Proceedings of the 3rd International Conference on Computing Innovation and Applied Physics
DOI: 10.54254/2753-8818/30/20241073

Unmanned aerial vehicle face recognition technology research progress

Jiakang Hao[1,3] and Xinglu Zhu[2]

[1]College of Mechanical Engineering and Automation, Beihang University, Beijing, China
[2]Shien-ming Wu School of Intelligent Engineering, South China University of Technology, Guangzhou, China

[3]21375416@buaa.edu.cn

Abstract. In recent years, face recognition technology has become a significant advance in the area of biometrics and machine vision. It paves the way for a wide range of applications, including security systems, access control, surveillance, and user authentication. These applications are undoubtedly a major innovation in the field of UAV, which will greatly expand and extend its application scenarios. This paper shows a compositive review of UAV facial identification technology, including its basic principles, techniques, challenges, and ethical considerations. According to the different stages, this paper focuses on RCNN and YOLO, two more widely used target detection technologies and their respective technical iterations, and through the comparison of the two in terms of technical characteristics and application scenarios, the advantages of the two are obtained, and combined with the current UAV workflow. Get the stage in which they play a specific role. This paper reviews the existing literature and research on face recognition, aiming to help people better understand the current process of this technology and its wider social application and impact.

Keywords: UAV, face recognition technology, RCNN, YOLO

1. Introduction

UAV face recognition can achieve data collection secretly, large-scale group data collection, and high mobility mobile data collection, and has great application prospects in traffic control, target tracking, and other aspects. To improve the precision of UAV face recognition technology, scholars have innovated and improved the data types and algorithms required for face recognition technology. The principle of face recognition is based on the captured human facial features, which is simplified to compare the identifying feature information with the existing information in the database. The face recognition process can be divided into three steps: capture portrait photos, detect and raise feature points, and compare feature points. In this paper, the algorithms are divided into face recognition technology based on first-stage target detector and two-stage target detector according to different stages. This paper mainly analyses the YOLO series in the first-order object detection algorithm and the R-CNN series in the second-order object detection algorithm, and subdivides these types in each stage detector, and the function principle, differences, and common points of each algorithm in the face recognition process are analysed, then it can obtain the research progress of UAV face recognition technology. This paper conducted an analysis of the data obtained from previous algorithm evaluations on The PASCAL VOC dataset, specifically focusing on the YOLO series and RCNN series. We constructed a framework for unmanned aerial vehicle (UAV) face recognition tasks based on the YOLO series and RCNN series and compare and analyse the performance of these two series in terms of certain metrics. Summarize the advantages and disadvantages between different kinds of algorithms, then identify the challenges encountered in the development of UAV face recognition technology utilizing these series and proposed potential solutions, considering the current advancements in UAV face recognition technology, the challenges and improvement directions for the future development of UAV face recognition technology are discussed at last.

2. Object detection technology

The first step in the facial recognition process relies on object detection technology. Unlike image classification tasks, object detection not only solves the classification problem but also addresses the localization problem, making it a multitask problem.

The development of the algorithm can be bifurcated: the traditional testing algorithm stage and the deep learning-based testing algorithm stage. The latter further develops into two technology routes based on anchor-based and anchor-free methods. The two detectors mentioned above with different extraction frequencies are the products of two different algorithms under the anchor-based method. The algorithm separates the testing issue into two phases: first generating candidate areas using convolutional neural networks or other methods, then classifying the candidate areas. Fast-RCNN and Faster-RCNN are typical representatives. Although this algorithm has low error rates, it has low detection rates for vulnerabilities, slow speed, and is not suitable for real-time monitoring scenarios, and naturally, it does not meet the requirements of drone facial recognition tasks [1].

For the one-stage approach, this algorithm only uses a single CNN network to produce the class probabilities and position coordinates of the targets directly, and the final testing results can be got in one testing process. Even though the precision is low, it is not slow, such as YOLO, SSD, etc.

