Smoke Detection Based on YOLOv5 and MediaPipe

Yusong Wang, Rouyi Fan, Xiaofeng Li*

School of Artificial Intelligence and Big Data, Henan University of Technology, Zhengzhou, Henan, China

**Corresponding Author*

Abstract: As awareness of smoking prohibition in public places increases, technology that automatically detects smoking behavior is particularly important. With the rapid development of deep learning and neural networks, the typical You Only Look Once version 5 (YOLOv5) algorithm can be used for prediction, the traditional algorithm performs well on the picture, but the detection on the video will have false detection, so the MediaPipe machine learning framework is introduced to improve the accuracy of video detection, and the relative distance calculation of the target's hand, mouth, and the position of the cigarette detected by YOLOv5 can comprehensively judge whether the object smokes. Deploy an improved algorithm to call the camera or import the video for detection, and the items similar to cigarettes in the video can be eliminated, reducing the probability of false detection.

Keywords: Object Detection; Deep Learning; Machine Learning; MediaPipE; YOLOv5

1. Introduction

One of the major global public health concerns is smoking. Effectively monitoring and controlling smoking behavior has become crucial to public health management, particularly in light of the growing implementation of smoking prohibition regulations in public places [1]. Even though there are manual monitoring techniques and conventional smoke detectors, they frequently have problems like false alarms and delayed responses, which makes it challenging to meet real-time monitoring requirements. Image recognition-based smoking behavior monitoring techniques are progressively showing considerable application promise as deep learning and computer vision technologies advance quickly [2]. Smoking behavior in photos can be effectively detected by using deep learning algorithms such as the You Only Look Once (YOLO) series, which offer precise and ongoing answers, especially in intricate and changing surveillance situations.

The main object detection algorithms used in computer vision-based smoking habit monitoring currently are the YOLO series, which can partially recognize important aspects of smoking behavior like smoke and cigarettes [3]. The YOLO algorithm's drawback is that, while it performs exceptionally well on static photos, it frequently misses and makes false detections in dynamic video surveillance since cigarette objects are little and have a propensity to blend in with other objects or the background. Additionally, current research frequently ignores the dynamic aspects of smoking behavior, especially the hand and mouth movements, which have a significant impact on the precision and usefulness of smoking behavior identification.

By selecting the best model YOLOv5x.pt [4] from many YOLO series models, and integrating it with the MediaPipe model to increase detection accuracy, this work seeks to suggest a new technique for monitoring smoking behavior. This study hypothesize that false detections from long-distance shooting can be effectively reduced and that comprehensive weighted judgment can provide more accurate smoking behavior identification by simultaneously detecting the cigarette object and the dynamic changes of the mouth and hands [5]. Specifically, the MediaPipe model will track the dynamic movements of the hands and lips, while YOLO will identify the cigarette item. It is anticipated that a thorough examination of these three components will improve the ability to identify smoking behavior.

The optimum model for cigarette target recognition was found by comparing several models, and a novel approach that combined the MediaPipe and YOLOv5 models was put forth, addressing the issue of monitoring smoking habits from several angles. In addition to improving the system's ability to judge smoking behavior, position detection for the mouth and hands also successfully lessens the false detection issue that arises with the single YOLOv5 model in complicated situations. Furthermore, this study's multi-model fusion method [6] offers a more reliable and accurate way to track smoking habits in real time, with a high application value.

2. MediaPipe Model and YOLOv5 Model

2.1 YOLOv5's Basic Architecture

YOLOv5 is a variation of the YOLO series. It significantly improves speed, accuracy, and usability over earlier iterations. YOLOv5's primary benefits are multi-scale prediction via multiple detection heads, effective feature extraction, and the capacity to detect targets at various scales. A model that is appropriate for object detection is constructed by dissecting the model structure layer by layer. The following major components shown in Figure 1 make up the structure of YOLOv5 [7].

