisting mainly of books, newspapers, and magazines. The features of the article's body, title, illustrations, and other edge features in the document layout analysis task have certain similarities with the partial features of the webpage rendering picture in the webpage segmentation task. For instance, titles in normal books and documents are generally in bold, black font, and appear before large blocks of text. Given the similarities in features between document layout and webpage rendering, we proposed introducing document layout analysis data as a source domain in webpage segmentation tasks to achieve knowledge transfer and faster model convergence for better results.

We propose to introduce DocLayNet data as the source domain and webpage segmentation task data as the target domain. By training a model based on the source domain, we obtain the model weights, which are used as the initial network weights for the webpage segmentation model. Building on this foundation, we continue to train the model using the webpage segmentation dataset to obtain a webpage segmentation model. Fig. 6 illustrates the process.

The specific steps for training the transfer model are as follows:

- Using the YOLO-WS model as the base and training it on the DocLayNet dataset until a highly accurate document layout analysis model is obtained.
- Using the weights obtained from the previous step as the initial weights for the model trained on the Webis-WebSeg-20 dataset. Specifically, certain convolutional layer weights are frozen during the training process.

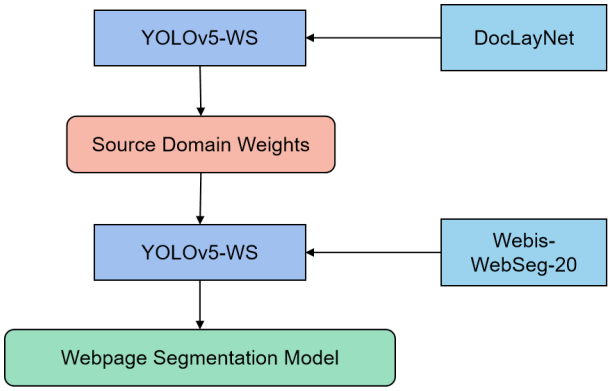

Figure 6: Transfer learning process.

3 MODEL TRAINING AND ANALYSIS

3.1 Dataset

DocLayNet is a large, human-annotated dataset for document layout analysis tasks, consisting of 80,863 diverse and complex document images from various public resources such as finance, science, patents, bids, legal documents, and handbooks. The dataset contains six types of documents, and 11 different class labels for each document image, along with corresponding ground-truth data for each image.

Webis-WebSeg-20 dataset is an open-source webpage segmentation dataset, containing 42,450 manually annotated webpage segmentation samples from 8,490 webpages. The dataset consists of layered samples from 4,824 different websites ranked by Alexa from high to low. The dataset provides data in three modes: DOM documents, edges, and webpage rendering pictures. Additionally, each data sample also has multiple annotation information for segmentation granularity, such as pixels, edges, nodes, and characters.

3.2 Experimental and parameters

For this experiment, the YOLO-WS algorithm model is used as the basis for transfer learning experiments, which are carried out in two modes:

- YOLO-WS model without transfer learning: The YOLO-WS model is trained directly on the Webis-WebSeg-20 dataset to obtain the webpage segmentation model.
- YOLO-WS model with transfer learning: The YOLO-WS model is trained on the DocLayNet dataset for document segmentation tasks to obtain document segmentation weights. These weights are then transferred to the webpage segmentation task with the aim of enhancing model generalization ability and convergence speed.

In the experiment, the default optimizer is set to SGD, with an initial learning rate of 1e-3, a momentum parameter of 0.9, and a weight decay of 0.0005. The warm-up strategy is also used to adjust the initial learning rate, and the input image resolution is set at 512. The experiment runs for 300 epochs, and the batch size is set to 32. The IoU threshold value for post-processing is set at 0.5.

Table 1: Results of ablation experiment

	Model	Precision	Recall	F1-Score
A	YOLOv5s	68.57	57.43	62.51
B	YOLOv5s+CA	71.38	56.22	62.90
C	YOLOv5s+SkipConnection	68.79	60.65	64.50
D	YOLOv5s+CA+SkipConnection	70.87	59.69	64.80
E	YOLO-WS	71.12	60.28	65.25

3.3 Analysis of experimental results

YOLO-WS model without transfer learning. In the experiment, we directly train and carry out experiments on the Webis-WebSeg-20 dataset using our YOLO-WS model. This includes comparing and evaluating the effectiveness of different optimization strategies for the YOLOv5 algorithm, with the aim of detecting the most effective optimization strategy.