3. Two-stage

3.1. Fast-RCNN

As the first deep learning-based method used in object detection, the RCNN algorithm has greatly contributed to the research and development in this field. Based on RCNN, Fast R-CNN introduced Spatial Pyramid Pooling and ROI Pooling, resulting in improved accuracy and simplified training steps. In the same network environment, training with R-CNN takes 84 hours, whereas Fast R-CNN only requires 9.5 hours. Additionally, Fast R-CNN reduces the inference time from 47 seconds to 0.32 seconds compared to R-CNN [2]. The main structure of Fast R-CNN can be described by Figure 1.

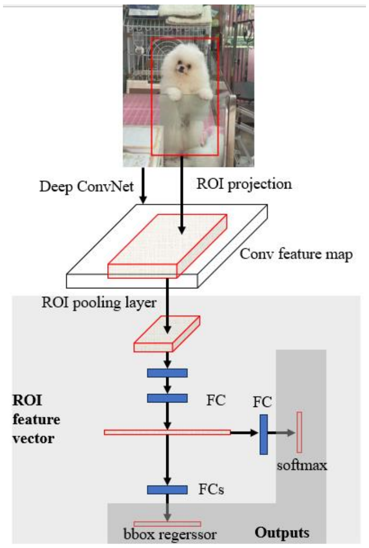

Figure 1. The main structure of Fast R-CNN [2]

3.2. *Faster-RCNN*

To address the drawback of time-consuming selective search for region proposal generation in Fast R-CNN, Shaoqing Ren et al. introduced Faster R-CNN, which utilizes a Region Proposal Network. By leveraging convolutional operations instead of traditional region proposal methods, Faster R-CNN optimizes the process of selecting candidate regions and improves the quality of the resulting proposals, thereby enhancing the accuracy of object detection.

The PASCAL VOC, serving as a competition that provides training resources for testing object detection algorithms, includes openly available image datasets, ground truth annotations, standardized evaluation software, and an annual competition and workshop. Researchers utilize the PASCAL VOC dataset to perform performance tests on various algorithms. The dataset is constructed based on the concept taxonomy, as shown in Table 1.

Table 1. The concept taxonomy [3].

Vehicles	Household	Animals	Other
Aero plane Bicycle Boat Bus Car Motorbike Train	Bottle Chair Dining table Potted plant Sofa TV	Bird Cat Cow Dog Horse Sheep	Person

Based on the publicly available information, the test results on the PASCAL VOC2012 dataset can be inferred in Table 2.

Table 2. The test results on the PASCAL VOC2012 dataset [4].

Method	Proposals	Training data	COCO val		COCO test-dev	
			mAP@.5	mAP@[.5,.95]	mAP@.5	mAP@[.5,.95]
Fast R-CNN	SS,2000	COCO train	-	-	35.9	19.7
Faster R-CNN	RPN,300	COCO train	41.5	21.2	42.1	21.5

From the above figure, it is obvious that as an upgraded version of Fast R-CNN, Faster R-CNN achieves 2% to 3% higher mean average precision (mAP) in object detection. In speed testing environment with a single NVIDIA Tesla P100 GPU, Faster R-CNN achieves around 10 FPS for test set 7 and model VGG-16. Despite multiple improvements made to the RCNN series, the speed of the RCNN series is still not optimal.

4. One-stage

4.1. YOLO

The feature extraction network typically utilizes a pre-trained CNN model (such as Darknet) to process input images through multiple convolutional and pooling layers, extracting high-level features from the images. These features contain information about the shape, texture, and other characteristics of the targets.

After feature extraction, the obtained feature maps are fed into the detection network. The detection network consists of a series of convolutional and fully connected layers, which predict the bounding boxes and class probabilities of the targets. The detection network has a compact and efficient structure.

In the detection network, each unit is responsible for predicting a set of bounding boxes and their corresponding probabilities. Each bounding box is expressed as a set of coords, including the position of the top-left and bottom-right corners, as well as the classification probability of the corresponding target. By using anchor boxes, a group of pre-defined boxes with disparate dimensions as well as aspect ratios, you can predict targets of different sizes.

The detection network achieves object detection by dividing the feature maps into grid cells of different sizes and predicting the bounding boxes and class probabilities within each cell. This one-pass design makes the YOLO algorithm efficient and real-time. Additionally, the algorithm can handle challenges like overlapping objects and small targets.

