DOI: https://doi.org/10.33103/uot.ijccce.23.3.7

A New Perspective for Mining COCO Dataset

Suha Dh. Athab[1], Abdulamir Abdullah Karim [2], Kesra Nermend [3]
[1,2]Computer Sciences Department, University of Technology, Baghdad, Iraq
[3]Computer Science Department University of Szczecin. Faculty of Economics and Administration, Boland
[1]cs.20.09@grad.uotechnology.edu.iq, [2]110004@uotechnology.edu.iq, [3]Kesra@wneiz.pi

Abstract—*Microsoft Common Objects in Context (COCO) is a huge image dataset that has over 300 k images belonging to more than ninety-one classes. COCO has valuable information in the field of detection, segmentation, classification, and tagging; but the COCO dataset suffers from being unorganized, and classes in COCO interfere with each other. Dealing with it gives very low and unsatisfying results whether when calculating accuracy or intersection over the union in classification and segmentation algorithms. A simple method is proposed to create a customized subset from the COCO dataset by determining the class or class numbers. The suggested method is very useful as preprocessing step for any detection or segmentation algorithms such as YOLO, SSPNET, RCNN, etc. The proposed method was validated using the link net architecture for semantic segmentation. The results after applying the preprocessing were presented and compared to the state of art methods. The comparison demonstrates the exceptional effectiveness of transfer learning with our preprocessing model.*

Index Terms— *COCO JSON files, Object detection, object tracking, semantic segmentation.*

I. INTRODUCTION

Microsoft COCO (Microsoft Common Objects in Context) is a comprehensive image dataset of ordinary scenarios, including typical things in their natural settings. The dataset includes images of 91 items. COCO has 328,000 pictures of people and common objects. COCO data set was adopted by many researchers within the machine learning field [1] [2]. A study by Sharma [3] investigates the information measure Shannon's to classify images that use deep learning and machine learning methods, the COCO dataset deployed for training the system. COCO is also used within Object detection [4-6], Wang [7] uses a large batch optimization framework for object detection called LargeDet, that successfully scales the batch size to 1056 with a ResNet50 backbone.

Besides captioning by natural language descriptions for each image, more than 200 000 key points for people photos including the right eye, nose, and left hip[8], moreover; segmentations by creating pixels maps of the 91 different types of amorphous, and panoptic involves full scene segmentation` [9] [10]. COCO is a huge image dataset that provides valuable information in the field of detection [11] [12], segmentation, classification, and tagging; Metadata in form of JSON file format is provided with the COCO dataset that has information about image classification as shown in the *Fig. (1-a),* object detection clarified by *Fig. (1-b),* semantic segmentation metadata shown in *Fig. (1-c)* and *(1-d)*; but it suffers from inorganizing [13] [14]. All classes in COCO have interfered with each other as shown in the *Fig. 2*. It's difficult to use COCO within a specific domain unless creating a customized subset from it. In this research suggests a simple method for mining COCO and build a customized dataset according to each researcher's interest. The paper is organized as follows: section II presents the suggested method for mining the COCO dataset. Section III has the results after applying the segmentation algorithm for the customized-created dataset. Section IV has the discussion part.

Received 13/December/2022; Accepted 01/February/2023
© 2017 University of Technology, Iraq ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.23.3.7

