Objects Detection from Digitized Herbarium Specimen based on Improved YOLO V3

Conference Paper · January 2020

DOI: 10.5220/0009170005230529

Abdelaziz Triki
Friedrich Schiller University Jena
4 PUBLICATIONS **10** CITATIONS
SEE PROFILE

Bassem Bouaziz
University of Sfax
48 PUBLICATIONS **1,008** CITATIONS
SEE PROFILE

Walid Mahdi
University of Sfax
93 PUBLICATIONS **580** CITATIONS
SEE PROFILE

Jitendra Gaikwad
Friedrich Schiller University Jena
22 PUBLICATIONS **317** CITATIONS
SEE PROFILE

Objects Detection from Digitized Herbarium Specimen based on Improved YOLO V3

Abdelaziz Triki[1], Bassem Bouaziz[1], Walid Mahdi[1,2] and Jitendra Gaikwad[3]
[1]*MIRACL/CRNS, University of Sfax, Sfax, Tunisia*
[2]*College of Computers and Information Technology, Taif University, Saudi Arabia*
[3]*Friedrich Schiller University Jena, German Centre for Integrative Biodiversity Research (iDiv) Halle-Jena-Leipzig, Leipzig, Germany*

Keywords: Functional Traits, Digitized Plant Specimens, Deep Learning, Improved YOLO V3, Object Detection, Herbarium Haussknecht.

Abstract: Automatic measurement of functional trait data from digitized herbarium specimen images is of great interest as traditionally, scientists extract such information manually, which is time-consuming and prone to errors. One challenging task in the automated measurement process of functional traits from specimen images is the existence of other objects such as scale-bar, color pallet, specimen label, envelopes, bar-code and stamp, which are mostly placed at different locations on the herbarium-mounting sheet and require special detection method. To detect automatically all these objects, we train a model based on an improved YOLO V3 full-regression deep neural network architecture, which has gained obvious advantages in both speed and accuracy through capturing deep and high-level features. We made some improvements to adjust YOLO V3 for detecting object from digitized herbarium specimen images. A new scale of feature map is added to the existing scales to improve the detection effect on small targets. At the same time, we adopted the fourth detection layer by a 4* up-sampled layer instead of 2* to get a feature map with higher resolution deeper level. The experimental results indicate that our model performed better with mAP-50 of 93.2% compared to 90.1% mean IoU trained by original YOLO V3 model on the test set.

1 INTRODUCTION

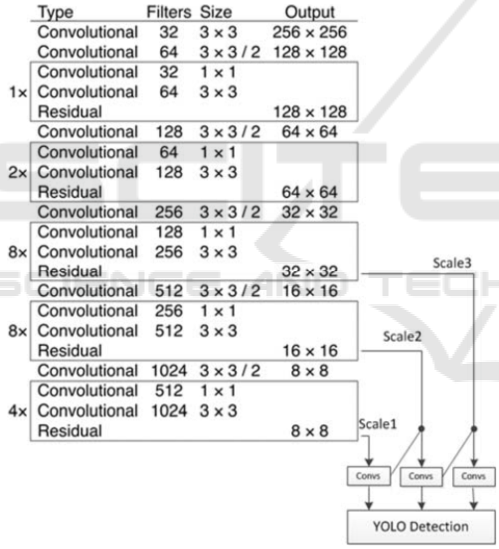
Most of the worldwide herbaria are subscribed in the initiative of the digitization. This process aims at transforming all physical mounted specimens into digital ones. Recent initiative as herbarium Haussknecht of Jena (HHJ) in Germany has started the digitization process and provides access to 30k herbarium images. Each digitized herbarium specimen (DHS) image includes seven classes of objects: plant specimen, scale bar, stamp, color pallet, specimen label, envelope, and bar-code. One challenging task in automated object detection is due to specific issues like occlusions and variations in scale and random placement on herbarium-mounting sheet. Several papers of object detections were built based on handcrafted approaches, which have existed before deep learning era like Haar-like features (Vondrick et al., 2015) (Heisele, 2003) (Viola and Jones, 2004), integrogram (Viola and Jones, 2001), histogram of oriented gradient (HOG) (Felzenszwalb et al., 2010) (Lowe et al., 1999) (Lowe, 2004) (Belongie et al., 2002), Hough Transform (Hussin et al., 2012) and Deformable Part-based Model (DPM) (Felzenszwalb et al., 2008) (Felzenszwalb et al., 2010). These approaches have adopted for various fields of object detection and they are still being used in practice. However, they are time-consuming and not reliable enough for automatic object detection. At present, with the continuous development of computer hardware, modern deep learning models have made incredible progress in computer vision tasks. They are widely used for object detection tasks by detecting objects in complex scenarios. Furthermore, these approaches can be grouped into two categories: two-stage detection and one-stage detection. R-CNN (Girshick et al., 2014) (Regions with CNN features) developed by Ross Girshick in 2014 is one of the first breakthroughs of the use of CNNs in an object detection system which made significant progress in the efficiency of object detection. The proposed algorithm is composed of three main steps: region proposals ex-