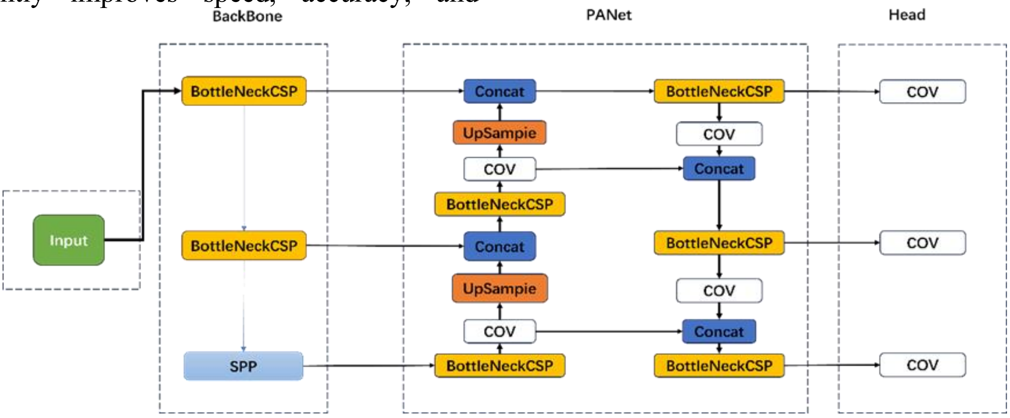

Figure 1. Basic Architecture of YOLOv5

YOLOv5's input module consists of the focus module and picture preprocessing. The focus module processes the supplied image using a unique slicing technique. In particular, it quadruples the number of channels and halves the input image's spatial dimensions (height and width) by rearranging each 2x2 pixel block to correspond to a single pixel position in the output feature map. For instance, the size of the output feature map from a 640*640*3 input image after processing by the Focus module is 320x320x12. By increasing the number of channels, this operation not only lessens the computational load for later network processing but also preserves local image information, which enables the model to better capture the fine details of small targets [8], greatly increasing the accuracy of small target detection. This solution establishes a strong basis for upcoming feature extraction and object identification tasks by deftly striking a balance between computing efficiency and feature extraction capacity.

The backbone of the feature extraction network is responsible for extracting useful feature maps from the input image. The feature extraction network used by YOLOv5 is called CSPDarknet53 (Cross-Stage Partial connections Darknet53), which is an enhanced Darknet53 that incorporates the CSP structure to decrease computing burden and increase computational efficiency.

The Neck network's function is to further process the backbone's features in order to make object detection at various scales easier. The neck structure of YOLOv5 is PANet (Path Aggregation Network), which efficiently combines data from various scales to improve the network's detection capabilities for both tiny and large objects.

YOLOv5's detection head is in charge of producing class, object location, and confidence predictions as well as handling the last object detection duty. At each scale, YOLOv5 employs a number of detection heads, each of which makes a prediction about the target class and location [9].

2.2 The MediaPipe Model

Real-time computer vision tasks like position estimation, face recognition, and gesture recognition are frequently performed using MediaPipe, a cross-platform, high-performance multimedia processing framework that Google suggested [10]. This system uses MediaPipe as one of its main elements to identify hand, facial, and position information in real time. It complements the YOLOv5 model to improve the system's robustness and recognition accuracy for smoking behavior.

The MediaPipe model's entire processing flow in this system is depicted in Figure 2. The structure of system consists of four primary phases: the input stream, preprocessing, inference, and postprocessing. The entire processing cycle begins when the system receives image frames from the camera or video as an input stream. The image moves on to the preprocessing phase, which entails data augmentation, normalization, and image scaling. The model's capacity to generalize is enhanced by data augmentation techniques, including image rotation and brightness adjustments; normalization stabilizes the pixel value distribution, and resizing unifies the input to the size required by the model.

