Yolov4-tiny and Spatial Pyramid Pooling for Detecting Head and Tail of Fish

Eko Prasetyo
Department of Informatics, Faculty of Intelligent Electrical and Informatics Technology
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
Department of Informatics, Engineering Faculty
Universitas Bhayangkara Surabaya
Surabaya, Indonesia
eko@ubhara.ac.id

Nanik Suciati*
Department of Informatics, Faculty of Intelligent Electrical and Informatics Technology
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
*corresponding author :
nanik@if.its.ac.id

Chastine Fatichah
Department of Informatics, Faculty of Intelligent Electrical and Informatics Technology
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
chastine@if.its.ac.id

***Abstract*— Touchless fish freshness monitoring is an appropriate approach to avoid the destruction of the fish due to fingering or bacterial contamination from the hands of consumers. A freshness monitoring system that classifies the fish's freshness based on body parts requires an object detection model, such as You Only Look Once (YOLO), for detecting the head and tail of fish. Yolov4-tiny is the recent tiny version of YOLO that has the advantage of a straightforward and fast in detecting objects. However, Yolov4-tiny obtain lower performance in object detection since the lack of diverse features generated by the backbone from the consecutive convolution layer. This paper proposes modifying Yolov4-tiny by inserting Spatial Pyramid Pooling (SPP) to expand the variety of feature maps using various kernel pooling from the same convolution layer. Our experimental results show that SPP increases the Recall of the model in detecting the expected object up to 83.42% and Recall up to 68.51% compared to the original versions.**

Keywords*— *yolov4-tiny, head, tail, object detection, spatial pyramid pooling

I. INTRODUCTION

Fish freshness is generally examined based on body parts such as head, gills, [1]–[4], or skins [5]. A touchless fish's freshness monitoring is an appropriate approach to reduce the destruction of the fish's body due to fingering or bacterial contamination from the hands of consumers. Of course, this procedure cannot be used on the gills because the gill covers should be revealed. In this approach, fish is photographed using a cellphone camera; then, the freshness of the fish is determined by looking at specific body features. The head, body, and tail are the parts that are visible in the shot. Hence, the detection and localization of this part become essential to be studied and addressed. Research conducted by [6] studied with Yolov3 and Mask-RCNN to discover which models provide optimal detection performance. The results show that using IOU 0.5, Yolov3 outperformed Mask-RCNN.

The detection stage of the body parts, particularly the head and tail, is required to classify the freshness of the body parts. The success of object detection is critical since it serves as an input for the classification of freshness. Furthermore, the fish freshness application is expected to be applied to mobile devices with limited resource capacity. We require a head and tail detection model with high performance and light in implementation; besides, the system is also portable and non-destructive [7]. As an object detection method based on Convolutional Neural Network (CNN), You Only Look Once (YOLO) has the advantage of a straightforward and fast model in detecting objects. Yolo version 3 (Yolov3) and version 4 (Yolov4) achieve an accuracy of 77.2% [8] and 78.7% [9], [10], respectively, on COCO dataset [11]. Yolo's standard version provides excellent performance using a large model with a complex layer design. Modifications in this version solve various detection problems, such as uneaten fish food [12], apple flower [13], mask wearing [14], [15], and non-helmeted motorcyclist [16]. On the other hand, the tiny version of Yolo version 3 (Yolov3-tiny) reduces the size of the backbone model from Darknet53 to 9 layers. The tiny version of Yolo version 4 (Yolov4-tiny) also cuts the CSPDarknet53 model's backbone to three CSPDarknet layers and four convolution layers [10]. The tiny version of YOLO gives small model sizes with lower performance. Hence, object detection with the Yolo family becomes interesting because it has high-speed detection [17], and straightforward architecture.

