XLIII
CILAMCE
21-25, November, 2022
FOZ DO IGUAÇU - PR - BRAZIL

Image-based detection and classification of screws and nuts using deep learning

Gizele P. do Nascimento[1], Karin S. Komati[2], Luiz A. Pinto[1]

[1]*Programa de Pós-Graduação em Engenharia de Controle e Automação (ProPECaut), Instituto Federal do Espírito Santo (IFES) Campus Serra*
gizelepolt@gmail.com, luiz.pt@ifes.edu.br
[2]*Programa de Pós-Graduação em Computação Aplicada (PPComp), Instituto Federal do Espírito Santo (IFES) Campus Serra*
kkomati@ifes.edu.br
Avenida dos Sabiás, 330 - Morada de Laranjeiras, 29166-630, Serra-ES, Brasil

Abstract. Object detection in images has been one of the biggest challenges for computer vision researchers. This paper presents a case study on screws and nuts detection and classification. The identification of screws is not a trivial task. There are about 1,500 unique bolts, and in some cases, the differences between the two pieces involve only tiny details, making identification difficult for untrained people. The experiments used an MVTec Screws dataset with 384 images of bolts and nuts on a wooden background. The classification problem consists of 10 classes that differ in the length and width of the screws or nuts diameter, the color of the metal, and the shape of the screw head, tip, or thread. Objects in some images are separated, in others ones, objects are together or overlapped. For screws and nuts detection and classification, the network YOLOv4 and the Darknet framework were used for training and inference. Performance was evaluated considering detection and classification after 4,200 epochs run. The results in detection, in terms of IoU and mAP, scored 77.79% and 97.79%, respectively. In classification tasks, all classes reached above 99% F1-Score.

Keywords: YOLO, Computer Vision, Neural Networks, MVTec Screws, Object Detection.

1 Introduction

Object detection in images is one of the most important machine vision tasks Ulrich et al. [1]. Because of its wide application, this topic has attracted the attention of industry and academia. In the last few years, the rapid advances in deep learning techniques have greatly accelerated the momentum of object detection. With deep learning networks and the computing power of GPUs, the performance of object detectors has greatly improved, achieving significant advances. The state-of-the-art object detection methods can be divided into two approaches. While two-stage detection techniques offer the advantage of high accuracy in detecting and locating objects, one-stage detectors predict bounding boxes directly over the images. This process consumes less time and can be used in real-time applications.

This paper presents a case study on screws and nuts detection and classification. The identification of screws and nuts is not a trivial task. There are about 1,500 unique types of screws, and in some cases, the differences between the two pieces involve only tiny details, making identification difficult for untrained people. The proposal is to use YOLO (You Only Look Once) network Bochkovskiy et al. [2], based on the one-stage approach, which uses a single structure to generate the bounding boxes and to estimate the class probabilities directly from the images. The classification problem consists of 10 classes from MVTec Screws dataset Ulrich et al. [1].

The remainder of the paper is organized as follows, the next section presents related works. Then, there is a description of the materials and methods, followed by the experiments, results, and discussion. Finally, we present the conclusions and future works.

2 Related work

Detecting and classifying types of objects with specific characteristics and applications has been the effort of many researchers who use computer vision techniques with deep learning. There are many architectures available in the literature for this task. In this section, we've selected articles that accomplish this task with a version of YOLO. One is the system proposed by Mangold et al. [3], which can locate and classify six different types of screw heads of varying sizes using two versions of YOLOv5 to adapt the robot's end-effector. The mean average precision (mAP) was above 0.98 for training data (550 images) and accuracy of 84.6% for the on-site validation. The proposal of Qiu et al. [4] is an improved YOLOv4 model, combining YOLOv4 with MobileNet lightweight convolutional neural network to detect insulator detection. The detection results of transmission line insulator and defect images show that the detection accuracy of the proposed model can reach 93.81%, and the detection accuracy can be further improved to 97.26% after the image preprocessing.

The work of Lee et al. [5] proposes a method using YOLOv3 to find the center value and orientation of an object even when the shape is not uniform such as a bolt. A data set for learning was produced by taking 1,000 images of bolts and nuts placed randomly on the floor. The work uses the Labelimg software for the labeling task. Bolt is divided into three classes: Whole bolt, Bolt head, and Bolt tail. On the other hand, the shape of the nut is circular, so one class is enough for nut detection and its geometric information. All four classes showed an accuracy of 90% or more, and the nut was 100% accurate.