The image is transmitted to the inference step for object detection and feature extraction following preprocessing. In this step, MediaPipe employs neural networks to identify important points in the image using its pre-trained models (HandLandmark, FaceMesh, Pose, etc.). The inference stage produces intermediate detection findings, including key point coordinates and their confidence levels [11].

The postprocessing phase ensues, with the goal of further refining the inference outcomes. First, redundant or low-confidence detections are eliminated using result filtering. Thereafter, the structured detection data is formatted to satisfy the downstream jobs' data needs. Lastly, functional modules like behavior identification, alarm system activation, or health prompt creation obtain the processed data as output.

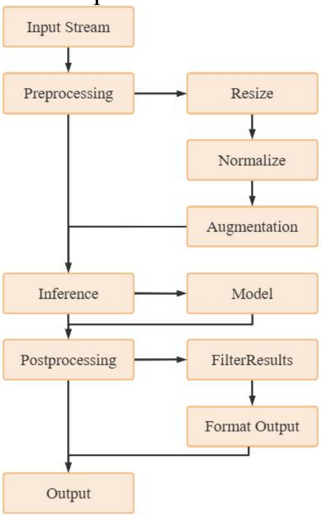

Figure 2. Structure of the MediaPipe Model

The MediaPipe model's high real-time speed and simplicity of deployment are its most significant advantages in this system. The entire processing flow can operate steadily on mobile and embedded devices thanks to its internal implementation, which blends GPU acceleration methods with graph calculation pipelines. Additionally, MediaPipe's modular design makes it easy for developers to combine various detection models (hand, face and pose, for example) to accomplish multi-dimensional feature fusion. This is crucial for detection tasks involving spatial, temporal, and contextual cross-features, such as smoking behavior [12].

In addition to being the system's main module for extracting picture features, MediaPipe's transparent process structure also shows how well-thought-out and effective the structural architecture of contemporary real-time visual systems is. This system effectively and accurately detects smoking behavior in complicated environments and responds in real-time by utilizing its potent multi-modal perception capabilities.

3. Preparing and Processing Datasets

The MediaPipe model and YOLO series models are the primary components of this algorithm model. The dataset simply needs to satisfy YOLO's standards because the latter mostly uses Google's pre-trained models. This experiment primarily employed two datasets: the first is a public dataset from AI Studio that includes a variety of file types, as illustrated in Figure 3. Nearly half of the files are in the JPEG format, with the remaining 7.7% being in the JPG and JSON file types. "Smoke" is a publicly available dataset from GitCode that includes statistics on file types: 2472 jpeg files, 2 cache files, and 2472 txt files. There are 2472 image files in the collection, out of a total of 4946.

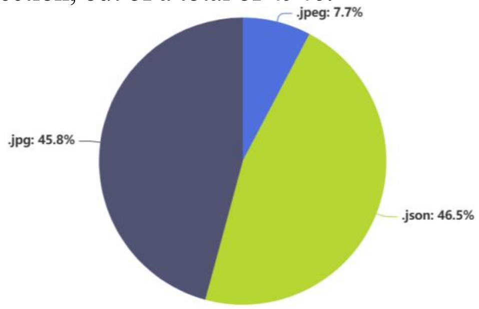

Figure 3. AI Studio Public Smoking Dataset Proportion

The original dataset's photos and associated JSON annotation files underwent uniform format conversion and organizing to comply with the YOLO model's data format criteria. At the same time, the preprocessed dataset was automatically divided and rearranged into training, validation, and test sets to ensure the model training's generalization capability and meet the data organization requirements of the YOLO training framework. To generate the appropriate configuration file data.yaml, it was also necessary to standardize the data reading format during YOLO model training.

4. The Process of Experimentation

4.1 The Experimental Setting

The compiler used was PyCharm. JetBrains created PyCharm, a Python Integrated Development Environment (IDE) that is popular in data science and Python programming domains. It offers a wealth of strong features to assist developers in increasing the effectiveness and caliber of their code. It enables the installation of related dependencies and the construction of separate environments for every project by allowing direct administration of Python virtual environments. Additionally, it facilitates project dependency management by supporting package management systems like Pip and Conda. Table 1 displays the computing environment that was utilized.