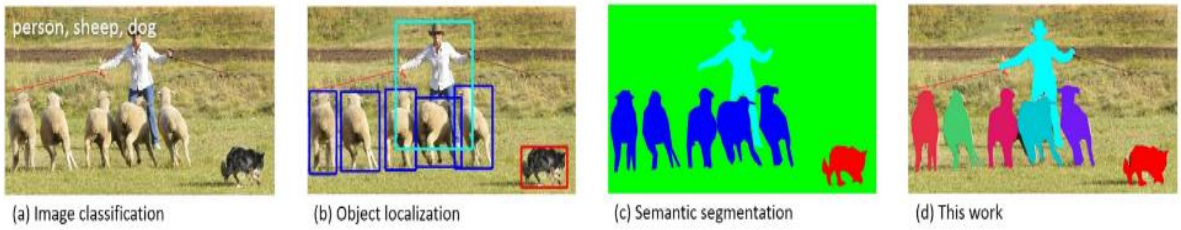

FIG. 1. GRAPHICAL REPRESENTATION FOR METADATA INFORMATION COMBINED WITH EACH IMAGE: (1-A) IMAGE CLASSIFICATION METADATA, (1-B) OBJECT DETECTION METADATA (1-C, D) SEMANTIC SEGMENTATION METADATA.

FIG. 2. SAMPLES FROM COCO SHOW THE INTERFERED CLASSES.

II. METHOD

The COCO dataset is combined with metadata as a nested dictionary in JSON format to supply information[15] about the dataset [16] [17]. The nested dictionary is composed of five sub-dictionaries that contain details about the dataset: such as info, licenses, images, annotations, and categories sub-dictionaries[18]. The info sub-dictionary has a general description of the dataset such as the year of uploading the image, contributor, and URL. The licenses sub-dictionary supplies details to allow using the dataset[19]. The licenses include the URL, id, and name of the dataset. Both the info and licenses section do not contribute to mining the COCO dataset or creating customized ones. The category sub-dictionary has the following keys: 'super category', 'id', and 'name', the sub-dictionary is indexed with the 'id' key. A tabular representation of the sub dictionaries is shown in *Fig. 3.*

Received 13/December/2022; Accepted 01/February/2023
© 2017 University of Technology, Iraq
ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.23.3.7