3 Methodology

This section describes the carried out steps to implement the detecting and classifying screws and nuts system. The used dataset is presented, as well as the procedure for the detector's parameters settings.

3.1 Dataset

The MVTec Screws dataset used in the experimental phase was obtained from a repository, provided by Ulrich et al. [1], which can be referenced for academic purposes for free. The dataset consists of 384 images in total, from different kinds of screws and nuts on a structured wooden background. All images in this application are 8-bit three-channel RGB color images of size 960 × 720 pixels. Among the images, there are 13 different object categories. However, in this work, only 10 of the 13 categories have been used for object detection purposes. The categories differ in the length and width of the screw or the diameter of the nut, the color of the metal, and the shape of the screw's head, tip, or thread. Figure 1, in the left, illustrates an example of a typical image from the dataset, showing some objects. As can be seen on the right, in some particular images, objects might be touching or overlapping each other. Such a situation offers an additional challenge for object detection and classification algorithms.

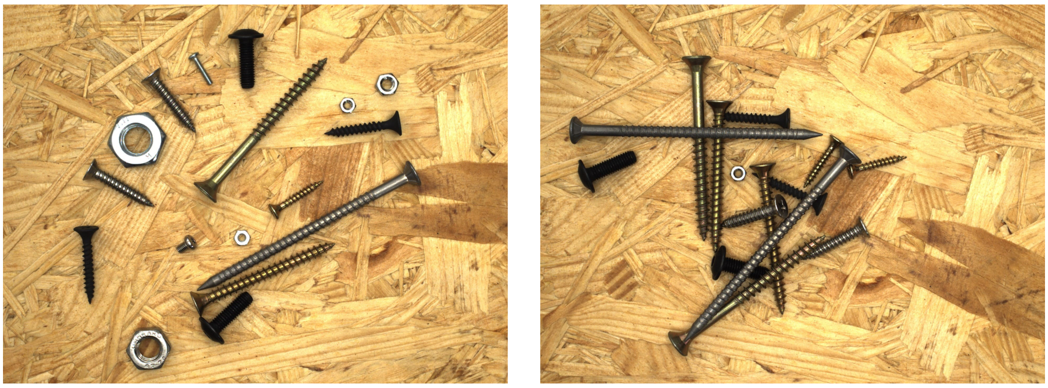

Figure 1. Examples of images from the dataset showing, at left the categories of the different objects, and at right overlapped objects. Source: Ulrich et al. [1].

3.2 Preprocessing and dataset preparation

In this work, between the 13 categories, only 10 were used in the experiments, being, six types of screws and four types of nuts, which can be seen in Fig. 2. Considering an object detection supervised approach, in the training phase of the network it is mandatory that the location of each object of all categories be annotated in the whole dataset images. Object annotations have been carried out using the Labelimg software Tzutalin [6], which is a free open source tool for labeling images graphically. By using the labeling tool it is possible to manually label objects in images and export a file containing the indication of the associated class and the respective coordinates of the label created Tzutalin [6]. As a result of the annotation phase, 3,143 objects were labeled. Table 1 shows the number of objects by category after the annotation phase.

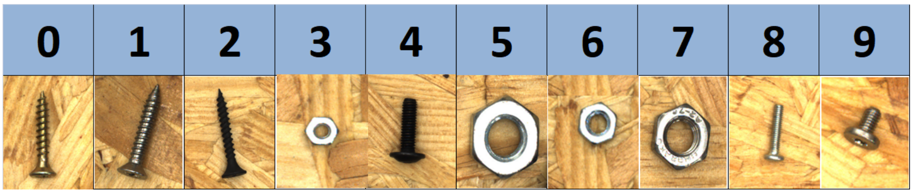

Figure 2. Tags versus imagens selecionadas. Source: adapted from Ulrich et al. [1]

Table 1. Object categories in the dataset.

Class	Object Name	Train	Validation	Number of Objects
0	n0	179	90	269
1	n1	235	100	335
2	n2	235	100	335
3	n3	206	104	310
4	n4	204	105	309
5	n5	223	88	311
6	n6	246	87	333
7	n7	241	90	331
8	n8	216	93	309
9	n9	209	92	301
	Total	3143	2194	949

3.3 YOLO

YOLO network is based on the single-stage detection principle and uses a single neural network to predict bounding boxes and class probabilities directly from images in a Redmon et al. [7] evaluation. The last version of YOLO architecture is YOLOv6, however, in this work, the authors used version 4 (YOLOv4, released in April 2020), since it achieved good performance for the problem at hand.

