ISSN: 1803-3814 (Printed), 2571-3701 (Online)
https://doi.org/10.13164/mendel.2020.2.017

DOG FACE DETECTION USING YOLO NETWORK

Alzbeta Tureckova✉, Tomas Holik, Zuzana Kominkova Oplatkova
Tomas Bata University in Zlin, Faculty of Applied Informatics, Czech Republic
tureckova@utb.cz✉, oplatkova@utb.cz

Abstract
This work presents the real-world application of the object detection which belongs to one of the current research lines in computer vision. Researchers are commonly focused on human face detection. Compared to that, the current paper presents a challenging task of detecting a dog face instead that is an object with extensive variability in appearance. The system utilises YOLO network, a deep convolution neural network, to predict bounding boxes and class confidences simultaneously. This paper documents the extensive dataset of dog faces gathered from two different sources and the training procedure of the detector. The proposed system was designed for realization on mobile hardware. This Doggie Smile application helps to snapshot dogs at the moment when they face the camera. The proposed mobile application can simultaneously evaluate the gaze directions of three dogs in scene more than 13 times per second, measured on iPhone XR. The average precision of the dogface detection system is 0.92.

Keywords: *Deep Learning, Deep Convolution Networks, Object Detection, Dog Face Detection, YOLO, iOS Mobile Application.*

Received: 09 October 2020
Accepted: 11 November 2020
Published: 21 December 2020

1 Introduction

Computer vision is an interdisciplinary field concerned with processing image data to gain a high-level understanding of a scene. In other words, it tries to mimic and automate the task that the human visual system can do. The computer vision research area is recently mainly occupied by deep convolutional neural networks [6]. There are four main tasks of understanding the image scene within computer vision with increasing level of complexity as follows: classification, object detection, semantic segmentation and instance segmentation, see Figure 1.

Aside from exhaustively studied image recognition tasks, object detection is an exciting piece of research in the computer vision area. It is closer to a real-world application but simultaneously is more complex. General-purpose object detection should be fast, accurate, and able to recognize a wide variety of objects. Unfortunately, the creation of big object detection datasets is time-consuming and more expensive than simple tagging for classification. Therefore the available datasets for object detection are not as extensive as the classification ones. Still, there are some reasonably successful models for object detection.

Initially, the object detector consisted of two parts. The first module acts as a region proposal, and the second module is a classifier. The most straightforward region proposal was a simple sliding window, which was unfortunately too ineffective. Therefore researches came with different improvements, such as Selective Search [16] or the complex region proposal network presented in Faster R-CNN [15]. Nowadays, many successful object detector architectures consist of a single feed-forward convolutional neural network (CNN) that directly predicts classes and anchor offsets without the need for a second stage per-proposal classification operation. The idea of sliding window came back. While the classical sliding window approach is inefficient since it applies a costly per-region classification hundreds of times, modern detectors allow only a few potential bounding boxes to be considered raw object locations and then amend it by predicting an offset of the actual location of the object. Simultaneously they predict scores for object categories, effectively combining the steps of region proposal and classification.

This approach was proposed by You Only Look Once (YOLO) [13] for the first time. The Single Shot Detector (SSD) [11] presents a similar approach but adds layers of feature maps for each scale. The improvement of the SSD detector by combining it with the state-of-the-art classifier Residual-101 [4] is presented in work [3]. In 2018, the state-of-the-art general-purpose detector is RetinaNet [9]. Its best model won COCO challenge[1] and can detect more than 80 categories with mean average precision (mAP) 55.2 at 0.5 intersection over union (0.5 IOU). RetinaNet with ResNet-101-FPN backbone and a 600-pixel image scale runs in 122 ms on an NVIDIA M40 GPU [9].

In practice, the general-purpose detector might be too demanding to run. Moreover, the application may not need to detect so many objects; one or two classes might be enough. On the contrary, the real-world utilization requires high precision and quick forward pass

[1]For more information about this dataset, please visit the web-page of COCO challenge: http://cocodataset.org/#home

Classification | Object Detection | Semantic Segmentation | Instance Segmentation

Figure 1: Illustration of the fundamental computer vision tasks from a computer science point of view: classification, object detection, segmentation, and instance segmentation.

to be able to run on a cheaper, smaller device like a laptop or a smart-phone in real-time.