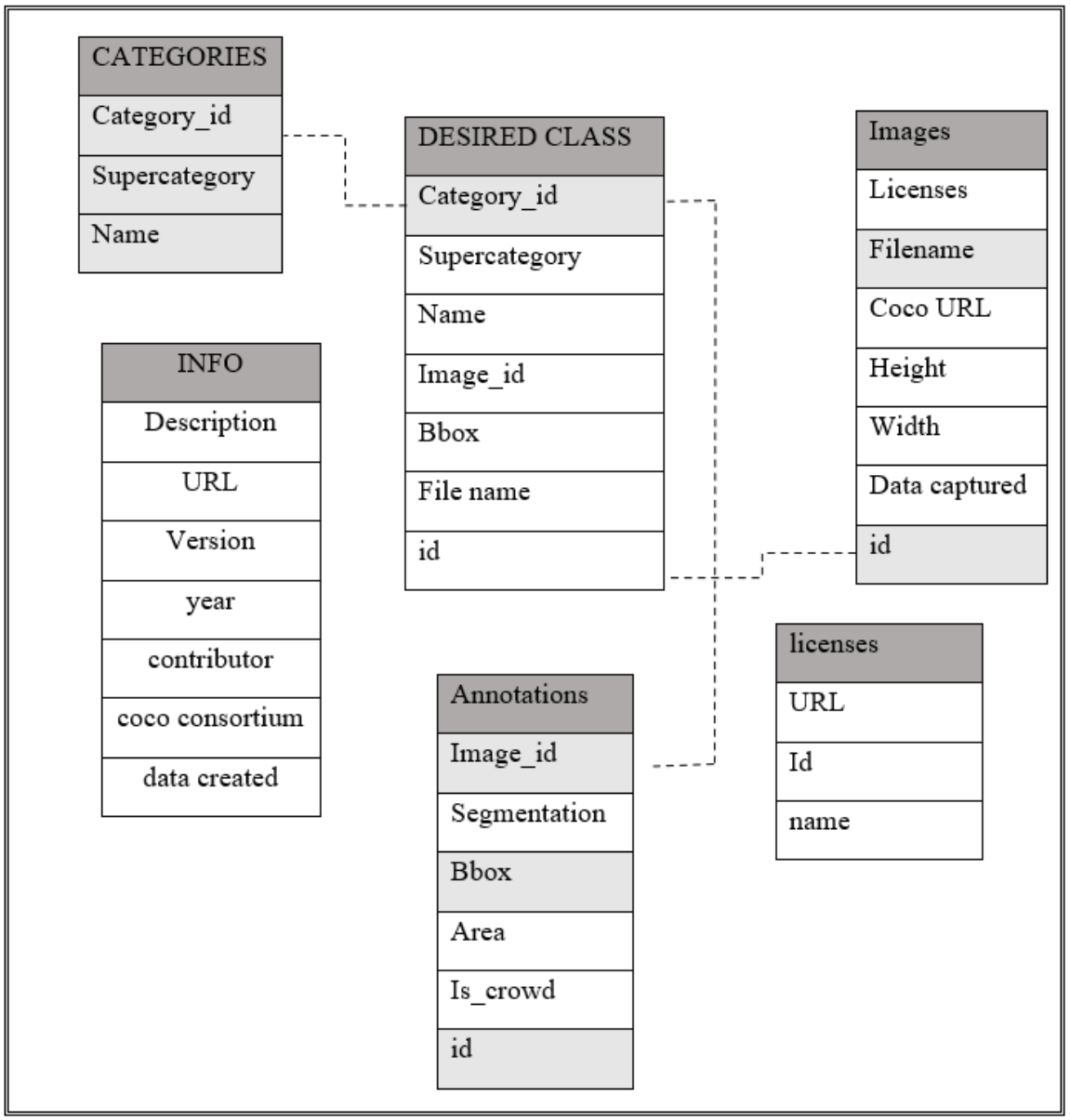

FIG. 3. TABULAR REPRESENTATION FOR SUB DICTIONARIES COMBINED WITH COCOA.

The first step to constructing a customized subset from COCO is creating a 'desired classes' list from the category sub-dictionary with over 91 different classes. For example, if the desired classes were: person, bicycle, or car that have the id numbers 1, 2, and 3 the desired list will have the numbers 1, 2, 3 or any combination for the desired classes numbers. Then create a new dictionary called 'desired class's that holds only the three desired classes as shown below:

{supercategory': 'person', 'id': 1, 'name': 'person'},

{'supercategory': 'vehicle', 'id': 2, 'name': 'bicycle'},

{'supercategory': 'vehicle', 'id': 3, 'name': 'car'}

A list of all object annotations is provided in the annotations sub-dictionary. If several objects of the same type are included in the image, it will be shown by the crowded field in

Received 13/December/2022; Accepted 01/February/2023
© 2017 University of Technology, Iraq ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.23.3.7

this sub-dictionary. The bounding box coordinates are provided via the ‘bbboxfield. The ‘category id’ parameter shows the category under which this object was placed. Here is an example of an annotation sub dictionary

{"segmentation":[[291.11,375.9,291.66,373.18,...]],"area":505.7744000000001,"iscrowd":0
,"image_id":500663,"bbox": [288.39,353.81,38.18,24.0],"category_id": 21,"id": 72296}

Double Check the category id with the desired list and ‘id’ key in ‘desiredClass’ dictionary. If it was identical in both then create a new dictionary call it ‘annotation’ dictionary indexed by ‘Image_id’ this dictionary will have all the content of ‘desiredClass’ dictionary in addition to the bounding box, a few instances of the created ‘annotation’ dictionary are shown below:

{'image_id':95999,'category_id':1,'supercategory':'person','name':'person','Bounding':[227,2
60, 397,82]}

{'image_id':518850,'category_id':1,'supercategory':'person','name':'person','Bounding':[37,1
77, 232, 69]},

{'image_id':133680,'category_id':3,'supercategory':'vehicle','name':'car','Bounding':[56,109,
173.18, 122.66]},

{'image_id':2139,'category_id':2,'supercategory':'vehicle','name':'bicycle','Bounding':[80,4.7
9, 463.38 , 419.15]},