As described before, head and tail detection required a small-size model for mobile device implementation; however, Yolov3-tiny and Yolov4-tiny performed poorly at recognizing heads and tails. The lack of diverse features generated by the backbone from the consecutive convolution layer is the cause of this issue. These features map may furthermore be improved with various features by pooling features using several kernel sizes [17]. This study proposes modifying Yolov4-tiny by inserting Spatial Pyramid Pooling (SPP) to expand various feature maps of various kernel pooling from the same straight convolution layer. SPP is inserted between the end of the backbone layer and the detection block, and it does a pooling feature map utilizing multiple kernel pooling sizes to achieve many variations of the features map in the backbone layer. The features produced by SPP strengthen the features map to detect objects of different sizes where the tail size is smaller than the head. We experimented by comparing the original Yolov3-tiny and Yolov4-tiny with our proposal (Yolov4-tiny-SPP) to prove the effectiveness of our proposal in detecting fish heads and tails. Our experimental results show that SPP increases the Recall of the model that detects the expected object and decreases the Precision of the model since more false-positive objects are obtained. In addition, the F1-score results reveal that the Yolov4-tiny-SPP is competitive with the original version.

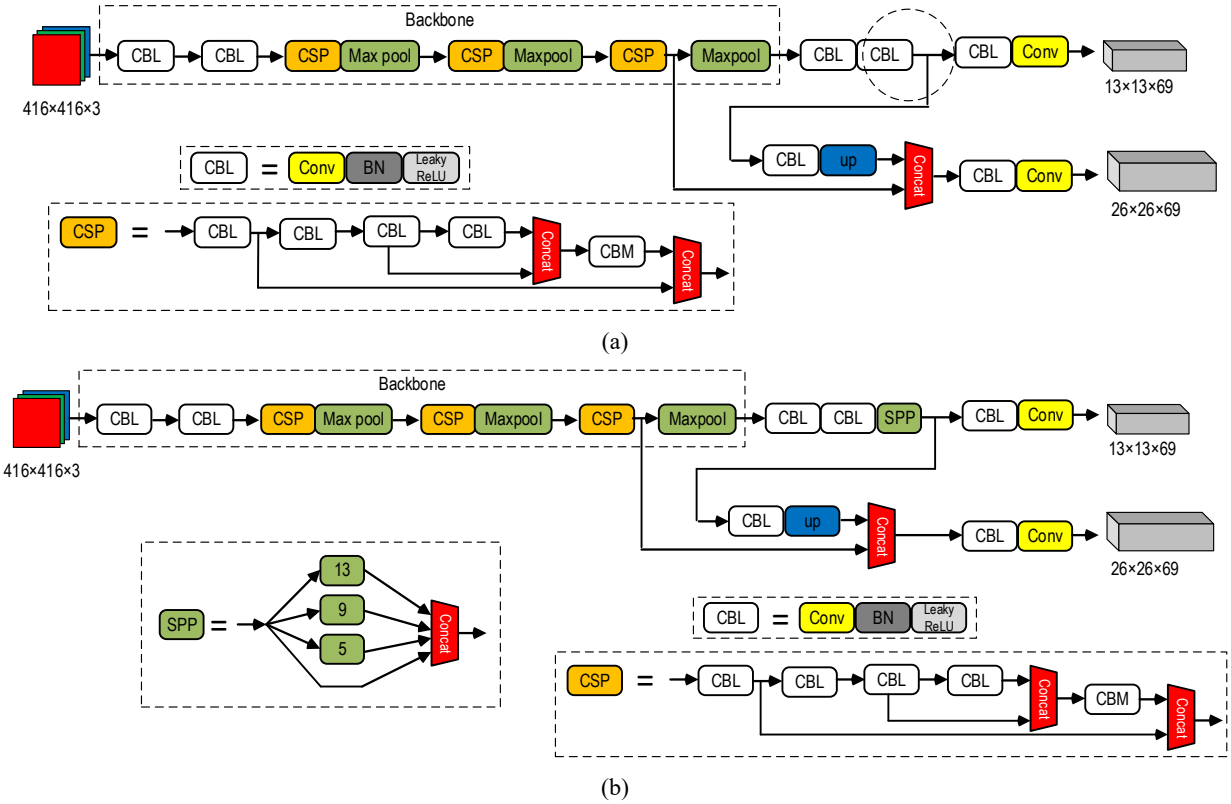

Fig. 1 Yolov4-tiny; (a) Original; (b) Plus SPP

II. Research Methodology

A. Proposed Method

The principal module of the Yolov4-tiny backbone is Cross-stage Partial (CSP), consisting of CBL, residual, and cross-stage features with skip connection several CBL (Convolution, Batch Normalization, Leaky Relu) layers. This module is used three times with other CBLs, as shown in Fig. 1(a). Yolov4-tiny uses two detectors at the end of the architecture. We insert the SPP between the final backbone and the detector to increase the variety of multi-scale features on the same convolution layer. A dashed circle indicates the location. SPP increases the computing volume (measured in BFLOPs) and the model size (measured by the file size model). We would present our experiments on Yolo versions 3 and 4 and our proposal in the next section.