According to Bochkovskiy et al. [2], if compared to YOLOv3 Redmon and Farhadi [8], the YOLOv4 main characteristics are the improvement in inference speed and accuracy. Also according to Bochkovskiy et al. [2], another important feature is the fact it is more efficient to run on GPUs, because it was optimized to use less memory. In addition, Espíndola et al. [9] points out that the accuracy of YOLOv4 is higher than that of YOLOv3. The YOLOv4 overall network Fig. 3 architecture consists of three parts: (i) backbone: CSPDarknet53, (ii) neck: SPP, PAN, and (iii) head: YOLOv3.

In the screws and nuts detection and classification problem, the subject of this paper, the so-called YOLOv4 network has been applied, by using the Darknet architecture Redmon [10]. The use of YOLOv4 for detection purposes requires the prior configuration of some hyperparameters Espíndola et al. [9]. According to Gao and Zhong [11], because it defines the number of samples to work with before updating the parameters of the internal model, the most important one is the batch size. In this paper, the batch size was set to 64. The subdivision parameter indicates the memory consumption, if the value is too low, the memory consumption will be high and the model may not run within Google Colaboratory. In this work, the value used was 64. Width and height parameters refer to the width and height dimensions of the dataset images, in pixels. The values of such parameters must be divisible by 32, and considering the YOLOv4, by default, is 608x608.

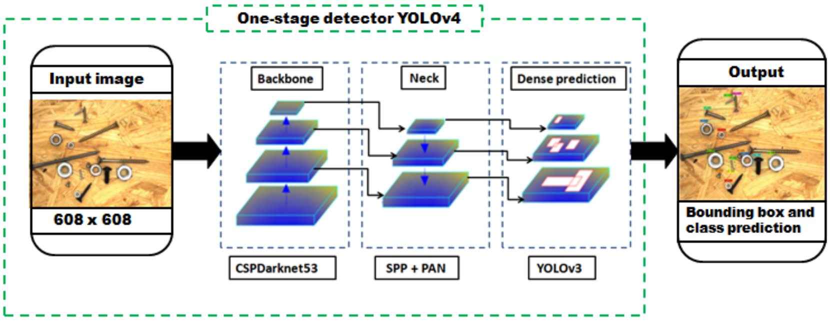

Figure 3. YOLOv4 network architecture. Source: adapted from Bochkovskiy et al. [2]

3.4 Training and validation settings

Deep learning models require a large amount of data to get better performance than other techniques, need GPUs, and take a long time to train. The transfer learning technique can overcome these limitations of deep learning methods by reusing a trained model on a specific task as part of the training process for a different task with a small amount of data [12].

In a general way, the number of epochs to train machine learning models is proportional to the number of categories in the dataset. In the case of YOLOv4, according to Bochkovskiy et al. [13], 2,000 iterations are necessary for each class. Therefore, as in the present problem there exist 10 categories, the ideal number would be 20,000 iterations. In the experiments carried out in this work, tests were executed in with 100, 430, 2,200, 3,700, and 4,200 epochs. In each training step, weights were stored and evaluation metrics calculated to verify the network's performance in the corresponding number of training epochs.

In the model building phase, initially, from the sample total amount in the dataset, 15% (34 samples) were separated to be used in the testing phase. From the remaining 350 samples, 70% were used in the training phase and 30% were applied to validate the model. To constitute the test set, 18 samples obtained from Google Image were added to the 34 initially separated samples of the original data set MVTec. All experiments were run on the Google Collaboratory Python Notebook, using a Google Pro GPU, CUDA 10, cuDNN, and OpenCV.

3.5 Evaluation measurement

The performance of the models was evaluated using the IoU, mAP, and F1-Score metrics Espíndola et al. [9]. Intersection over Union (IoU) is an evaluation metric used to measure the accuracy of an object detector. In practice, IoU is a value that quantifies the degree of overlap of the predicted bounding box coordinates to the ground truth box. Higher IoU indicates the predicted bounding box coordinates closely resemble the ground truth box coordinates.