This paper focuses on such applicability of the YOLO object detector in the detection of the dog face and its gaze for the purpose of taking beautiful photographs and snapshots. This approach has not been studied by researchers lately. Currently, two papers concerned to the topic of dog identification or analysis were identified. The paper [12] deals with a semantic segmentation and an instance segmentation to discriminate the family of the pet and the breed. In another paper [18], authors used four directional features for the directional edge-based dog and cat face detection that can be processed fast together with a multi-layer classifier to judge a face. The approach presented in this paper processes the same topic of the paper [18] but in a different manner. A YOLO network that detects the dog faces is employed together with landmark detectors provided by a cascade of regressors.

The challenging dog face detection was chosen as an example task for two reasons: (i) the inter-variability of this object class is enormous, (ii) except for humans, dogs may be the most photographed species in the world, and (iii) as mentioned, the dog face analysis is not a theme commonly solved in research papers lately. Moreover, the dog face detection is attractive among people since it can serve many following image/video analysis or tasks, i.e., dog breed identification or gaze estimation. We also show and quantitatively evaluate the implementation of such the dogface detector in an iOS mobile application. The application analyses dogs' gaze direction and allows the user to automatically take pictures of the dogs at the exact moment when they are looking into a camera objective.

2 YOLO Detector

The YOLO system [13] divides the input image into an $S \times S$ grid and predicts B bounding boxes $(x, y, width, height)$ and confidences that the predicted box contains an object. The confidence prediction is described as the Intersection over Union between the predicted box and any ground truth box. Regardless of the number of boxes B, one set of class probabilities C is predicted for each grid cell. Class-specific confidence scores for each box are defined in (1) as multiplication of the conditional class probability and the individual box confidence predictions:

$$Pr(Class_i|Object) * Pr(Object) * IOU_{truth}^{pred} = \\ = Pr(Class_i * IOU_{truth}^{pred}) \quad (1)$$

These scores inform us both how well the predicted box fits the object and the probability of that class appearing in the box. These predictions are encoded as an $S \times S \times (B * 5 + C)$ tensor [13]. The example described in this paper uses: $S = 7$, $B = 5$ and $C = 1$, therefore the final prediction is a $7 \times 7 \times 21$ tensor.

The YOLO model is implemented as a convolutional neural network, consisting of 24 convolutional layers followed by two fully connected layers. The tiny YOLO architecture prioritizes the speed of object detection before the precision. It consists of only nine convolution layers instead of 24. For more information about the network architecture, please refer to the original paper [13].

3 Experiments

This section describes the proposed experiments with the utilized dataset and summarizes the training process. Then both quantitative and qualitative performance of the proposed mobile application is evaluated.

3.1 Datasets

In order to train a robust detector, we merged two different datasets containing the dog's head class with bounding box. For more information about each dataset, please refer to the corresponding subsections below. In total, there are 10 849 images with 11 792 marked dog faces, which were randomly split into two parts; 80 % was used for training and 20 % as a test set. During the training, the maximum input size of the image was set to 300×300 px. Samples of images from the dataset can be seen in Figure 2.