The model's detection section consists of one-time CBL and convolution followed by a detector layer. The backbone's features map is upsampled and combined with residue from the same features map resolution for detection at the second scale. The first scale is 13 × 13 for detecting large objects, while the second is 26 × 26 for detection of smaller objects, as shown in Fig. 1(a). Because it uses a features map that flows through, the features map variations are slightly sensitive at specific scales.

To strengthen the model with various multi-scale feature maps from the same convolution layer, we propose Spatial Pyramid Pooling (SPP) to increase the model's ability to achieve multiple feature map variations in detecting various object sizes all scales. Feature map variations are achieved by pooling several different kernels at the same layer and then concatenate them. We modify Yolov4-tiny by inserting the SPP between the backbone and detection blocks, as shown in Fig. 1(b).

We use SPP by selecting the kernel size based on the following formula.

$$size_{pool} = \lceil size_{fmap}/n_i \rceil \quad (1)$$

Where i = 1, 2, 3 so that the sliding window pooling haves size $\lceil size_{fmap}/1 \rceil \times \lceil size_{fmap}/1 \rceil$, $\lceil size_{fmap}/2 \rceil \times \lceil size_{fmap}/2 \rceil$, and $\lceil size_{fmap}/3 \rceil \times \lceil size_{fmap}/3 \rceil$, where max-pooling is conducted using stride = 1 and certain padding so that the pooled features map is the same size as the input. All pooling results are concatenated into a single features map. The feature map generated by the backbone now has a variety of multi-scale features, allowing the detector to perform better.

B. Dataset

We used a fish head and tail detection dataset in research [6], consisting of 200 images divided into 160 training images and 40 testing images [18]. The dataset has two object classes, namely head and tail, containing 723 heads and 585 tails; the head is divided into 591 and 132 annotations for training and testing, while the tail is divided into 482 and 103 annotations for training and testing. The annotations use the YOLO format [x, y, w, h], where x and y represent the bounding box's central points, while w and h represent the height and width of the bounding box.

C. Evaluation Metric

We evaluate the performance of all models using Precision, Recall, and F1-score. Precision is the proportion of objects detected correctly from all detected objects, while Recall is the proportion of objects detected from existing objects. The formula for calculating Precision and Recall is shown in the equation below.