Triki, A., Bouaziz, B., Mahdi, W. and Gaikwad, J.
Objects Detection from Digitized Herbarium Specimen based on Improved YOLO V3.
DOI: 10.5220/0009170005230529
In *Proceedings of the 15th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2020)*, pages 523-529
ISBN: 978-989-758-402-2

traction, CNN (Convolutional Neural Networks) feature computation and bounding-box regression. The proposed RCNN uses a selective search algorithm (van de Sande et al., 2011) to extract 2000 region proposals from the input image. Each candidate region proposal fed into CNN to produce features as output. Consider that a large number of overlapping regions take a huge amount of time to train the network, resulting in a waste of computing resources and leads to an extremely slow detection speed. Furthermore, RCNN could lead to the generation of bad candidate region proposals since it uses the selective search algorithm which is a slow and time-consuming process affecting the performance of the network. Hence, to solve some of R-CNN drawbacks, Spatial Pyramid Pooling Networks (SPPNet) (He et al., 2014) was proposed by k. He et al. Unlike the previous CNN models involving a fixed-size of the input image, SPPNet uses a Spatial Pyramid Pooling (SPP) layer allowing a CNN to produce a fixed-length representation where any image sizes can be inputted. In spite of its improvements over RCNN model, there are sitll some disadvantages: (1) the training stage is too slow, (2) SPPNet focuses only on fine-tuning its fully connected layers whereas all previous layers are neglected. In 2015, Ross took into consideration these limitations and has proposed Fast-RCNN (Girshick, 2015), which makes the class classification faster. The input image feeds into a CNN to generate a convolutional feature map. The region of proposals are determined directly from the convolutional feature map where Fast-RCNN integrates a RoI pooling layer to reshape the identified region proposal into a fixed size making the classification faster but it still relies on selective search which can take around 2 seconds per image to generate bounding box proposals. Thus, it has high mAP but it can't meet real-time detection. Faster-RCNN (Ren et al., 2015) replaces a selective search algorithm and integrates an RPN branch networks to predict the region proposals. These solutions have improved the speed of Faster-RCNN but it is still difficult to meet the real-time engineering requirements. Compared with the two-stage detection approaches, the one-stage detection approaches often involves finding the right trade-off between accuracy and computational efficiency. The SSD (Liu et al., 2015) is a common object detection algorithm which performs a single forward pass of the network to locate and identify multiple objects within the input image. Therefore, it achieves good speed efficiency compared with two-shot RPN-based approaches. After continuous iterative improvement of YOLO, Joseph Redmon proposed YOLO V3 (Redmon and Farhadi, 2018) which is three times faster than SSD. For 320x320 images, the detection speed of YOLO V3 can reach 22ms. Considering the variability in size and position of objects within the digitized herbarium specimens images, it is more appropriate to use YOLO V3 as the target detection network because it offers a very fast operation speed with good accuracy to predict the objects within the DHS images. However, YOLO V3 often struggled with small and occluded objects. To address this issue, we proposed an automatic object detection method based on an improved YOLO V3 deep neural network, which is developed, based on the Darknet framework. The proposed approach uses the last four scales of feature maps, which are rich in detail localization information to detect small and occluded objects from the DHS (figure 2). Furthermore, we adopted the fourth detection layer by a 4* up-sampled layer instead of 2* to get a feature map with higher resolution and lower level. The improved YOLO V3 was trained on data provided by the herbarium Haussknecht in Germany. The experimental results show very high detection speed and accuracy under the same detection time.