The results of the ablation experiments on the YOLO-WS model on the Webis-WebSeg-20 dataset are shown in Table 1. The experiments show that adding Coordinate Attention to the output of the C3 module in the YOLOv5s backbone network and adjusting the network parameters (Model B) can improve the model's detection precision to 71.38%, but it also reduces the model's recall to 56.22%. Model B is improved by combining skip connections in the PANet to preserve the initial unfused information at the output layer and prevent information loss, in order to improve the model's generalization ability. The experiments show that this improvement method can increase the model's recall rate. Model D, which combines the improvement strategies of Models B and C, achieves improvements in both accuracy and recall compared to the standard YOLOv5s model, achieving a precision of 70.87% and a recall of 59.69%. The improved model YOLO-WS proposed in this chapter achieves a precision of 71.11%, a 2.54% improvement over the standard YOLOv5s model, while the recall reaches 60.28%, a 2.86% improvement over the standard YOLOv5s model, with an F1-Score improvement of 2.74%. The results of the ablation experiments demonstrate the effectiveness of the improvement strategies proposed in this chapter. At the same time, the experiments in this section also demonstrate the feasibility and effectiveness of introducing object detection algorithms into webpage segmentation tasks.

YOLO-WS Model with transfer learning. In this experiment, we used the YOLO-WS model as the basis and conducted transfer learning experiments on the DocLayNet and Webis-WebSeg-20 datasets. During the training process of the target domain model, we adopted the method of freezing specific convolutional layers for transfer learning to accelerate the training speed.

Fig. 7 shows the Precision curve of our proposed YOLO-WS model trained directly on the target dataset and the model after weight transfer by incorporating DocLayNet as the source domain. By analyzing the results, it can be seen that the transfer learning approach with frozen convolution layers improves the convergence speed and prediction performance of the model. YOLO-WS-TL, which uses transfer learning on the YOLO-WS model with DocLayNet as the source domain, achieved a 2.9% improvement in

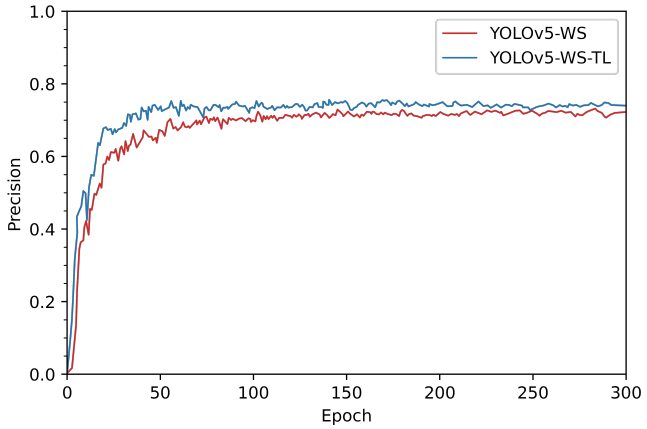

Figure 7: Comparison of precision curves between transferred learning and non-transferred learning models

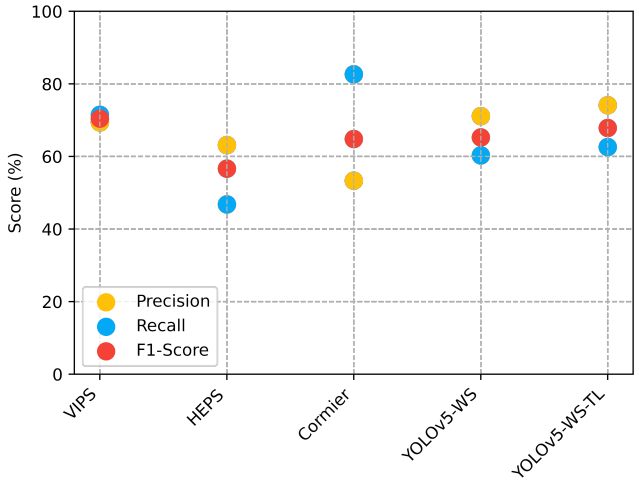

Figure 8: Performance Comparison of Webpage Segmentation Algorithms

Table 2: Performance Comparison of Webpage Segmentation Algorithms

Algorithm	Precision	Recall	F1-Score
VIPS[1]	69.38	71.49	70.41
HEPS[7]	63.12	46.78	56.65
Cormier[2]	53.31	82.62	64.81
YOLO-WS	71.12	60.28	65.25
YOLO-WS-TL	74.11	62.58	67.86

Precision, reaching 74.11%, and a 2.8% improvement in recall, reaching 62.58%, compared to YOLO-WS. The experiment demonstrates that using transfer learning with document layout segmentation task data as the source domain to transfer weights to the target domain of the webpage segmentation task can improve performance, which verifies the effectiveness of this optimization method.

Table 2 and Fig. 8 show the comparison of our proposed model and common webpage segmentation algorithms on the webpage segmentation task. Through analysis, it can be observed that the YOLO-WS algorithm proposed in this chapter, as well as the YOLO-WS-TL model with transfer learning, both perform well in terms of Precision, outperforming other common algorithms. In terms of Recall, they have a certain disadvantage compared to the classic VIPS algorithm and Cormier. et al.'s algorithm. However, compared with all the algorithms, the YOLO-WS-TL model achieves an F1-Score of 67.86%, which is higher than the average. Based on the experiments, we can conclude that our proposed YOLO-WS model has performance above the average of common algorithms on the Webis-WebSeg-20 dataset. This also proves the feasibility of using document layout analysis data to assist webpage segmentation tasks, demonstrating the practical value of the model we constructed for webpage segmentation tasks.