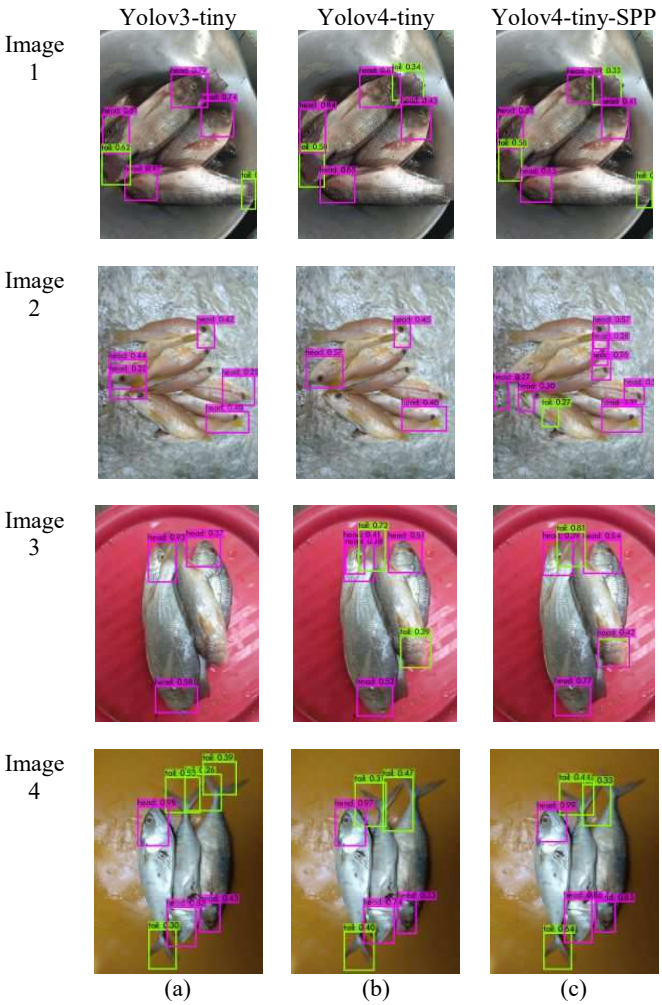

Fig. 2 Sample detection result [18]

$$P = \frac{TP}{TP+FP} \quad (2)$$

$$R = \frac{TP}{TP+FN} \quad (3)$$

Where TP is a proper object that was detected (true positive), FP is a false object that was detected (false positive), and FN is an actual object that failed to be detected (false negative). The F1-score is calculated from the harmonic mean of the Precision and Recall with balanced weight. We use the formula as follows.

$$F_{1-score} = \frac{2 \times P \times R}{P+R} \quad (4)$$

Precision provides performance in terms of the correctness of the detection results, while Recall provides performance in terms of the model's capability to detect the expected object, and the F1-score balances the two into one performance.

Especially for object detection, the metric mean of Average Precision (mAP) is also utilized to evaluate the detection performance. Average Precision (AP) is calculated from the average of interpolated Precision to Recall for each detected object. The interpolated Precision is calculated by the formula as follows.

$$P_{interp}(R) = \max_{\tilde{R}>R} P(\tilde{R}) \quad (5)$$

Furthermore, the AP is obtained by calculating the average of interpolated Precision on Recall for each class with the formula as follows.

$$AP = \frac{1}{N} \sum_{R \in \{0, 0.1, \ldots, 0.9, 1\}} P_{interp}(R) \quad (6)$$

mAP is calculated from the average AP in all classes as follows.

$$mAP = \frac{1}{C} \sum_{i=1}^{C} AP_i \quad (7)$$

We also use BFLOPS (Binary Floating-Point Operations) to find out the model computation volume and model size to evaluate the storage capacity required by the model.

III. Results and Discussions

A. Experiment scenario

We experiment using Colab with Tensorflow 2.3.0 and Keras 2.4.3, the hyperparameters for Yolo with two class detection as follows, max batch 4000, steps 3200 and 3600, batch size 12, subdivision 4. We utilize GPU from Colab with one GPU Tesla T4 16 GB for training 160 images during training. After that, we use 40 images as testing images. The results are presented in the next sections.

B. Detection Results

The examples of detection results on 40 testing images are presented in Fig. 2. In Image 1, Yolov4-tiny-SPP was better at detecting seven objects, while Yolov3-tiny and Yolov4-tiny detect six objects each. In this image, one tail was not detected by the original version, while our proposal was successful in detecting both tails. In Image 2, Yolov3-tiny detects five heads (4 TP and 1 FP), Yolov4-tiny detects three heads (TP), while Yolov4-tiny-SPP excels by detecting seven heads (6 TP and 1 FP) and one head (TP). Moreover, in Image 3 and 4, Yolov4-tiny-SPP outperformed the original version, where more objects were detected, but some objects should not be detected (FP).

C. Performance Analysis

The principal purpose of developing a model for object detection is to obtain the objects in the image. Therefore, we compare our proposed model with the original version of both Yolo versions 3 and 4 in terms of detection results, as presented in Table 1. In our experiment, Precision and Recall are utilized to measure model performance. Precision measure the performance of the model that the object it detects is correct. Recall measuring the performance of a robust system in detecting the expected object. The results achieved by the model as in Table 1 show that Recall Yolov4-tiny-SPP outperforms all other models, where the head class reaches 80.30% compared to the original version 78.79%, the tail class reaches 53.40% compared to the original version 51.46%, while all classes reach 68.51 % compared to the original version 66.81%. In Recall training, Yolov4-tiny-SPP outperformed the original version. On Precision, Yolov4-tiny-SPP also outperformed other models on training performance, but validation performance decreased, as many objects instead of heads or tails were detected as heads or tails. On the one hand, this way can improve the detection performance of objects that should be detected, but some non-required objects are also detected.