2 PROPOSED APPROACH

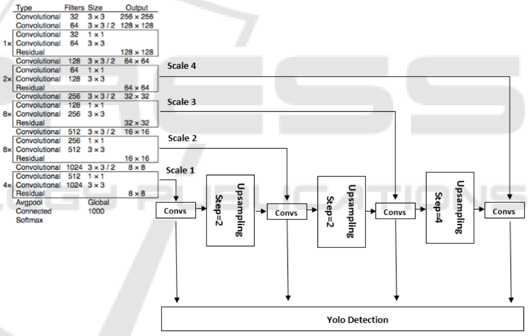
YOLO V3 is the third generation of You Only Look Once (YOLO). YOLO was originally proposed by Joseph Redmon of Washington University where the algorithm uses the Google LeNet model designed by Google to realize end-to-end object detection. The core idea of YOLO is to divide the input image into grid cells of the same size. If the center point of the object's ground truth falls within a certain grid, the grid is responsible for detecting the target. Note that each grid generates **K** anchor boxes of different scales and outputs **B** prediction bounding boxes, including position information of the bounding box (center point coordinates x, y, width w, height h), and prediction confidence. To alleviate the defect of the previous generation of YOLO, YOLO V3 integrates residual network and adds batch normalization (BN) layer and Leaky ReLU layer after each convolution layer. At the same time, YOLO V3 adopts a multi-scale prediction method similar to FPN (Lin et al., 2016) (Feature Pyramid Networks) network to have a better detection effect for large, medium and small targets. As presented in figure 1, it uses three scales of prediction (13 x 13, 26 x 26 and 52 x 52) in order to output different sizes of feature maps. On the other hand, YOLO V3 borrows the idea of using dimension clusters as anchor boxes (Ren et al., 2015) for predicting bounding boxes of the system. It uses nine cluster centers (three for each scale), which can better cover the characteristics of the ground truth of the train set. Each bounding box is represented by a quintuple: x, y, w, h, and a confidence score. Confidence scores represent the precision of the predicted bounding box when the grid contains an object. Note that the confidence score is equal to 1 if the bounding box prior overlaps a ground truth object by more than any other bounding box prior. Else, it will ignore the prediction when the confidence score of detection is lower than the threshold. YOLO V3 uses Darknet53 network as the backbone (Figure 1) which originally composed of 53-layer network trained on ImageNet (Deng et al., 2009). For the task of detection, 53 more layers are stacked onto it. This is the reason behind the slowness of YOLO v3 compared to YOLO v2. Regarding the performance, it has been proved to be more effective than Darknet-19, 1.5 times more efficient than ResNet-101 and 2 times much better than ResNet-152 (Redmon and Farhadi, 2018).



	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1×	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2×	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8×	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8×	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4×	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8

Figure 1: YOLO V3 network structure.