The images sub-dictionary provides raw image information such as width, height, and file name. For each image in the collection, in this step check if the key index of the created annotation dictionary is equal to the id provided with the images sub dictionary if so, then create a new dictionary indexed by the image file name and add all the content of annotation dictionary, the created dictionary will contain the image file name and the category in one dictionary as shown in the following instances:

‘000000025758.jpg': {'image_id': 25758, 'category_id': 1,'supercategory': 'person', 'name': ‘person’, 'Bounding': [114.34, 5.43, 205.48, 231.37]},

'000000015725.jpg': {'image_id': 15725,'category_id': 1,'supercategory': 'person','name': ‘person’, 'Bounding': [127.28, 42.07, 377.53, 430.38]},

'000000188411.jpg': {'image_id': 188411,'category_id': 3,' supercategory': 'vehicle','name': 'car', ‘Bounding’: [298.65, 345.56, 13.46, 5.12]},

'000000503569.jpg': {'image_id': 503569,'category_id': 2,'supercategory': 'vehicle','name': ‘bicycle’, 'Bounding': [445.04, 209.69, 22.35, 20.19]},

This dictionary enables the process of creating a customized dataset with the desired classes instead of the dataset with interference classes. Examples of the customized-created dataset are shown in *Fig. 4 (a, b, c, d)* which presents a giraffe, bus, broccoli and Laptop customized-created datasets from COCO. A step-by-step mining process for the COCO dataset is described below by Algorithm 1.

Received 13/December/2022; Accepted 01/February/2023
© 2017 University of Technology, Iraq ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.23.3.7