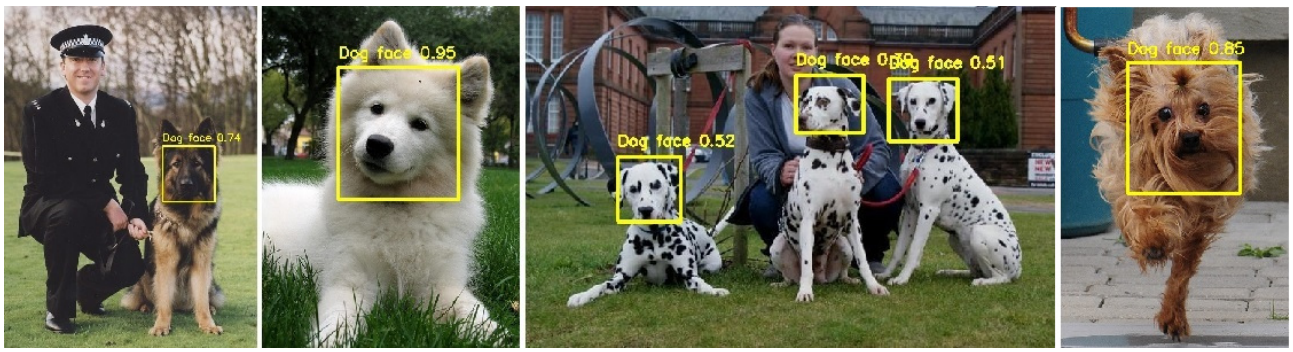

Figure 2: Examples of dog face detection in images randomly chosen from the dataset.

Columbia Dogs Dataset The Columbia Dogs Dataset[1], was introduced in [10]. The dataset contains 8 351 images of 133 different dog breeds. The dataset was revised by authors of this paper, and some additional bounding boxes were added as well as some very loose ones were fixed. Finally, the dataset contains 9 240 marked dog faces.

Oxford-IIIT Pet Dataset The Oxford-IIIT Pet Dataset[2] was created by authors of [12]. It contains 37 pet breed categories with roughly 200 images for each class. For purposes of this experiment, only the dog breed categories were chosen, which makes a total number of 4 978 images with the labeled dog faces.

3.2 Training

The training was implemented in Python with the usage of Keras library [2]. Since the final merged dataset is large enough, no augmentation scheme was applied. The images were only normalized to have pixel values between 0 and 1. The YOLO network was trained using Adam optimizer [8] with the base learning rate of 0.001. The training was done using batch size 64 for 100 epochs. Following the suggestion in [14], the proportions of the bounding boxes used by the YOLO network were generated from training data by the k-means method with $k = 5$. The predicted bounding boxes were enforced to exactly match the size of these anchors during the first three epochs of training. This trick seems to improve precision empirically. The test error was only evaluated once for this model setting.

3.3 Implementation of iOS Application Doggie Smile

The mobile application is implemented in Swift 5, utilizing the library Core ML [5], a machine learning library optimized for on-device performance of a wide variety of model types by leveraging Apple hardware and minimizing memory footprint and power consumption.

The app is designed to automatically choose the right moment to picture the dog(s) in the scene. Following our previous work [17] the detected dog faces are consequently analyzed by landmark detection, locating the eyes, the ears, and the muzzle. The positions of these landmarks were found by a cascade of regressors [7], which allows very fast and reliable detection.

The system computes the Euclidean distance between the muzzle and the left and right ears ($DLEar$ and $DREar$), and similarly, the muzzle and both eyes ($DLEye$ and $DREye$). Thus, it compares the left and the right side (separately for eyes and ears). If the difference does not exceed the threshold defined as 20% of average distance (see equations 2 and 3), we assume that the dog is looking in the direction of the camera lens; and so the system takes a picture.

$$DLEye - DREye <= 0.2 * \frac{DLEye + DREye}{2} \quad (2)$$

$$DLEar - DREar <= 0.2 * \frac{DLEar + DREar}{2} \quad (3)$$

The process of automatic shooting is activated and deactivated by the user. Afterward, the user can choose the final photo from several automatically taken ones where automatically means such photos with the condition of the dog's direct gaze to the camera lens. The maximum number of pictures taken in one run is 10 (can be changed in the user settings).

The two main application screens are shown in Figure 3. We can see three control buttons on the main screen (in Figure 3 on the left). The middle button starts the process of automatic shooting. The right one switches to the display of automatically taken photos (in Figure 3 on the right). Finally, the left one serves to play sound meant to attract the dog attention while a user can choose from different sound variants ranging from cat mow to toy squeaks. The work [5] describes the whole implementation process in detail.