Furthermore, the structure of YOLO V3 is similar to ResNet (He et al., 2015); both use the residual to make the network deeper. To preserve the target characteristics and facilitate the calculation of loss function, YOLO V3 adds a large number of convolution layers of 1x1 and 3x3. YOLO V3 makes detection at three different scales (13x13, 26x26 and 52x52) where the up-sampled layers concatenated with the previous layers preserve the fine-grained features, which help in detecting small objects. The first detection is made by few convolutional layers, which detect the high-resolution and low-level features. For the second detection layer, layers are up-sampled by a factor of 2 and concatenated with the features maps from the earlier network having identical feature map sizes. Another detection is now made at the third detection layer with stride 16 where the same up-sampling procedure is repeated between 2* up-sampled features from the second detection layer and the much earlier network to detect the low-resolution and high-level features. After conducting experiments, the detection performance of the original YOLO V3 is poor for small and occluded objects. That is because the feature maps used for prediction in YOLO v3 network only have three scales and lack of fine-grained information of small objects like plant specimens with small leaves. As shown in figure 2, we added a new scale of feature map to the existing scales. As a result, four scales of feature maps are adopted to detect small and occluded objects. Besides, we changed the fourth detection layer which is rich in detail localization information by a 4* up-sampled layer instead of 2* to get a feature map. As a result, the improved YOLO V3 can detect all objects within the DHS with accurate and stable bounding boxes.



	Type	Filters	Size	Output
	Convolutional	32	3 × 3	256 × 256
	Convolutional	64	3 × 3 / 2	128 × 128
1×	Convolutional	32	1 × 1	
	Convolutional	64	3 × 3	
	Residual			128 × 128
	Convolutional	128	3 × 3 / 2	64 × 64
2×	Convolutional	64	1 × 1	
	Convolutional	128	3 × 3	
	Residual			64 × 64
	Convolutional	256	3 × 3 / 2	32 × 32
8×	Convolutional	128	1 × 1	
	Convolutional	256	3 × 3	
	Residual			32 × 32
	Convolutional	512	3 × 3 / 2	16 × 16
8×	Convolutional	256	1 × 1	
	Convolutional	512	3 × 3	
	Residual			16 × 16
	Convolutional	1024	3 × 3 / 2	8 × 8
4×	Convolutional	512	1 × 1	
	Convolutional	1024	3 × 3	
	Residual			8 × 8
	Avgpool		Global	
	Connected		1000	
	Softmax			

Figure 2: Improved YOLO V3 network structure.

3 EXPERIMENTAL SETUP

During the experiment step, we have used the free cloud service provided by Google, which is based on Jupyter Notebooks that support free GPUs. Google Colaboratory is a free cloud service for machine learning education and research with 12 GB of RAM and NVIDIA Tesla T4. Furthermore, the performance evaluation of the involved system is tested on data provided by the herbarium Haussknecht of FSU Jena.

3.1 Database

We train our CNN model on data provided by the herbarium Haussknecht in Germany, which gave access to more than 30k scanned specimen images to researchers and the public. The collected data contain objects with a high degree of variability in scale and occlusion, making it one of the most challenging data sets. Among them, we annotated manually 4000 images having specimen images with distorted leaves (leaves with missed part) or overlapping leaves, not only specimens containing perfect leaves and some samples are shown in Figure 5. Herbarium specimen images contain seven main regions of scale-bar, barcode, stamp, annotation label, color pallet, envelope, and the plant specimen. Every region is represented by a bounding box described by x, y, width and height within the XML file (Figure 3). We emphasize that the bounding box is dedicated by annotating all objects within the digitized specimen images except the plant specimen region. Otherwise, because of its irregular shape, we describe the plant specimen region by a bounding polygon.