We combined Precision and Recall into an F1-score to present the overall performance as the harmonic mean between the two. The performance shown by the F1-score in the training session shows that Yolov4-tiny-SPP achieves the best performance where for the head, tail, and all classes are 91.84%, 84.44%, and 78.71%, respectively. The validation session is different, where the best F1-score on head detection is achieved by the Yolov4-tiny original, while Yolo4-tiny-SPP achieves the tail. Meanwhile, the performance in all classes

shows that the original Yolov4-tiny is slightly better than Yolov4-tiny-SPP, namely 75.85% and 75.23%, respectively. This shows that our proposal competes with the original version.

The data presented in Table 2 shows that our proposed model uses more computation volume and storage space because additional SPP results in additional layers and weights. The computation volume required by our proposal Yolov4-tiny-SPP is 8.03 BFLOPS; this is higher than the original version; the storage size required by our proposal is 36 MB while the original version is smaller, which is 33 MB. However, with the addition of SPP, our proposal was successful in increasing the mAP using IUO 0.5. Our proposal increase both mAP training and validation from 91% to 93.55% for training and 76.14% to 76.54% for validation. Compared to standard Yolov3 and Mask-RCNN as in research [6], where Yolov3 was superior to our proposal, Yolov3 achieved 80.12% mAP, while Mask-RCNN was not superior to our proposal, where Mask-RCNN achieved 73.39% mAP, but the size of our proposed model is highly more diminutive than the Yolov3 and Mask-RCNN.

Therefore, the results of experiments and analyses using the head and tail of fish dataset finding that Yolov4-tiny-SPP improves model performance in detecting the expected objects, namely heads and tails, with the performance of Precision 83.42% and Recall 68.51%.

Notwithstanding our proposed model having a higher computation volume and model size than the original version, 8.03 BFLOPs computation and 36 MB storage size are yet inside the small limits of the standard version. This increase is accompanied by improved performance in both Precision and Recall.

TABLE I. MODEL PERFORMANCE

Object	Sess.	Met.	Yolov3-tiny	Yolov4-tiny	Yolov4-tiny-SPP
Head	Train	Prec.	88.16	91.99	**92.31**
		Rec.	79.36	87.48	**91.37**
		F1-score	83.53	89.68	**91.84**
	Val.	Prec.	85.19	**91.23**	86.89
		Rec.	69.70	78.79	**80.30**
		F1-score	76.67	**84.55**	83.47
Tail	Train	Prec.	85.64	89.85	**91.73**
		Rec.	66.80	73.44	**78.22**
		F1-score	75.06	80.82	**84.44**
	Val.	Prec.	75.47	**81.54**	77.46
		Rec.	38.83	51.46	**53.40**
		F1-score	51.28	63.10	**63.22**
All	Train	Prec.	87.11	91.11	**92.07**
		Rec.	59.30	65.29	**68.74**
		F1-score	70.56	76.07	**78.71**
	Val.	Prec.	81.99	**87.71**	83.42
		Rec.	56.17	66.81	**68.51**
		F1-score	66.67	**75.85**	75.23

TABLE II. MODEL COMPARISON

Model	BFLOPS	Size (MB)	mAP	
			Train	Val.
YOLOv3 [6]	-	243	-	**80.12**
Mask-RCNN [6]	-	244	-	73.39
YOLOv3-tiny	**5.45**	33	85.49	69.5
YOLOv4-tiny	6.79	**22**	91	76.14
Yolov4-tiny-SPP	8.03	36	**93.55**	76.54

IV. CONCLUSIONS

Our experimental results by inserting Spatial Pyramid Pooling (SPP) on Yolov4-tiny are proven in improving the model's performance in detecting heads and tails. SPP is inserted between the backbone and the detector to increase the variety of pooled feature maps using multiple kernel pooling from the same layer. The insertion of SPP increases detection performance as well as computation volume and storage capacity. In addition, there are still many non-head and non-tail objects detected by the proposed model. As future work, research is needed to improve detection performance while maintaining low computation and storage volume.