By minimizing the loss between the predicted bounding boxes and the ground truth boxes during the training process, the YOLO detection network can adjust its weights through backpropagation, improving the exactitude of object testing. Continuous training and optimization allow the algorithm to better adapt to specific tasks and scenarios [5].

4.2. YOLO v6

YOLOv6 is a single-stage object testing framework designed specifically for applications on industries, surpassing other real-time detectors in terms of accuracy and speed. Compared to YOLOv5, it introduces the repVGG backbone, which provides more feature representation capability in a compact network, while maintaining similar inference speed. It also incorporates the cspStackRep module used in larger models. The neck of YOLOv6 adopts a PAN topology structure, enhanced by repBlocks or cspstackRep Blocks. To improve efficiency and simplify decoupling heads, an efficient decoupling head is introduced to make it more efficient. The framework uses variation loss for classification, SiOU GiOU loss for regression, and RepopTinizer for training to obtain ptq-friendly weights.

YOLOv6 achieves impressive performance on the COCO dataset, with an AP of 35.9% for YOLOv6-n and 43.5% for YOLOv6-s. The new version of YOLOv6-s achieves the latest levelof AP of 43.3% with the speed of 869 FPS. Compared to other detectors (YOLOv5, YOLOX, PPYOLOE) with similar inference speeds, YOLOv6-m/L also reaches a more accuracy performance (49.5%/52.3%) [6].

The performance on the COCO dataset and the high throughput of the NVIDIA Tesla T4 GPU demonstrate its potential for real-time object testing in industrial applications.

In addition, YOLOv6 also introduces industry-friendly improvements such as self-distillation and quantization, which contribute to enhancing its performance. And, Table 3 showed the results of contrasts with other YOLO detectors on the COCD DS.

Table 3. The results of contrasts with other YOLO detectors on the COCO DS [6].

Method	Input Size	$\mathbf{AP}^{val}$	$\mathbf{AP}^{val}_{50}$	FPS (bs=1)	FPS (bs=32)
YOLOv5-N	640	28.00%	45.70%	602	735
YOLOv5-S	640	37.40%	56.80%	376	444

Table 3. (continued).

YOLOv5-M	640	45.40%	64.10%	182	209
YOLOv5-L	640	49.00%	67.30%	113	126
YOLOX-Tiny	416	32.80%	50.30%	717	1143
YOLOX-S	640	40.50%	59.30%	333	396
YOLOX-M	640	46.90%	65.60%	155	179
YOLOX-L	640	49.70%	68.00%	94	103
PPYOLOE-S	640	43.10%	59.60%	327	419
PPYOLOE-M	640	49.00%	65.90%	152	189
PPYOLOE-L	640	51.40%	68.60%	101	127
YOLOv7-Tiny	416	33.30%	49.90%	787	1196
YOLOv7-Tiny	640	37.40%	55.20%	424	519
YOLOv7	640	51.20%	69.70%	110	122
YOLOv6-N	640	35.90%	51.20%	802	1234
YOLOv6-T	640	40.30%	56.60%	449	659
YOLOv6-S	640	43.50%	60.40%	358	495
YOLOv6-M	640	49.50%	66.80%	179	233
YOLOv6-L-ReLU	640	51.70%	69.20%	113	149
YOLOv6-L	640	52.50%	70.00%	98	121

4.3. YOLO v7

YOLOv7, proposed by Chien-Yao Wang, is a trainable bag-of-freebies object detector that improves the exactitude of object testing by solving the problem of substitution of reparametrized modules and the assignment problem of dynamic tag assignment, enabling state-of-the-art results in real-time object detection. Its biggest feature is that it is trained from scratch on the MS COCO DS and haven't used any additional DS or other pre-trained weights. The model uses a composite scaling method to amplify the deepness of the computational section by 1.5 times and the width of the transform block by 1.25 times. It employs techniques such as depth supervision, assigning tags using soft tags, and using reparametrized modules to improve the exactitude of object testing.