The mean average precision (mAP) metric is the mean value of Average Precision (AP), which is calculated separately for each category based on the recall and precision values. The mAP compares the ground-truth bounding box to the detected box and returns a score. The higher the score, the more accurate the model is. Usually, IoU and mAP metrics are efficient in object detection tasks, however, they are not suitable in classification tasks. In this way, the F1-Score metric was used to verify the model's performance in classifying the objects.

4 Results and discussions

Figure 4 and Fig. 5 show the performance of the model in detecting the objects in the training phase. In Fig. 4, results are presented in terms of IoU metric. As can be seen, in the initial training steps, from 100 to 420 training epochs the IoU value remains stable at around 30%. Considering IoU measures the overlapping degree among the predicted bounding box and the ground truth box, it means that at this training step the model's performance in detecting objects is very poor. As the number of training epochs increases in the range from 420 to 2,200 epochs, the IoU value increases significantly, reaching a value of 77.71%, which means the screws and nuts detection performance greatly improves. By analyzing the graph in Fig. 4 it can be seen that further increases in the number of training epochs to values above 2,200 result in marginal increases in IoU. The maximum number of training epochs was 4,200, which resulted in an IoU value of 80.71%. However, the best model performance considering IoU (81.53%) was obtained with 3,700 epochs.

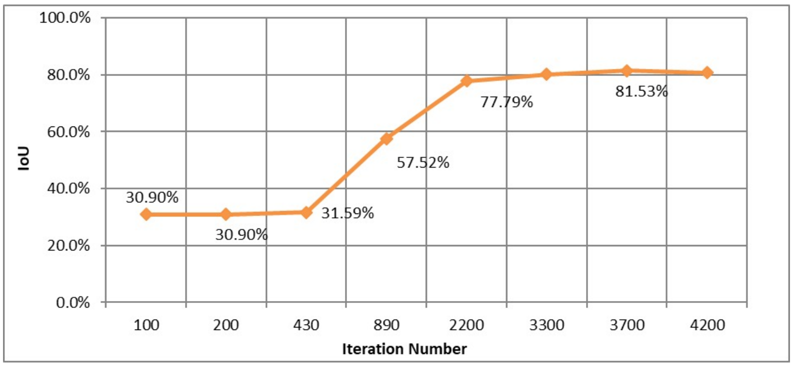

Figure 4. IoU in the training phase.

Figure 5 shows the performance of the model in the training phase, considering the mAP metric. Similar to what happened with IoU, in the initial training phase (from 100 to 200 epochs), the value of mAP was extremely low (about 7.87%). Since mAP is an average of the AP values, which in turn is an average of the APs of all classes, the low value of mAP means that in the initial training phase the model's ability to correctly detect classes is very low. Considering the number of epochs in the range 200 to 890, it can be seen that the greater the number of epochs, the better the accuracy of the model. In 890 epochs the model's average accuracy in correctly detecting objects in each category is 95.71%. In the tests performed in this work, the highest value mAP was obtained when the model was trained with 3,700 epochs (98.90%).

Table 2 specifies the performance of the YOLO network in detecting each class individually. For nine of the 10 classes, the best performance was obtained with 3,700 epochs. According to the AP metric, the worst performance was for class five, where the AP value was 96.81%. The network performance can be considered acceptable at 2,200 epochs. Increasing the epochs number values above up to 2,200, few or no improvement in class detection was noted. Overall, the network shows an excellent performance in correctly detecting objects of all categories.

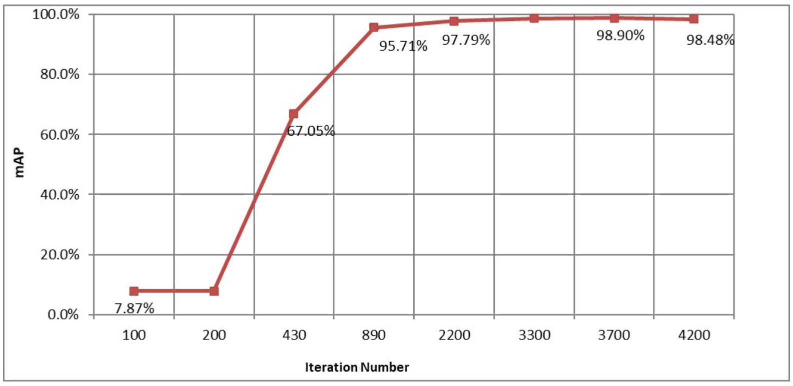

Figure 5. mAP in the training phase.