[1]The Columbia Dogs Dataset is available at http://faceserv.cs.columbia.edu/DogData/

[2]The Oxford-IIIT Pet Dataset is available at http://www.robots.ox.ac.uk/~vgg/data/pets/

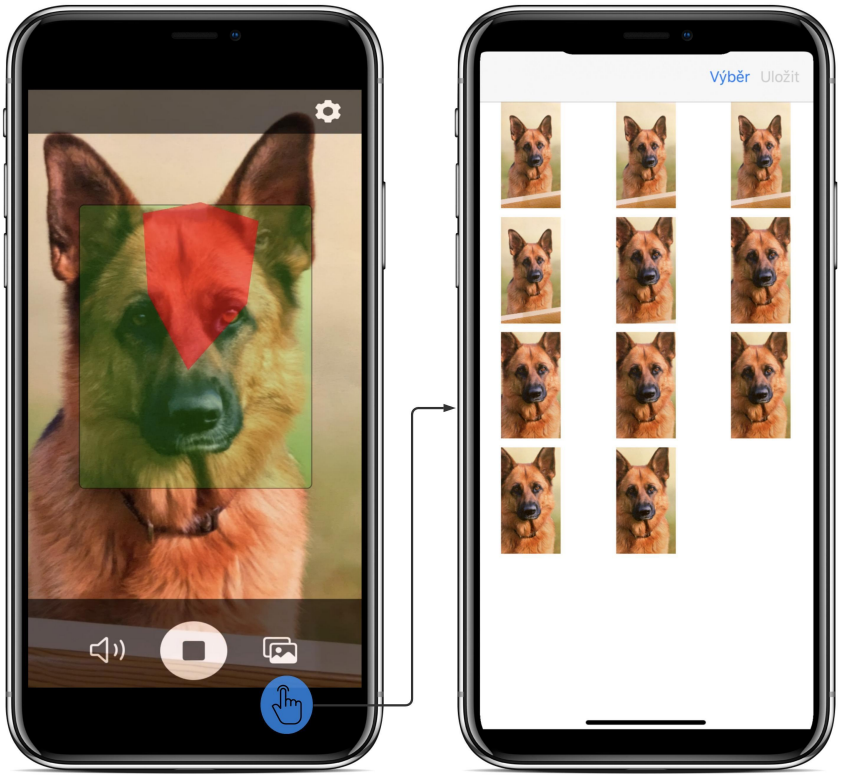

Figure 3: Two main application's screens: automatic shooting screen (on the left) and display of automatically photographed photos (on the right). The shooting screen shows the detected dog face rectangle in green and the identified dogface landmarks in red.

3.4 Results

Two architecture variants of YOLO detector were tested in our experiments, YOLO and YOLO tiny. Table 1 shows the comparison of these architectures from the perspective of time and average precision. It can be stated that the tiny version benefits from its economical design and is 6.5 times faster than the full architecture version while losing just six points in average precision measure. Therefore the YOLO tiny detector was selected for subsequent analysis and mobile implementation. In the following paragraphs, any reference to object detector means the YOLO tiny architecture.

Figure 4 plots the Precision-Recall curve of the YOLO tiny detector using the intersection over union (IOU) 0.5. The average precision (AP) at 0.5 IOU is 0.9150. Samples of pictures with bounding boxes found by the dog-face detector are shown in Figure 2.

We analyzed the evaluation time of critical (key) parts of the proposed mobile application. All tests were averaged from one thousand runs on iPhone XR with processor Apple A12 Bionic and 3 GB RAM. The detection of dog faces in an image of size 300 × 300 pixels takes on average 0.012s (83FPS[3]). The consequent analysis of landmarks takes on average 0.009s (111FPS) for each detected dog face. Overall, including data handling and the gaze estimation, the application is able to run on 17.5FPS if there is one dog in the scene or on 13.2FPS for three dogs in the view. The average evaluation times of critical parts of the application are displayed in Table 2.