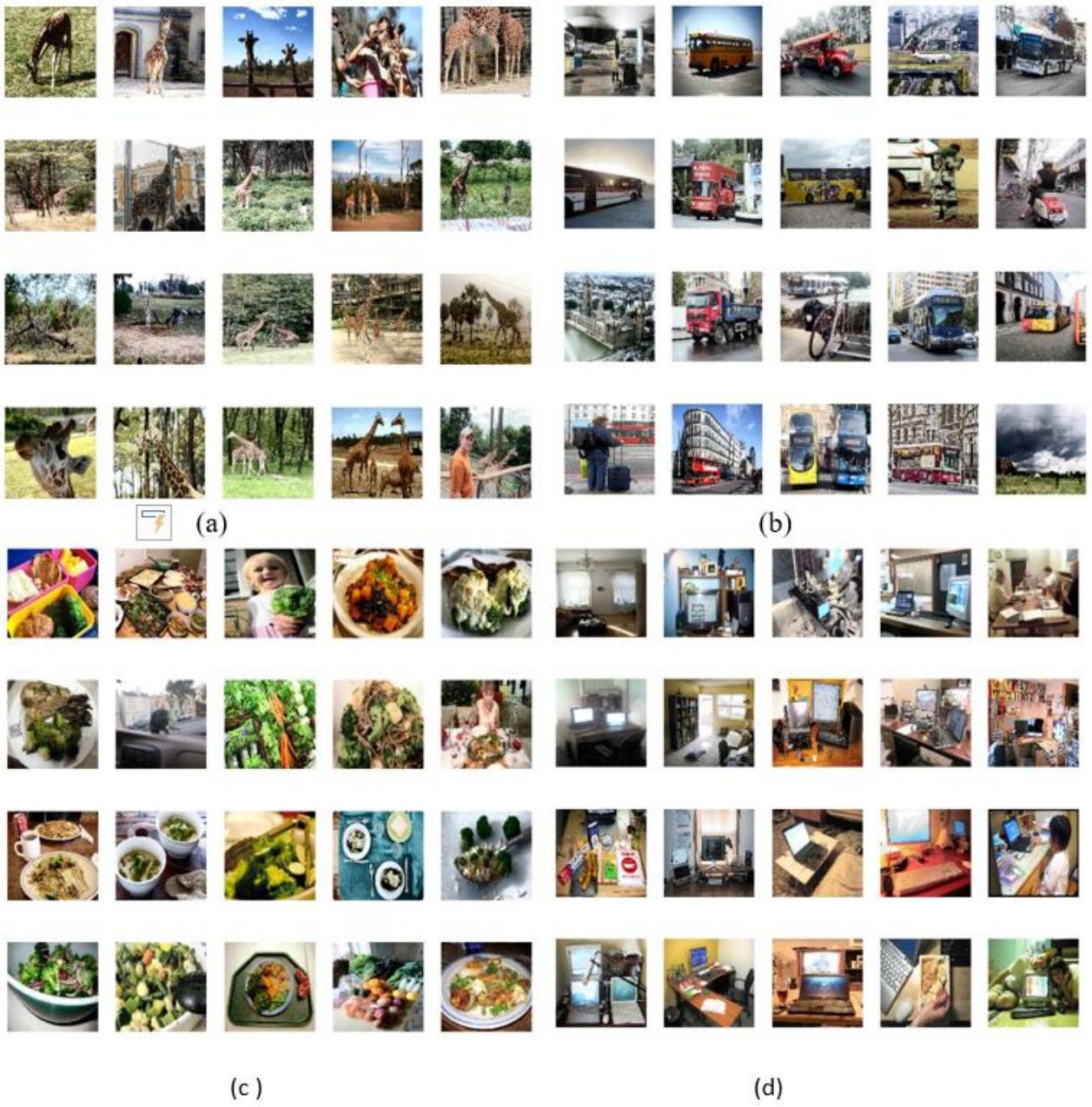

FIG. 4. CUSTOMIZED CREATED DATASETS FROM COCO: (A) PRESENTS A GIRAFFE DATASET, (B) BUS DATASET, (C) BROCCOLI DATASET, (D) LAPTOP DATASET.

Algorithm (1): Mining COCO dataset algorithm
Input: 90 class datasets **Output:** customized dataset with prespecified classes

Begin
Step1: set Coco ← json file // Read json file (Nested Dictionary)
Step2: set Coco[categories] ← *Coco* // Read Coco categories sub dictionary
Step3: set desired_list ← *list*= [1,2, 3,….] // Create list with desired class id
Step4:set category ← dict() // initiate empty dictionary to put the desired extension of the desired classes

Received 13/December/2022; Accepted 01/February/2023
© 2017 University of Technology, Iraq
ISSN (Print) 1811-9212 *ISSN (Online) 2617-3352*

DOI: https://doi.org/10.33103/uot.ijccce.23.3.7

Step5: For all X Do, where {X key of Coco[categories]} // get the key of categories Sub dictionary
Set cat_id ← *X*[id]
Set cat_super ← X[supercategory]
Set cat_name ← X[name]
If cat_id in desired_list then
category[cat_id]={id:cat_id,supercategory:cat_super,name:cat_name}//{add the required required keys, values to the created dictionary}
End if
End for
Step6: set Coco[annotation]← Coco// Read Coco annotation sub dictionary
Step7: set annot ← dict () // initiate empty dictionary
Step8: For all J Do, Where {J key of Coco[annotation]}
Set img_id ← J[image_id]
Set cat_id ← J[category_id]
If cat_id in desired_list then
For all I Do, Where I is counter from 0 to the length of category dictionary
Set cat_ID ← category[I][id]
If cat_id equal to cat_ID then
Set cat_super=[I][supercategory]
Set cat_name=[I][name] annot[img_id]{img_id: imgID,id:cat_id,supercategory:cat_super,name:cat_nam}
End if
End for
End if
End for
Step9: set Coco[images]← Coco // Read Coco images sub dictionary
Step10: set coco images← dict () // initiate empty dictionary
Step11: For all Z Do, Where Z is the key for sub dictionary images in nested dictionary Coco
Set id← Z[id]
Set name← Z[filename]
For all X Do, Where X is the key of an not dictionary
Set img_id← annot[X][img _id]
If id==img_id Then
Set cat_id← annot[X][category_id]
Set cat_super← annot[X]['supercategory']
Set cat_name ← annot[X][name]
coco_images[name]={'image_id':img_Id, 'category_id' :cat_id, 'supercategory': cat _super ,'name':cat_name}
End if
End For
End For
End