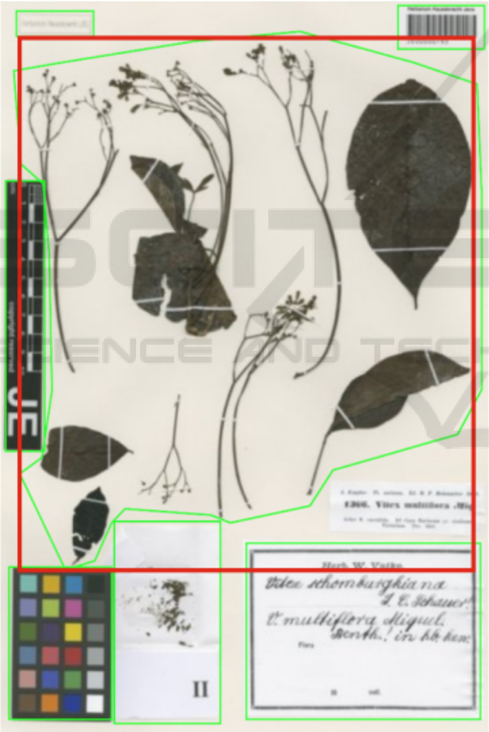
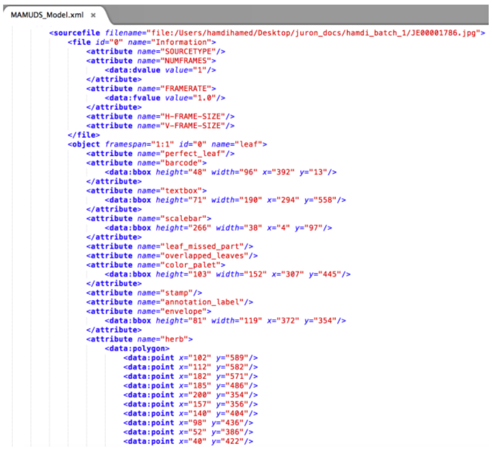


Figure 3: Annotation process of a digitized specimen image.

Consequently, the associated element within the XML annotation file is described by a set of coordinates x, y attributes (Figure 4). To avoid over-fitting problems, we used data augmentation techniques by applying some simple transformations such as horizontal and vertical flipping, rotation and color space to increase the number of samples in our network (Zhang et al., 2016).



```
<sourcefile filename="file:/Users/hamdihamed/Desktop/juron_docs/hamdi_batch_1/JE00001786.jpg">
    <file id="0" name="Information">
        <attribute name="SOURCETYPE"/>
        <attribute name="NUMFRAMES">
            <data:dvalue value="1"/>
        </attribute>
        <attribute name="FRAMERATE">
            <data:fvalue value="1.0"/>
        </attribute>
        <attribute name="H-FRAME-SIZE"/>
        <attribute name="V-FRAME-SIZE"/>
    </file>
    <object framespan="1:1" id="0" name="leaf">
        <attribute name="perfect_leaf"/>
        <attribute name="barcode">
            <data:bbox height="48" width="96" x="392" y="13"/>
        </attribute>
        <attribute name="textbox">
            <data:bbox height="71" width="190" x="294" y="558"/>
        </attribute>
        <attribute name="scalebar">
            <data:bbox height="266" width="38" x="4" y="97"/>
        </attribute>
        <attribute name="leaf_missed_part"/>
        <attribute name="overlapped_leaves"/>
        <attribute name="color_palet">
            <data:bbox height="103" width="152" x="307" y="445"/>
        </attribute>
        <attribute name="stamp"/>
        <attribute name="annotation_label"/>
        <attribute name="envelope">
            <data:bbox height="81" width="119" x="372" y="354"/>
        </attribute>
        <attribute name="herb">
            <data:polygon>
                <data:point x="102" y="589"/>
                <data:point x="112" y="582"/>
                <data:point x="182" y="571"/>
                <data:point x="185" y="486"/>
                <data:point x="200" y="354"/>
                <data:point x="157" y="356"/>
                <data:point x="140" y="404"/>
                <data:point x="98" y="436"/>
                <data:point x="52" y="386"/>
                <data:point x="40" y="422"/>
```

Figure 4: Example of the XML annotation file.