Table 1. Environment of the System

Item	Information
Operating System	Windows 10 10.0.26100
Platform	Windows-10-10.0.26100-SP0
Python Version	3.8.18 (default, Sep 11 2023, 13:39:12) [MSC v.1916 64 bit (AMD64)]
Number of CPU Cores	6
Number of CPU Logical Cores	12
CPU Usage 13.9%	13.9%
Number of GPU Devices	1
GPU 0	NVIDIA GeForce RTX 3050 Laptop GPU
CUDA Version	PyTorch CUDA 11.8

4.2 Measures of Evaluation

Accurately assessing model performance is essential for comprehending its advantages and disadvantages in object detection jobs. Precision, recall, mean average precision at Intersection over Union (IoU) is 0.5 (mAP_0.5), and the extended mean average precision over IoU thresholds from 0.5 to 0.95 (mAP_0.5:0.95) were chosen as the primary evaluation measures to thoroughly assess the model's detection ability. These metrics offer a scientific foundation for model comparison and optimization by reflecting the model's performance in the detection job from several perspectives.

IoU is a crucial statistic for determining precision and recall. It quantifies the degree of correspondence between the ground truth box and the projected bounding box. The IoU formula is:

$$\text{IoU}=\frac{\text{Area of Intersection}}{\text{Area of Union}}=\frac{\text{Area}(A\cap B)}{\text{Area}(A\cup B)} \quad (1)$$

The value range of IoU is [0,1]. Where: Area of Union: The union area of predicted box A and ground truth box B. Region of Intersection: The overlapping region between forecasted box A and ground truth box B. A number nearer 1 indicates a greater match between the predicted box and the ground truth box.

Precision a crucial indicator of the model's prediction accuracy, precision shows the percentage of really positive samples among those that were anticipated to be positive (i.e., detected targets). The precision formula is:

$$\text{Precision}=TP/(TP+FP) \quad (2)$$

FP (False Positives) is the number of instances in which the model incorrectly predicts background or other non-target areas as targets, i.e., detection results where the IoU between the predicted box and the ground truth box is less than the set threshold. TP (True Positives): The number of targets that the model finds correctly, i.e., detection results where the IoU between the predicted box and the ground truth box is greater than or equal to a set threshold (e.g., 0.5).

A higher level of precision means that the model has a lower false positive rate and is better at evaluating targets during detection. But focusing only on achieving high precision could make the model overly cautious and overlook some real goals.

Recall assesses the model's capacity for detection by concentrating on the percentage of all real, active targets that the model is able to identify. The recall formula is:

$$\text{Recall}=TP/(TP+FN) \quad (3)$$

The term FN refers to the number of real targets that the model failed to identify, which includes ground truth boxes that were not accurately matched by any forecast box.

Greater recall denotes a model's improved capacity to detect targets, allowing for greater coverage of all real, active targets. High recall, however, could come at the expense of precision, resulting in a higher number of false positives from the model.

MAP_0 metric computes the average precision across all recall levels, providing a thorough assessment of the model. The precise procedure is as follows: Rank the predicted boxes for each category according to confidence, then use the ground truth boxes to determine their IoU value. A TP is defined as an IoU ≥ 0.5, a FP otherwise, and FN is defined as unmatched ground truth boxes. This amount is used to compute precision at various recall levels and plot the precision-recall curve. According to the formula below, the Average Precision (AP) for that category is represented by the area under this curve. Lastly, mAP_0.5, which fully represents the model's detection ability across all categories, is obtained by averaging the AP values for each category.

$$AP=\sum_{r=0}^{1}Precision(r)\times\Delta Recall(r) \qquad (4)$$

In this context, ΔRecall (r) represents the small change in recall, while r denotes the range of recall values, which spans from 0 to 1.