A customized subset created from the previous steps was used in the encoder-decoder architecture LinkNet for instance segmentation. The preparation of the data set for the training stage is as follows:

Received 13/December/2022; Accepted 01/February/2023
© 2017 University of Technology, Iraq ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.23.3.7

1. Read the images and resize them to a fixed size as the image in the dataset has a different size. all the images were resized to (128 × 128)

2. Read ground truth masks associated with each image and resize it to (128 × 128)

3. Get the number of classes for each image by counting the number of unique values in ground truth images

4. ground truth images are labelled with numbers representing unique color values for each label starting from 0 to 255 Ground truth images should be converted to a categorical array with the following dimensions (*m,h,w,n)* where *m* is the number of images, *h,w* is the height and width of ground truth image, and *n* is the number of classes since the creation of such array requires huge storage size hence the ground truth images values should be started from 0,1,2,3, etc. using the following procedure:

Algorithm 2 Convert Ground truth images to arrays starting from 0,1,2...

Input: Ground Truth images
Output: Ground Truth images start from 0

```
Begin
Step1: set c ← (1 × n) // where n is the number of classes in dataset
Step 2: train_cat ← [ ] // initiate an empty list
Step 3: for all X Do // where X is ground truth images
            Read X
             For all I Do // where I is the ground truth image height
                  For all J Do // where j is the width of ground truth images
                           For all K Do // where k is counter for c array
                                If x[I][j] equal to c[0][k] then
                                   Set X[I][j] ← k
                                End if
                           End for
                      End for
                  End for
             Add X to train_cat list
            End for
End
```

5. The dataset is divided into a training set (80%) and a validation set (20%). The ResNet50 architecture is used as a base to train our data. The plan is to fine-tune the higher layers of the pre-trained model to our domain while freezing the lower layers of the model that can capture general information. Additionally, the last layer is redefined to provide 8 values one for each class. Utilized the SGD optimizer with weight decay Eq.1[20]

$$w = w - \mu.\nabla E(w; x(i: i+n); y(i: i+n)) \qquad (1)$$

Where $\nabla \mathbf{E}(\mathbf{w})$ is the error gradient concerning weight *w* and $\boldsymbol{\mu}$ is the learning rate that defines the step size to take along the gradient. The learning rate value that gets the model to converge to the global minimum was 0.0001. The fine-tuned model run SGD for 30k with 32 batch iterations

Received 13/December/2022; Accepted 01/February/2023
© 2017 University of Technology, Iraq *ISSN (Print) 1811-9212* *ISSN (Online) 2617-3352*

DOI: https://doi.org/10.33103/uot.ijccce.23.3.7

III. EXPERIMENTAL RESULTS

The proposed method for mining the COCO dataset was tested as a preprocessing step to enhance the performance of instance segmentation algorithms. The validation for Intersection over Union (IoU) Eq.2 [21] was used as the first metric to calculate the overlapped area. The result of IoU is shown in *Fig. 5*. The loss was considered as the second metric Eq.3 [22]

$$IOU = \frac{1}{N}\sum_{i=1}^{N}\frac{y_i \hat{y_i}}{y_i + \hat{y_i} - y_i \hat{y_i}} \quad (2)$$