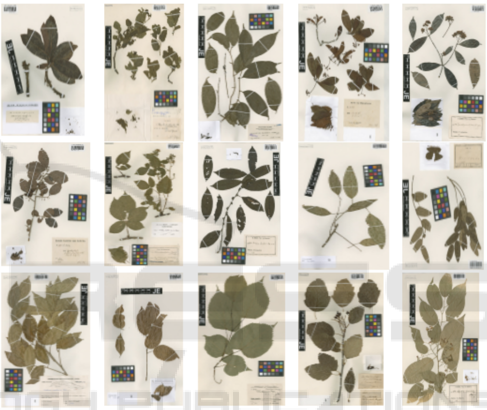


Figure 5: Examples of herbarium Haussknecht dataset.

The dataset was divided into 80% training set, 10% validation set and 10% test set. We trained the original YOLO V3 and improved YOLO V3 models on DHS database. In both networks, the parameters are set as follows: the initial learning rate is reduced to 0.0001 and batch size is 6. Furthermore, all networks were trained for 10000 iterations and we got the avg loss curve as presented in Figure 7.

4 RESULTS AND EVALUATION

We test our object detection model based on improved YOLO V3 on the HHJ database. We selected 400 testing samples as input to the network. Currently, the evaluating metrics commonly used in object detection include mean Intersection Over Union (IoU), recall, precision, mean Average Precision (mAP) and so on. As shown in Table 1, when thresh is set to 0.25, it is verified that the precision accuracy of our proposed approach is increased by more than 3% com-

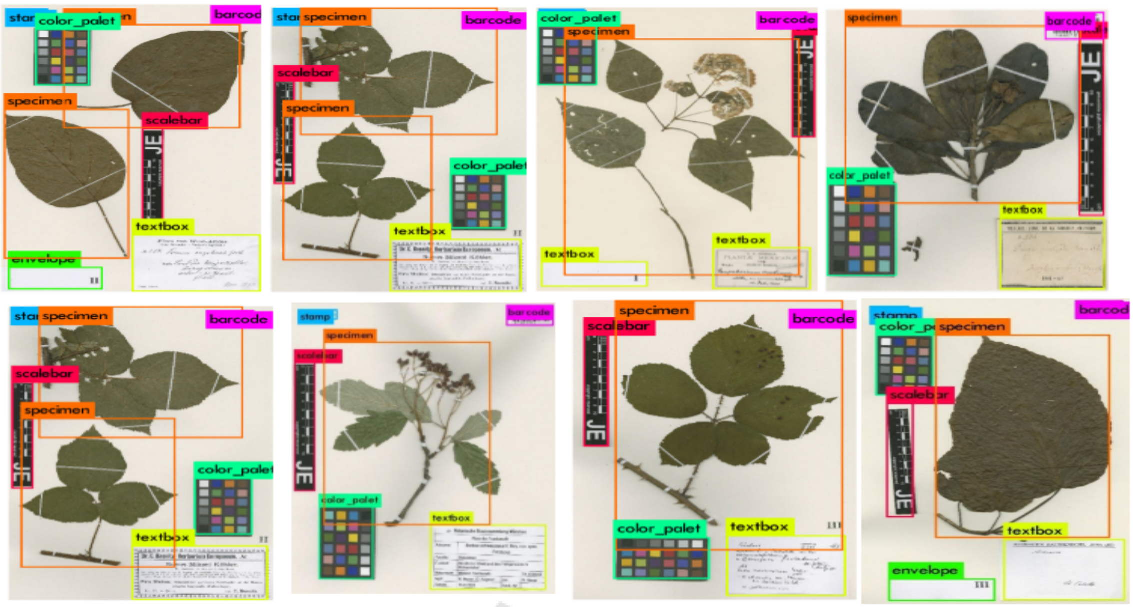


Figure 6: Object detection results by improved YOLO V3 on HHJ dataset.

pared with the original YOLO V3. Whereas the recall reaches 94%, improving 5%. It can be seen that our object detection model has good adaptability and robustness to objects with a high degree of variability in scale and occlusion. Regarding the calculation time, improved YOLO V3 takes more time compared to the original YOLO V3. This is due mainly to the bigger scale of the feature map, which increases the calculation time.

Table 1: Precision / Recall Accuracy.