ACKNOWLEDGMENT

This research is funded by the Deputy in Strengthening Research and Development, Ministry of Research and Technology / National Research and Innovation Agency, Indonesia with contract number 829/PKS/ITS/2021, on March 10, 2021.

REFERENCES

[1] M. K. Dutta, A. Issac, N. Minhas, and B. Sarkar, “Image processing based method to assess fish quality and freshness,” *Journal of Food Engineering*, 2016.

[2] A. Issac, M. K. Dutta, and B. Sarkar, “Computer vision based method for quality and freshness check for fish from segmented gills,” *Computers and Electronics in Agriculture*, 2017.

[3] L. K. S. Tolentino *et al.*, “Fish Freshness Determination through Support Vector Machine,” *Journal of Telecommunication, Electronic andComputer Engineering*, vol. 9, no. 2–5, pp. 139–143, 2017.

[4] H. Mohammadi Lalabadi, M. Sadeghi, and S. A. Mireei, “Fish freshness categorization from eyes and gills color features using multi-class artificial neural network and support vector machines,” *Aquacultural Engineering*, vol. 90, p. 102076, Aug. 2020.

[5] N. Sengar, V. Gupta, M. K. Dutta, and C. M. Travieso, “Image Processing Based Method For Identification Of Fish Freshness Using Skin Tissue,” in *International Conference on "Computational Intelligence and Communication Technology", CICT 2018*, 2018.

[6] E. Prasetyo, N. Suciati, and C. Fatichah, “A Comparison of YOLO and Mask R-CNN for Segmenting Head and Tail of Fish,” in *International Conference on Informatics and Computational Sciences (ICICoS)*, 2020.

[7] S. Kunjulakshmi *et al.*, “Development of portable, non-destructive freshness indicative sensor for Indian Mackerel (Rastrelliger kanagurta) stored under ice,” *Journal of Food Engineering*, vol. 287, p. 110132, Dec. 2020.

[8] J. Redmon and A. Farhadi, “Yolov3,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2017.

[9] A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, “YOLOv4: Optimal Speed and Accuracy of Object Detection,” Apr. 2020.

[10] A. Bochkovskiy, “GitHub - AlexeyAB/darknet: YOLOv4 / Scaled-YOLOv4 / YOLO - Neural Networks for Object Detection (Windows and Linux version of Darknet),” 2020. [Online]. Available: https://github.com/AlexeyAB/darknet. [Accessed: 24-May-2021].

[11] T. Y. Lin *et al.*, “Microsoft COCO: Common objects in context,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2014, vol. 8693 LNCS, no. PART 5, pp. 740–755.

[12] X. Hu *et al.*, “Real-time detection of uneaten feed pellets in underwater images for aquaculture using an improved YOLO-V4 network,” *Computers and Electronics in Agriculture*, vol. 185, p. 106135, Jun. 2021.

[13] D. Wu, S. Lv, M. Jiang, and H. Song, “Using channel pruning-based

YOLO v4 deep learning algorithm for the real-time and accurate detection of apple flowers in natural environments," *Computers and Electronics in Agriculture*, vol. 178, p. 105742, Nov. 2020.

[14] A. Kumar, A. Kalia, K. Verma, A. Sharma, and M. Kaushal, "Scaling up face masks detection with YOLO on a novel dataset," *Optik*, vol. 239, p. 166744, Aug. 2021.

[15] M. Loey, G. Manogaran, M. H. N. Taha, and N. E. M. Khalifa, "Fighting against COVID-19: A novel deep learning model based on YOLO-v2 with ResNet-50 for medical face mask detection," *Sustainable Cities and Society*, vol. 65, p. 102600, Feb. 2021.

[16] Y. Jamtsho, P. Riyamongkol, and R. Waranusast, "Real-time license plate detection for non-helmeted motorcyclist using YOLO," *ICT Express*, Aug. 2020.

[17] Z. Huang, J. Wang, X. Fu, T. Yu, Y. Guo, and R. Wang, "DC-SPP-YOLO: Dense connection and spatial pyramid pooling based YOLO for object detection," *Information Sciences*, 2020.

[18] E. Prasetyo, N. Suciati, and C. Fatichah, "Dataset for Detecting Head and Tail of Fish," *Mendeley Data*. Mendeley, 2021.