4 Discussion

Lately, there is a trend moving from bounding box detection towards mask detection; for example, the COCO object detection challenge[4] features the detection task with object segmentation output (that is, instance segmentation) starting from 2019. The state-of-the-art method [1], which ranks 1st in the COCO 2019 Challenge Object Detection Task with over 500,000 object instances segmented, achieves the 41.2 mask AP (Average Precision) on the test-challenge split, running at 2.1 FPS on a TITAN X GPU. In 2018, the state-of-the-art general-purpose detector was RetinaNet [9]. Its best model can detect more than 80 categories from COCO challenge with 55.2 mAP (mean average precision) at 0.5 IOU (intersection over union). With this setting, the model reached the evaluation speed of 8.2 FPS on an NVIDIA M40 GPU. These powerful

[3]FPS - Frames Per Second, the frequency at which consecutive images called frames appear on display.

[4]For more information about this dataset, please visit the web-page of COCO challenge: http://cocodataset.org/#home

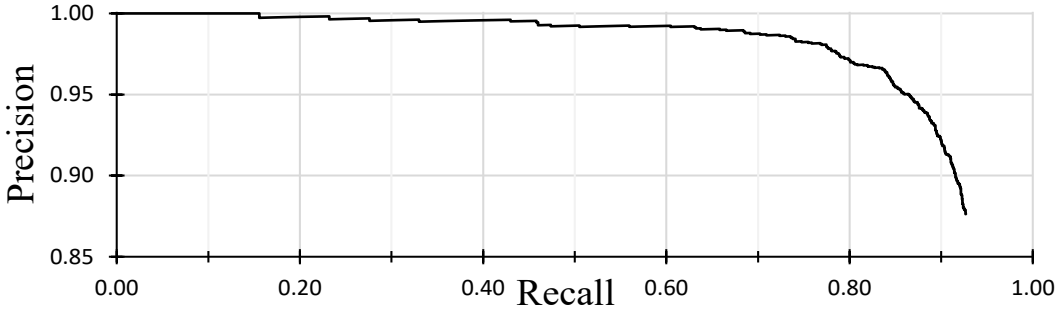

Figure 4: The Precision-Recall curve of the YOLO tiny dog face detector (at 0.5 IOU).

Table 1: Average precision and timing comparison of YOLO dog face detectors

Net architecture	Average precision	Evaluation time [s]	Frame rate [FPS]
YOLO	0.98	0.078	12.8
YOLO Tiny	0.92	0.012	83.3

Table 2: The evaluation time (the mean from 1000 runs) of critical parts of the application, measured on iPhone XR with processor Apple A12 Bionic and 3 GB RAM.

	Evaluation time [s]	Frame rate [FPS]
dog face detection (image size 300×300)	0.012	83.3
landmark detection (per one dog face)	0.009	111.1
overall evaluation (one dog in a scene)	0.057	17.5
overall evaluation (three dogs in a scene)	0.076	13.2

models were unfortunately too exhaustive and sources demanding to run on a small device like a smart-phone.

Fortunately, in practice, the pixel vise detection of each object may not be necessary to enable sufficient insight into the scene happening. Bounding boxes can be enough. Simultaneously, the application does not need to detect as many different object types; one or two classes might suffice. Instead, real-world utilization requires high precision and quick forward pass to run on a cheaper, smaller device like a laptop or a smart-phone in real-time. It is challenging to select an optimal detection network architecture that brings a perfect balance between speed, memory, and accuracy for a specific application on the embedded hardware.

Our paper focuses on such applicability of a dog face detector. We examine two different model variants and compare their suitability for a mobile application. The paper describes the process of building and training the object detection model and also the realization of the iOS mobile application that utilizes it.

5 Conclusion

The paper shows that the object detector based on deep CNN is capable of fast and reliable detection of objects with significant inter variability, i.e., dog faces. A detector enabled to learn a single task from a unique dataset may successfully serve in real-world application where the reliability should be trustworthy, and also the quick processing on a low-cost device is necessary.

The dog face detector, described in this paper, is based on YOLO tiny architecture. As the results show, it achieves 0.92 average precision at 0.5 IOU. Evaluating one image of size 300×300 pixels on the mobile device (iPhone XR) takes 0.012s. Compared to dog-face detection based on Faster RCNN with Resnet 101 extractor presented in [17], the YOLO tiny detector achieves a 0.06 lower score of average precision but is substantially faster.