$$H(y) = -\sum_i \grave{y_i} Log(y_i) \quad (3)$$

where y_i is a binary value (label) of the corresponding pixel i and $\hat{y_i}$ is predicted probability for the pixel. IoU is used to get the value of the predicted boundary overlaps with the ground truth (the real object boundary). *Fig. 6* shows the results of the proposed method using loss metric. A side-by-side comparison of our proposed method with the most recent methods. The results are compared to the state of art methods and show significant enhancement as shown in Table I. Puri et al. [18] use a convolutional neural network and offers a better foundation for pixel-level tasks, the best experimental results of segmented COCO was 50% over IoU metric. There is considerably enhanced segmentation performance where the IoU value exceeds 80% for the proposed method as for other methods like MaskFormer [21] which provides a novel method for semantic segmentation based on mask classification rather than the conventional per-pixel classification the IoU was 37.1%. The comparison demonstrates the exceptional effectiveness of transfer learning with our preprocessing model.

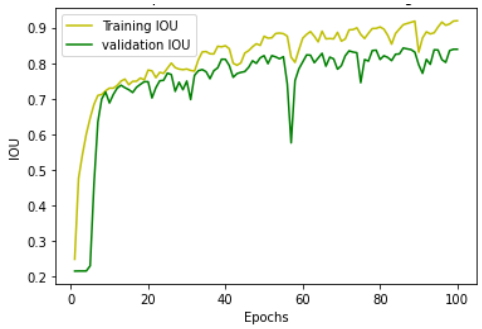

FIG. 5. THE RESULT OF INTERSECTION OVER UNION FOR SEGMENTATION USING CUSTOMIZED-CREATED SUBSET FROM COCO.

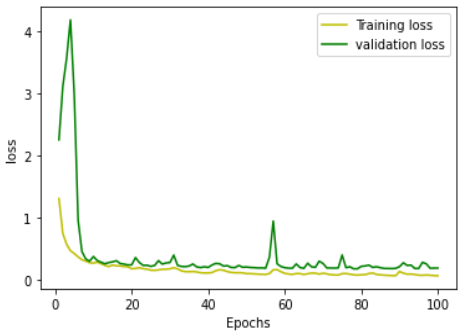

FIG. 6. THE RESULT OF SEGMENTATION USING CUSTOMIZED CREATED SUBSET FROM COCO USING LOSS METRIC.

Received 13/December/2022; Accepted 01/February/2023
© 2017 University of Technology, Iraq
ISSN (Print) 1811-9212 ISSN (Online) 2617-3352

DOI: https://doi.org/10.33103/uot.ijccce.23.3.7

TABLE I IOU COMPARISON BETWEEN THE PROPOSED METHOD AND COMMON STATE OF ART METHODS

`	IoU
Puri et al. [18]	50.0
MaskFormer [21]	37.1
Proposed method	89.6

IV. CONCLUSIONS

The existence of labelled datasets is crucial for training deep learning algorithms. COCO dataset is a rich dataset with various classes such as cat, dog, horse, cow, elephant, bear, motorcycle car airplane, traffic light, fire hydrant, stop signs, parking meters, trucks, boats, etc. Thousands of indoor and outdoor images but it suffers from interferences and unorganized classes. A simple method was proposed to automatically create a database dedicated to any researcher according to the nature of their work. The method is based on isolating the interfered classes by creating a dictionary containing the image number and the class number corresponding to a predefined category. The model shows strong performance improvement on pertinent tasks with little data and computation.

REFERENCES

[1] N. Ali, M. Abdulmunem, "Constructed model for micro-content recognition in lip reading based deep learning," vol. 10, no. 5, pp. 2557-2565, 2021.

[2] Guo, Jian, and Stephen Gould. "Deep CNN ensemble with data augmentation for object detection." *arXiv preprint arXiv:1506.07224* (2015).

[3] D. Sharma, "Information Measure Computation and its Impact in MI COCO Dataset," *7th International Conference on Advanced Computing and Communication Systems (ICACCS)*, 2021, vol. 1, pp. 1964-1969: IEEE in *2021*.

[4] T.-Y. Lin *et al.*, "Microsoft coco: Common objects in context," in *European conference on computer vision*, pp. 740-755: Springer 2014.