4 CONCLUSION

We have proposed a targeted improvement on the YOLOv5 object detection model, which is suitable for the webpage segmentation task. On the one hand, we have improved the model's ability to extract webpage structural features by introducing coordinate attention mechanisms and FReLU, as well as using skip connections to optimize the model. The experimental results have demonstrated that our model achieves better performance. On the other hand, we have utilized the transfer learning method by using the DocLayNet dataset, as the source domain to transfer knowledge to the target domain of webpage segmentation. This optimization method successfully speeds up the convergence of the model, reduces the training time cost of the target domain model, and improves the performance of the model in the target domain task. This demonstrates the feasibility of using similar tasks as auxiliaries and transferring them as knowledge to the webpage segmentation task. Overall, our proposed algorithmic model and method have practical value in webpage segmentation tasks.

ACKNOWLEDGMENTS

This work is supported by the National Natural Science Foundation of China (Grant No. 61433012) and National Basic Research Program of China (973 Program, 2014CB340506).

REFERENCES

[1] Deng Cai, Shipeng Yu, Ji-Rong Wen, and Wei-Ying Ma. 2003. Vips: a vision-based page segmentation algorithm. (2003).
[2] Michael Cormer, Richard Mann, Karyn Moffatt, and Robin Cohen. [n. d.]. Towards an Improved Vision-Based Web Page Segmentation Algorithm. In *2017 14th Conference on Computer and Robot Vision (CRV)* (2017-05). 345–352. https://doi.org/10.1109/CRV.2017.38
[3] Qibin Hou, Daquan Zhou, and Jiashi Feng. [n. d.]. Coordinate Attention for Efficient Mobile Network Design. arXiv:2103.02907 [cs] http://arxiv.org/abs/2103.02907
[4] Glenn Jocher, Ayush Chaurasia, Alex Stoken, Jirka Borovec, NanoCode012, Yonghye Kwon, Kalen Michael, TaoXie, Jiacong Fang, imyhxy, Lorna, 曾逸夫(Zeng Yifu), Colin Wong, Abhiram V, Diego Montes, Zhiqiang Wang, Cristi Fati, Jebastin Nadar, Laughing, UnglvKitDe, Victor Sonck, tkianai, yxNONG, Piotr Skalski, Adam Hogan, Dhruv Nair, Max Strobel, and Mrinal Jain. [n. d.]. *ultralytics/yolov5: v7.0 - YOLOv5 SOTA Realtime Instance Segmentation.* https://doi.org/10.5281/zenodo.7347926

[5] Johannes Kiesel, Florian Kneist, Lars Meyer, Kristof Komlossy, Benno Stein, and Martin Potthast. [n. d.]. Web Page Segmentation Revisited: Evaluation Framework and Dataset. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management* (New York, NY, USA, 2020-10-19) *(CIKM '20)*. Association for Computing Machinery, 3047–3054. https://doi.org/10.1145/3340531.3412782

[6] Ningning Ma, Xiangyu Zhang, and Jian Sun. [n. d.]. Funnel Activation for Visual Recognition. https://doi.org/10.48550/arXiv.2007.11824 arXiv:2007.11824 [cs]

[7] Tomohiro Manabe and Keishi Tajima. [n. d.]. Extracting logical hierarchical structure of HTML documents based on headings. 8, 12 ([n. d.]), 1606–1617. https://doi.org/10.14778/2824032.2824058

[8] Birgit Pfitzmann, Christoph Auer, Michele Dolfi, Ahmed S. Nassar, and Peter W. J. Staar. [n. d.]. DocLayNet: A Large Human-Annotated Dataset for Document-Layout Analysis. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining* (2022-08-14). 3743–3751. https://doi.org/10.1145/3534678.3539043 arXiv:2206.01062 [cs]

[9] Roman Solovyev, Weimin Wang, and Tatiana Gabruseva. [n. d.]. Weighted boxes fusion: Ensembling boxes from different object detection models. 107 ([n. d.]), 104117. https://doi.org/10.1016/j.imavis.2021.104117

[10] Dongxian Wu, Yisen Wang, Shu-Tao Xia, James Bailey, and Xingjun Ma. [n. d.]. Skip Connections Matter: On the Transferability of Adversarial Examples Generated with ResNets. arXiv:2002.05990 [cs, stat] http://arxiv.org/abs/2002.05990

[11] Yi-Fan Zhang, Weiqiang Ren, Zhang Zhang, Zhen Jia, Liang Wang, and Tieniu Tan. [n. d.]. Focal and Efficient IOU Loss for Accurate Bounding Box Regression. arXiv:2101.08158 [cs] http://arxiv.org/abs/2101.08158