Detected dog faces are consequently analyzed to find the face landmarks and estimate the dogs' gaze directions, as suggested in [17]. The designed mobile application is able to automatically picture the dogs in a scene in the precious moment when all of the present dogs are looking into the camera. The automatic analysis of dog gaze direction takes on average 0.057s and 0.076s for one and three dogs in the image, respectively. Therefore, the application can examine three dogs' gaze direction in the scene faster than 13 times per second, running on 13.2FPS. Moreover, the paper presents the practical mobile application's layout designed in a way that the user can easily control it.

A similar approach can be readily transferred to realize analogous applications, i.e., detecting a cat's gaze direction or identifying an individual dog/cat to open a pet-door.

Acknowledgement: This work was supported by Internal Grant Agency of Tomas Bata University under the Project no. IGA/CebiaTech/2020/001 and by resources of A.I. Lab (ailab.fai.utb.cz).

References

[1] Chen, K., Pang, J., Wang, J., Xiong, Y., Li, X., Sun, S., Feng, W., Liu, Z., Shi, J., Ouyang, W., Loy, C. C., and Lin, D. Hybrid task cascade for instance segmentation. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)* (2019), pp. 4969–4978.

[2] Chollet, F., et al. Keras. https://github.com/fchollet/keras, 2015.

[3] Fu, C.-Y., Liu, W., Ranga, A., Tyagi, A., and Berg, A. C. Dssd: Deconvolutional single shot detector. *arXiv preprint arXiv:1701.06659* (2017).

[4] He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2016), pp. 770–778.

[5] Holik, T. Doggiesmile: An ios/iphone application for video dog detection. Master's thesis, Tomas Bata University in Zlin, faculty of Applied Informatics, 2020.

[6] Huang, J., Rathod, V., Sun, C., Zhu, M., Korattikara, A., Fathi, A., Fischer, I., Wojna, Z., Song, Y., Guadarrama, S., et al. Speed/accuracy trade-offs for modern convolutional object detectors. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2017), pp. 7310–7311.

[7] Kazemi, V., and Sullivan, J. One millisecond face alignment with an ensemble of regression trees. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2014), pp. 1867–1874.

[8] Kingma, D. P., and Ba, J. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[9] Lin, T.-Y., Goyal, P., Girshick, R., He, K., and Dollár, P. Focal loss for dense object detection. In *Proceedings of the IEEE international conference on computer vision* (2017), pp. 2980–2988.

[10] Liu, J., Kanazawa, A., Jacobs, D., and Belhumeur, P. Dog breed classification using part localization. In *European conference on computer vision* (2012), Springer, pp. 172–185.

[11] Liu, W., Anguelov, D., Erhan, D., Szegedy, C., Reed, S., Fu, C.-Y., and Berg, A. C. Ssd: Single shot multibox detector. In *European conference on computer vision* (2016), Springer, pp. 21–37.

[12] Parkhi, O. M., Vedaldi, A., Zisserman, A., and Jawahar, C. Cats and dogs. In *2012 IEEE conference on computer vision and pattern recognition* (2012), IEEE, pp. 3498–3505.

[13] Redmon, J., Divvala, S., Girshick, R., and Farhadi, A. You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2016), pp. 779–788.

[14] Redmon, J., and Farhadi, A. Yolo9000: better, faster, stronger. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (2017), pp. 7263–7271.

[15] Ren, S., He, K., Girshick, R., and Sun, J. Faster r-cnn: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems* (2015), pp. 91–99.

[16] Uijlings, J. R., Van De Sande, K. E., Gevers, T., and Smeulders, A. W. Selective search for object recognition. *International journal of computer vision 104*, 2 (2013), 154–171.

[17] Vlachynska, A., Oplatkova, Z. K., and Turecek, T. Dogface detection and localization of dogface's landmarks. In *Computer Science Online Conference* (2018), Springer, pp. 465–476.

[18] Yamada, A., Kojima, K., Kiyama, J., Okamoto, M., and Murata, H. Directional edge-based dog and cat face detection method for digital camera. In *2011 IEEE International Conference on Consumer Electronics (ICCE)* (2011), pp. 87–88.