[5] M. M. Mahmoud, C. Nasser, COMMUNICATION, CONTROL, and S. ENGINEERING, "Dual Architecture Deep Learning Based Object Detection System for Autonomous Driving," vol. 21, no. 2, 2021.

[6] T. H. Obaida, N. F. Hassan, C. Jamil, COMMUNICATIONS, CONTROL, and S. ENGINEERING, "Comparative of Viola-Jones and YOLO v3 for Face Detection in Real time," vol. 22, no. 2, pp. 63-72, 2022.

[7] T. Wang *et al.*, "Large Batch Optimization for Object Detection: Training COCO in 12 minutes," in *European Conference on Computer Vision*, 2020, pp. 481-496: Springer.

[8] N. Yadav, U. Binay, and Technology, "Comparative study of object detection algorithms," vol. 4, no. 11, pp. 586-591, 2017.

[9] M. N. Abdullah and S. Ali, "Vehicles Detection System at Different Weather Conditions," pp. 2040-2052, 2021.

[10] S. I. Mohammed, N. A. Jaafar, and E. Hussien, "Face Recognition Based on Viola-Jones Face Detection Method and Principle Component Analysis (PCA)," vol. 18, no. 3, pp. 52-59, 2018.

[11] S. Athab,. and N Selman., "Disc and Cup Segmentation for Glaucoma Detection". *Journal of mechanics of continua and mathematical sciences*, *14*(6), pp.369-383 2019.

[12] Y. Fang *et al.*, "Eva: Exploring the limits of masked visual representation learning at scale," 2022.

[13] G. Patterson and J. Hays, "Coco attributes: Attributes for people, animals, and objects," in *European Conference on Computer Vision*, 2016, pp. 85-100: Springer.

[14] T. T. Yang *et al.*, "AICOM-MP: an AI-based Monkeypox Detector for Resource-Constrained Environments," 2022.

[15] J. Cha, J. Mun, and B. Roh, "Grounded Contrastive Learning for Open-world Semantic Segmentation."

[16] H. Caesar, J. Uijlings, and V. Ferrari, "Coco-stuff: Thing and stuff classes in context," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 1209-1218.

Received 13/December/2022; Accepted 01/February/2023
© 2017 University of Technology, Iraq *ISSN (Print) 1811-9212* *ISSN (Online) 2617-3352*

DOI: https://doi.org/10.33103/uot.ijccce.23.3.7

[17] Khafaji AP. VKDP: A New Approach in Knowledge Discovery Process. IRAQI JOURNAL OF COMPUTERS, COMMUNICATIONS, CONTROL AND SYSTEMS ENGINEERING. 2011 Jun 28;11(1):1-4.

[18] D. Puri, "COCO dataset stuff segmentation challenge," in *2019 5th international conference on computing, communication, control and automation (ICCUBEA)*, 2019, pp. 1-5: IEEE.

[19] S. Rostianingsih, A. Setiawan, and S. Halim, "COCO (creating common object in context) dataset for chemistry apparatus," vol. 171, pp. 2445-2452, 2020.

[20] Z. Zhang, "Improved adam optimizer for deep neural networks," in *2018 IEEE/ACM 26th international symposium on quality of service (IWQoS)*, 2018, pp. 1-2: Ieee.

[21] Z. Zheng, P. Wang, W. Liu, J. Li, R. Ye, and D. Ren, "Distance-IoU loss: Faster and better learning for bounding box regression," in *Proceedings of the AAAI conference on artificial intelligence*, 2020, vol. 34, no. 07, pp. 12993-13000.

[22] M. A. Rahman and Y. Wang, "Optimizing intersection-over-union in deep neural networks for image segmentation," in *International symposium on visual computing*, 2016, pp. 234-244: Springer.

Received 13/December/2022; Accepted 01/February/2023
© 2017 University of Technology, Iraq *ISSN (Print) 1811-9212* *ISSN (Online) 2617-3352*