Since IoU and mAP metrics only evaluate the model's performance in detecting objects, the F1-Score metric was used to evaluate the model's ability to classify the objects (six screw types and four nut types) into a specific category among the ten possibles classes. Figure 5 shows the overall YOLOv4 network performance in classifying the objects according to the network's number of training epochs.

As with IoU and mAP metrics, it can be seen the classification performance in the range between 100 and 200 epochs is very poor. As the number of training epochs increases, in the range between 200 and 2,200 epochs, the F1-Score value increases significantly. At 2,200 epochs, the F1-Score value is 98%. Further increase in the number of epochs over 2,200 will not significantly improve the network's performance in the object classification task.

Table 2. AP values in the testing phase.

Class	100	430	2200	3700	4200
0	1.32	48.89	98.89	100.00	97.75
1	2.08	54.04	100.00	100.00	100.00
2	1.09	78.86	100.00	100.00	100.00
3	0.86	61.61	92.20	99.04	100.00
4	7.07	78.79	99.04	99.05	99.04
5	38.19	57.05	97.37	96.81	96.62
6	6.36	46.74	98.91	99.47	98.41
7	16.20	71.46	97.72	98.86	98.85
8	3.42	86.98	96.36	.97.71	97.04
9	2.16	86.09	97.44	98.10	97.05

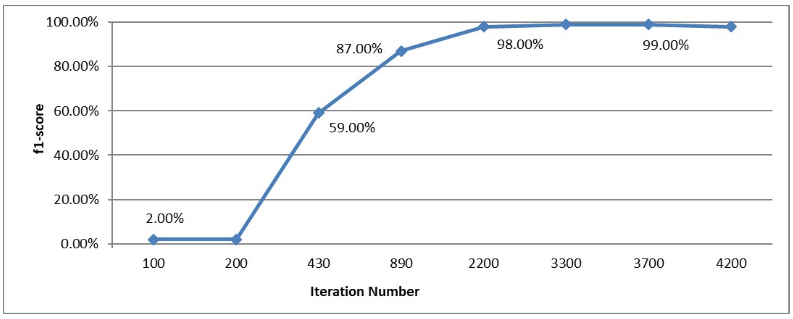

Figure 6. F1-score in the training phase.

4.1 Performance evaluation in the testing phase

For a more robust evaluation, the detection and classification capacity of the network was tested with the test set as described in Section 3.4. As previously stated, the test set consisted of 15% of the images from the MVTec collection and 18 images randomly acquired from Google Images.

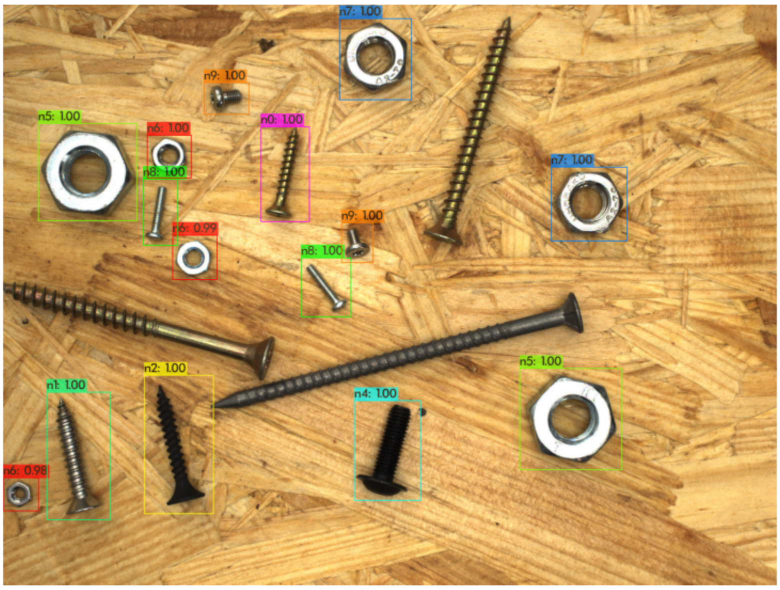

Figure 7. Objects correctly detected and classified in the test phase and their respective bounding boxes. Source: adapted from Ulrich et al. [1].