Compared to YOLOv6, YOLOv7 achieves higher accuracy, with AP 13.7 improvement compared to the most precise yolov6-s model on the COCO dataset. At the same time, because YOLOv7 reduces the number of parameters and the number of calculations. (For example, YOLOv5-x6 (r6.1) has 45 fewer parameters and 63 fewer calculations compared to Yolov5-x6 (r6.1).), resulting in faster inference and outperforming YOLOv7 models such as YOLOv6-W6 and 8 fps faster than YOLOR-P6 [7].

YOLOv7 obtains a higher AP than YOLOv6. For example, YOLOv7-d6 has a similar speed of inference to YOLOR-E6 while has a 0.8 increase on AP [7].

The inference speed of yolov7-e6e is similar to that of YOLOR-D6 but improves AP by 0.3 [].

Weikai He et al. present a real-time tiny object testing algorithm called YOLOv7-UAV, it builds on the YOLOv7 algorithm with the introduction of several improvements, including the deletion of the second-downsampling layer and the deepest probe head, the introduction of the dpsPPF module, the majorization of the K-means algorithm, and the use of weighted normalized Gauswasserstein distance (nwd) and joint intersection (IoU) as instructions for sample allocation [8].

Table 4 showed the comparison of baseline object detectors. Experimental results show that the real-time detection speed of YOLOv7-UAV is at least 27% higher than that of YOLOv7, while significantly reducing the number of parameters and gFLOP. In terms of the average exactitude (maps) of the visdrone2019 and TinyPerson datasets, it also outperformed YOLOv7, improving by 2.89% and 4.30%, respectively [8].

Table 4. Comparison of baseline object detectors [7]

Model	**#Param.**	**FLOPs**	**Size**	$\mathbf{AP}^{val}$	$\mathbf{AP}_{50}^{val}$	$\mathbf{AP}_{75}^{val}$	$\mathbf{AP}_{S}^{val}$	$\mathbf{AP}_{M}^{val}$	$\mathbf{AP}_{L}^{val}$
YOLOv4	64.4M	142.8G	640	49.7%	68.2%	54.3%	32.9%	54.8%	63.7%
YOLOR-u5 (r6.1)	46.5M	109.1G	640	50.2%	68.7%	54.6%	33.2%	55.5%	63.7%
YOLOv4-CSP	52.9M	120.4G	640	50.3%	68.6%	54.9%	34.2%	55.6%	65.1%
YOLOR-CSP	52.9M	120.4G	640	50.8%	69.5%	55.3%	33.7%	56.0%	65.4%
YOLOv7	36.9M	104.7G	640	**51.2%**	**69.7%**	**55.5%**	**35.2%**	**56.0%**	**66.7%**
improvement	-43%	-15%	-	+0.4	+0.2	+0.2	+1.5	=	+1.3
YOLOR-CSP-X	96.9M	226.8G	640	52.7%	**71.3%**	71.3%	36.3%	57.5%	68.3%
YOLOv7-X	71.3M	189.9G	640	**52.9%**	71.1%	**71.1%**	**36.9%**	**57.7%**	**68.6%**
improvement	-36%	-19%	-	+0.2	-0.2	+0.1	+0.6	+0.2	+0.3
YOLOv4-tiny-3l	8.7	5.2	320	30.8%	47.3%	32.2%	**10.9%**	31.9%	51.5%
YOLOv7-tiny	6.2	3.5	320	**30.8%**	**47.3%**	**32.2%**	10.0%	**31.9%**	**52.2%**
improvement	-39%	-49%	-	=	=	=	-0.9	=	+0.7
YOLOR-E6	115.8M	683.2G	1280	55.7%	73.2%	60.7%	40.1%	**60.4%**	69.2%
YOLOv7-E6	97.2M	515.2G	1280	**55.9%**	**73.5%**	**61.1%**	**40.6%**	60.3%	**70.0%**
improvement	-19%	-33%	-	+0.2	+0.3	+0.4	+0.5	-0.1	+0.8
YOLOR-D6	151.7M	935.6G	1280	56.1%	73.9%	61.2%	**42.4%**	60.5%	69.9%
YOLOv7-D6	154.7M	806.8G	1280	56.3%	73.8%	61.4%	41.3%	60.6%	70.1%
YOLOv7-E6E	151.7M	843.2G	1280	**56.8%**	**74.4%**	**62.1%**	40.8%	**62.1%**	**70.6%**
improvement	=	-11%	-	+0.7	+0.5	+0.9	-1.6	+1.6	+0.7

4.4. YOLO v8

YOLOv8 is the latest generation of the YOLO algorithm, which was open-sourced by Ultralytics. It is the next major version following the YOLOv5 and currently supports tasks such as image classification, object detection, and instance segmentation.