Index	Original YOLO V3	Improved YOLO V3
Precision	88%	91%
Recall	89%	94%

Numerous metrics have been used by different approaches for evaluating the object detection models (Lateef and Ruichek, 2019). One of the most important evaluation metrics is the Intersection over Union (IoU). This metric quantifies the similarity between the ground truth and the predicted bounding boxes for each class and provides a mean IoU which is calculated by taking the IoU of each class and averaging them. As shown in Equation 1, MIoU calculates the ratio of true positives over the sum of true positive, false positive and false negative.

$$MIoU = \frac{1}{C}\sum_{x=1}^{C}\frac{TP_{xx}}{\sum_{y=1}^{C} FP_{xy} + \sum_{y=1}^{C} FN_{yx} - TP_{xx}} \tag{1}$$

Where C is the total number of classes, TPxx represents the true positives samples, FPxy represents the false positive samples and FNyx represents the false negative samples.

As shown in Table 2, both the original and improved YOLO V3 models performed well on validation and test sets where the proposed model can detect small objects with a MIoU of 94% for stamp object while the performance is slightly better with both validation and test MIoU of 96% for specimen object.

To verify the efficiency of our model, the loss curves of improved YOLO V3 is shown in figure 7. The loss curve shows a gradual decrease in the loss as the training progress. This behavior is observed until iteration 9000 where the loss no longer decreases, which indicates that the training is sufficient and the network can be tested. Furthermore, our proposed approach progressively improved and eventually produced better results.

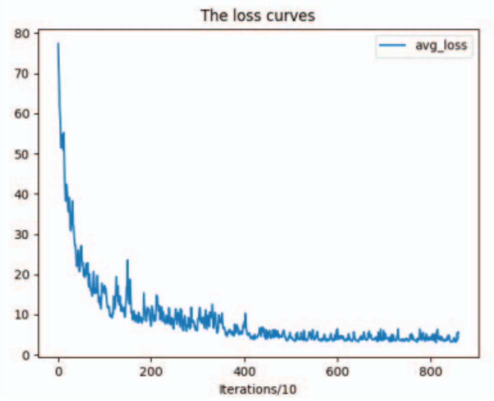


Figure 7: Loss curve.

Table 2: Mean Intersection over Union Measurements.

Objects	Original YOLO V3	Improved YOLO V3
Stamp	88%	94%
bar-code	86%	93%
Scale Bar	88%	91%
Color Pallet	92%	93%
Specimen Label	91%	94%
Envelope	92%	91%
Specimen	94%	96%
Time(s)	0.1	0.4

From the experimental results, the improved YOLO V3 performances were evaluated using mAP-50 metric which is an extension of average precision where we take the average of all AP's to get the mAP and the model score threshold is set at 50%. The mAP-50 of the proposed method is increased by 2.1% compared to the original YOLO V3 (Table 3). This shows that the proposed method achieves good accuracy without a significant speed-up drop.

Besides, the detection results in Figure 6 demonstrate that our method is effective to detect multiple small objects within the digitized herbarium specimens such as a stamp, bar-code and plant specimens with small leaves.

Table 3: mAP.

	Original YOLO V3	Improved YOLO V3
mAP	90.1%	93.2%

5 CONCLUSION AND FUTURE WORK

In this paper, we proposed an improved YOLO V3 based method for detecting objects of different sizes and locations from DHS images. In the prediction process, we added a new scale of feature map to the existing scales to detect smaller objects. At the same time, we adopted the fourth detection layer by a 4* up-sampled layer instead of 2* to get a feature map with a higher resolution level. In terms of efficiency, the proposed algorithm improves the detection accuracy compared to the original YOLO V3. In future work, we intend to extend this work by utilizing our proposed method in building an identification system for herbarium collected specimens by considering the detected plant bounding box as input to a leaves species classifier and serve as a base for leaves species measurements.