Figure 7 presents the network performance in the test phase, detecting and classifying screws and nuts in an image from the MVTec dataset, with the network trained at 2,200 epochs, i. The results considering the IoU and F1-Score were, respectively, 77.79% and 98%. According to the IoU value obtained in the test phase, the network successfully detected screws and nuts in the test images. In addition, the F1-Score value (98%) shows the high network capacity in correctly classifying screws and nuts.

5 Conclusions

This work investigated a case study on the problem of detection and classification of screws and nuts. Considering the large number and different object types, computer vision systems can assist experts and ordinary users in the screws and nuts identification. The obtained results in the testing phase (IoU = 77.79% and F1-Score = 98%) in a 10 classes dataset, both in detection and classification, show, if properly configured, the YOLOv4 network can be successfully used in such applications. As can be seen, the results based on the IoU, mAP, and F1-Score metrics, obtained in the training and testing phases are very close, indicating the model shows no signs of overfitting or underfitting, and, therefore, the network parameters are well-adjusted for the application. Since YOLOv4 network makes object detection and classification based on the one-stage approach, bounding boxes are directly predicted over the images. As a result, object detection and classification based on such network architecture are less time-consuming and, therefore, more suitable in real-time applications. In future work, retail software will be developed to be applied in the industry.

Acknowledgements. This work was supported by the CAPES/FAPES (process: 2021-CFT5C, nº FAPES 133/2021) in PDPG (Programa de Desenvolvimento da Pós-Graduação - Parcerias Estratégicas nos Estados). Authors thanks the support from Instituto Federal do Espírito Santo (IFES). The second author is supported by the CNPq under grant #308432/2020-7 (Bolsa de Produtividade DT-2), and by the FAPES under grant #293/2021.

Authorship statement. The authors hereby confirm that they are the sole liable persons responsible for the authorship of this work, and that all material that has been herein included as part of the present paper is either the property (and authorship) of the authors, or has the permission of the owners to be included here.

References

[1] M. Ulrich, P. Follmann, and J. H. Neudeck. A comparison of shape-based matching with deep-learning-based object detection. *tm - Technisches Messen*, vol. 86, n. 11, pp. 685–698, 2019.

[2] A. Bochkovskiy, C. Y. Wang, and H. Y. M. Liao. Yolov4: Optimal speed and accuracy of object detection. *arXiv preprint arXiv:2004.10934*, vol. abs/2004.10934, 2020a.

[3] S. Mangold, C. Steiner, M. Friedmann, and J. Fleischer. Vision-based screw head detection for automated disassembly for remanufacturing. *Procedia CIRP*, vol. 105, pp. 1–6, 2022.

[4] Z. Qiu, X. Zhu, C. Liao, D. Shi, and W. Qu. Detection of transmission line insulator defects based on an improved lightweight yolov4 model. *Applied Sciences*, vol. 12, n. 3, pp. 1270, 2022.

[5] Y. J. Lee, S. H. Lee, and D. H. Kim. Mechanical parts picking through geometric properties determination using deep learning. *International Journal of Advanced Robotic Systems*, vol. 19, n. 1, pp. 17298814221074532, 2022.

[6] Tzutalin. Labelimg. Free Software: MIT License, 2015.

[7] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 779–788. IEEE, 2016.

[8] J. Redmon and A. Farhadi. Yolov3: An incremental improvement. *CoRR*, vol. abs/1804.02767, 2018.

[9] A. C. Espíndola, G. T. d. M. Freitas, and E. F. Nobre Júnior. Pothole and patch detection on asphalt pavement using deep convolutional neural network, 2021.

[10] J. Redmon. Darknet: Open source neural networks in c. `http://pjreddie.com/darknet/`, 2013–2016.

[11] F. Gao and H. Zhong. Study on the large batch size training of neural networks based on the second order gradient. *CoRR*, vol. abs/2012.08795, 2019.

[12] B. Jin, L. Cruz, and N. Gonçalves. Deep facial diagnosis: deep transfer learning from face recognition to facial diagnosis. *IEEE Access*, vol. 8, pp. 123649–123661, 2020.

[13] A. Bochkovskiy, C. Wang, and H. Mark. Yolo v4, v3 and v2 for windows and linux. `https://github.com/ccie29441/Yolo-v4-and-Yolo-v3-v2-for-Windows-and-Linux/`, 2020b.