Jordan Kupec et al. propose a broad model for real-time testing of articles of flight, which is trained on datasets containing 40 separate types of flying objects, and a fine model achieved by transferring learning on datasets that represent real-world environments. These models address challenges such as differences in object space size/aspect ratio, velocity, occlusion, and clustering background. They utilized the YoloV8 single-shot detector, considered to be the most advanced detector currently available, to find the best trade-off between inference speed and Map, and provided an in-depth explanation of the architecture and capabilities of YOLOv8 [9].

The final generalized model achieves an average inference speed of 0.685 for Map50-95 and 50fps on a 1080p video, while the improved model maintains the same inference speed, with Map50-95 improving by 0.835 [9].

5. Comparison and analysis

The application of unmanned aerial vehicle (UAV) face recognition algorithms in UAV face recognition tasks requires consideration of multiple factors, such as photo clarity, information transmission speed, platform information processing speed, and accuracy. The general workflow for UAV face recognition tasks is shown in Figure 2.

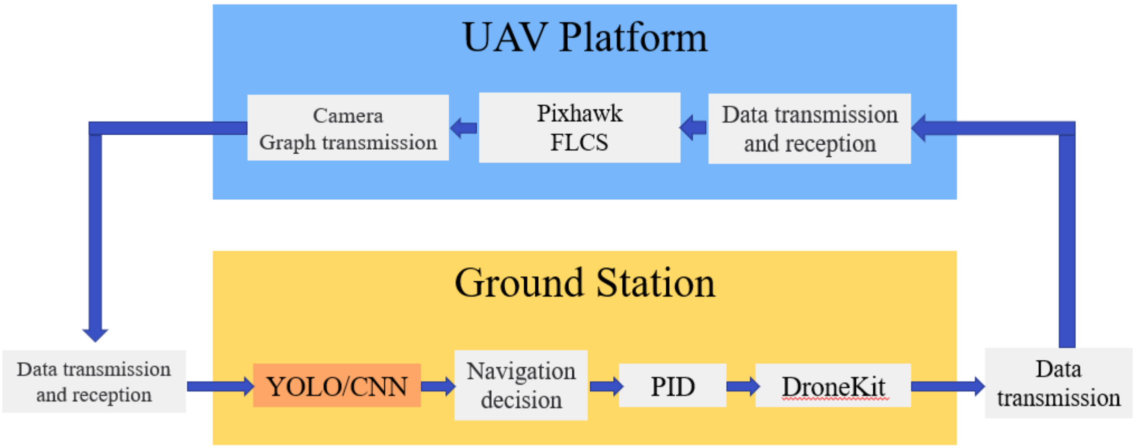

Figure 2. The general workflow for UAV face recognition tasks (photo credited: original)

First, after the UAV reaches the target location, the camera captures images and uploads them to the ground information processing end. The ground station uses YOLO series or RCNN series algorithms to process the received images and determines the next position of the UAV to obtain a clearer image of the target information. Then, the control signals for PID, etc., are transmitted to the UAV platform for motion control, and the process of capturing images is repeated.

The application of YOLO series and RCNN series algorithms in UAV face recognition technology needs to consider the specific execution scenarios of UAV tasks. Next, we will analyze the advantages and disadvantages of YOLO series and RCNN series in three UAV face recognition task scenarios:

- Scenario 1: When the UAV is performing personnel control in a densely populated area, where accuracy is more important than speed and there is a lower requirement for timeliness. Most algorithms in this situation are based on Faster R-CNN, for instance, the OR-CNN. Therefore, Faster R-CNN is more suitable in this situation.
- Scenario 2: When the UAV needs to search for missing persons, individuals with mental disorders, or suspects in criminal investigations and requires timely location information, YOLO series algorithms are suitable. The YOLO series only needs one forward pass to complete bounding box localization and category recognition.
- Scenario 3: When the UAV needs to track special individuals movement and rapidly transmit their location information, a combination of RCNN series and YOLO series is more suitable. The YOLO series can accurately identify the target and quickly upload location information.