ACKNOWLEDGEMENTS

This work is part of the MAMUDS project (Management Multimedia Data for Science) which is funded by the German ministry of education and research (BMBF Project No. 01D16009) and Tunisian ministry of higher education and research including herbarium Haussknecht, University of Sfax and Friedrich-Schiller University, Jena. Authors would like to thank Prof.dr. Birgitta König-Ries, Prof.Dr.Frank Hellwig, and Dr. Jörn Hentschel for supporting this project.

REFERENCES

Belongie, S., Malik, J., and Puzicha, J. (2002). Shape matching and object recognition using shape contexts. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(4):509–522.

Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., and Fei-Fei, L. (2009). ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*.

Felzenszwalb, P., Mcallester, D., and Ramanan, D. (2008). A discriminatively trained, multiscale, deformable part model. *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*.

Felzenszwalb, P. F., Girshick, R. B., and McAllester, D. (2010). Cascade object detection with deformable part models. In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2241–2248.

Felzenszwalb, P. F., Girshick, R. B., McAllester, D., and Ramanan, D. (2010). Object detection with discriminatively trained part-based models. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(9):1627–1645.

Girshick, R. (2015). Fast r-cnn. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ICCV '15, pages 1440–1448, Washington, DC, USA. IEEE Computer Society.

Girshick, R., Donahue, J., Darrell, T., and Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '14, pages 580–587, Washington, DC, USA. IEEE Computer Society.

He, K., Zhang, X., Ren, S., and Sun, J. (2014). Spatial pyramid pooling in deep convolutional networks for visual recognition. *CoRR*, abs/1406.4729.

He, K., Zhang, X., Ren, S., and Sun, J. (2015). Deep residual learning for image recognition. *CoRR*, abs/1512.03385.

Heisele, B. (2003). Visual object recognition with supervised learning. *IEEE Intelligent Systems*, 18(3):38–42.

Hussin, R., Juhari, M. R., Kang, N. W., Ismail, R., and Kamarudin, A. (2012). Digital image processing techniques for object detection from complex background image. *Procedia Engineering*, 41:340 – 344. International Symposium on Robotics and Intelligent Sensors 2012 (IRIS 2012).

Lateef, F. and Ruichek, Y. (2019). Survey on semantic segmentation using deep learning techniques. *Neurocomputing*, 338:321 – 348.

Lin, T., Dollár, P., Girshick, R. B., He, K., Hariharan, B., and Belongie, S. J. (2016). Feature pyramid networks for object detection. *CoRR*, abs/1612.03144.

Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S. E., Fu, C., and Berg, A. C. (2015). SSD: single shot multibox detector. *CoRR*, abs/1512.02325.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110.

Lowe, D. G., Lowe, D. G., and Lowe, D. G. (1999). Object recognition from local scale-invariant features. In *Proceedings of the International Conference on Computer Vision-Volume 2 - Volume 2*, ICCV '99, pages 1150–, Washington, DC, USA. IEEE Computer Society.

Redmon, J. and Farhadi, A. (2018). Yolov3: An incremental improvement. *CoRR*, abs/1804.02767.

Ren, S., He, K., Girshick, R., and Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'15, pages 91–99, Cambridge, MA, USA. MIT Press.

van de Sande, K. E. A., Uijlings, J. R. R., Gevers, T., and Smeulders, A. W. M. (2011). Segmentation as selective search for object recognition. In *2011 International Conference on Computer Vision*, pages 1879–1886.

Viola, P. and Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. pages 511–518.

Viola, P. and Jones, M. J. (2004). Robust real-time face detection. *Int. J. Comput. Vision*, 57(2):137–154.

Vondrick, C., Khosla, A., Pirsiavash, H., Malisiewicz, T., and Torralba, A. (2015). Visualizing object detection features. *CoRR*, abs/1502.05461.

Zhang, C., Bengio, S., Hardt, M., Recht, B., and Vinyals, O. (2016). Understanding deep learning requires rethinking generalization. *CoRR*, abs/1611.03530.