This paper focuses on analyzing the YOLO family of one-stage algorithms and the R-CNN family of two-stage algorithms that can be used in UAV face recognition. In the RCNN series, Fast R-CNN to Faster R-CNN, although the candidate region selection improvement, but still need to go through the detection and selection of the corresponding candidate region and consume a lot of time, while the YOLO series directly on the original target detection eliminates the candidate region of the relevant process, the detection speed is greatly improved, but in the use of scenarios requiring relatively accurate detection of the situation However, in the case where the usage scenario requires relatively accurate detection, the two stage algorithm represented by the RCNN series is more suitable and has higher accuracy.

6. Conclusion

At present, the latest version of YOLO V8 has been updated in August 2023, and the development of the target detection algorithm field has been further advanced. In the application to UAV face detection, there are still corresponding problems that need to be solved and appropriate solution is considered as well, such as how to efficiently capture the face features and the corresponding projection when there is an occluded object, how to balance the speed and accuracy of UAV face recognition, how to quickly and accurately recognize and mark the target face in a wide range of people, etc.

In order to deal with the questions above, We can introduce an Occlusion-aware R-CNN or a PED detector [10]. By combining additional detectors with base detectors such as YOLO series and RCNN series, we can improve the detection accuracy in crowded areas. We can also use the recently introduced Partial Occlusion-aware Region of Interest pooling unit to merge prior structural information and visibility prediction into the network, thereby addressing object occlusion and face recognition in large crowds. Furthermore, we can reduce the number of proposal boxes, use feature pyramids, predefined boxes for localization and classification, and simplify the network structure to balance the accuracy and speed of UAV face recognition technology. It is believed that in the information age, with the iterative development of algorithms and the continuous improvement of theoretical knowledge, the object detection algorithms required for UAV face recognition will be able to make great progress.

Authors Contribution

All the authors contributed equally and their names were listed in alphabetical order.

References

[1] Alexey Bochkovskiy, Chien-Yao Wang, Hong-Yuan Mark Liao. (2020). YOLOv4: Optimal Speed and Accuracy of Object Detection. arXiv.org

[2] Girshick R. Fast R-CNN[J]. Computer Science, 2015.

[3] Everingham, M., Eslami, S. M., A., Van Gool, L., Williams, C. K., I., et al. (2015). The pascal visual object classes challenge: A retrospective. International Journal of Computer Vision, 111(1), 98-136

[4] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun, “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks”, Advances in Neural Information Processing Systems, 2015.

[5] Joseph, Redmon., Santosh, K., Divvala., Ross, Girshick., Ali, Farhadi. (2016). You Only Look Once: Unified, Real-Time Object Detection. arXiv.org

[6] Chuyin, Li., Lu, Li., Hongliang, Jiang., Kaiheng, Weng., Yifei, Geng., Lin, Li., Zaidan, Ke., Qingyuan, Li., Meng, Cheng., Weiqiang, Nie., Yiduo, Li., Yufei, Liang., Linyuan, Zhou., Xiaoming, Xu., Xiangxiang, Chu., Xiaoming, Wei., Xiaolin, Wei. (2022). YOLOv6: A Single-Stage Object Detection Framework for Industrial Applications. arXiv.org

[7] Chien-Yao, Wang., Alexey, Bochkovskiy., Hong-Yuan, Mark, Liao. (2022). YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors. arXiv.org, abs/2207.02696

[8] Weikai, He. (2023). YOLOv7-UAV: An Unmanned Aerial Vehicle Image Object Detection Algorithm Based on Improved YOLOv7. Electronics, doi: 10.3390/electronics12143141

[9] Jordan, Kupec. (2023). Real-Time Flying Object Detection with YOLOv8. arXiv.org

[10] Shifeng Zhang, Longyin Wen, Xiao Bian, Zhen Lei, and Stan Z. Li, “Occlusion-aware R-CNN: Detecting Pedestrians in a Crowd”, Computer Science, 2018.