When evaluating the model's detection performance for various target categories, mAP_0.5 strikes a compromise between precision and recall. It is a commonly used metric that is particularly well-suited for the thorough assessment of detection completeness and accuracy in object detection activities.

MAP_0.5:0.95 expanded evaluation metric is used to more thoroughly assess the model's performance under various detection accuracy requirements. In contrast to mAP_0.5, mAP_0.5:0.95 takes into account several scenarios in which the IoU threshold varies from 0.5 to 0.95 (in steps of 0.05), in addition to the situation when the IoU threshold is equal to 0.5. The model's performance under various accuracy criteria can be more accurately reflected by mAP_0.5:0.95, which is especially significant for situations demanding high-precision detection. It examines the model's detection performance under higher accuracy requirements (e.g., IoU=0.95) and its approximate localization ability for targets (e.g., IoU=0.5). mAP_0.5:0.95 [13] allows for a more thorough assessment of the model's resilience and flexibility in various situations.

4.3 Comparing Multiple Models

To identify the model most suited for the target detection job, we performed a performance comparison analysis of several models in the YOLO series. We found the best model by comparing the evaluation parameters of six models: YOLOv5n, YOLOv5s, YOLOv5x, YOLOv8m, YOLOv8n, and YOLOv8x [14]. Table 2 below lists each model's primary parameters.

The performance of six models from the YOLOv5 and YOLOv8 series was compared in the study using the same number of 10 training epochs in the test for detecting smoking habits. With a precision of 0.76690, recall of 0.73324, mAP_0.5 of 0.73984, and mAP_0.5:0.95 of 0.36214, the YOLOv5x model outperformed the other models by a large margin, indicating its outstanding detection accuracy and stability, according to the results. The overall detection efficacy of the YOLOv8 series was nevertheless somewhat lower than that of YOLOv5x, although it had reasonably balanced performance in certain lightweight models (for example, YOLOv8n had mAP_0.5 of 0.60107 and mAP_0.5:0.95 of 0.34212). To guarantee excellent recognition accuracy and dependability in real-world situations, YOLOv5x was finally chosen as the target detection model for this system after taking into account both detection accuracy and application requirements. The best model was chosen, and training was conducted using it. Smaller batches were first used for comparison due to the model's size; in order to perhaps obtain better parameters, training was then done on a server. Table 3 displays the outcomes of the server's training.

Table 2. Comparison of Multi-Model Evaluation Parameters

Model	Precision	Recall	mAP_0.5	mAP_0.5:0.95	Epoch
yolov5n.pt	0.65193	0.61200	0.54197	0.26836	10
yolov5s.pt	0.66274	0.61497	0.57296	0.29061	10
yolov5x.pt	0.76690	0.73324	0.73984	0.36214	10
yolov8m.pt	0.63868	0.62567	0.59335	0.35047	10
yolov8n.pt	0.64780	0.64800	0.60107	0.34212	10
yolov8x.pt	0.66578	0.63280	0.61207	0.29921	10

Table 3. Training Log for Servers

Epoch	Precision	Recall	mAP_0.5	mAP_0.5:0.95
0	0.6088	0.4300	0.4945	0.1907
1	0.7719	0.7450	0.8048	0.3240
2	0.7927	0.7150	0.7998	0.3558

3	0.8385	0.7010	0.7647	0.3574
4	0.8518	0.7700	0.8293	0.4251
5	0.7669	0.7300	0.7398	0.3621
6	0.8343	0.7200	0.8208	0.4419
7	0.8279	0.7450	0.7865	0.4219
……	……	……	……	……
28	0.9479	0.9150	0.9335	0.5520
29	0.9430	0.9200	0.9295	0.5403

Figure 4 shows that the model has a satisfactory fitting effect on the dataset because both the accuracy and the loss rate meet high standards. Since just the smoking target is recognized, the cls _ loss stays at 0.

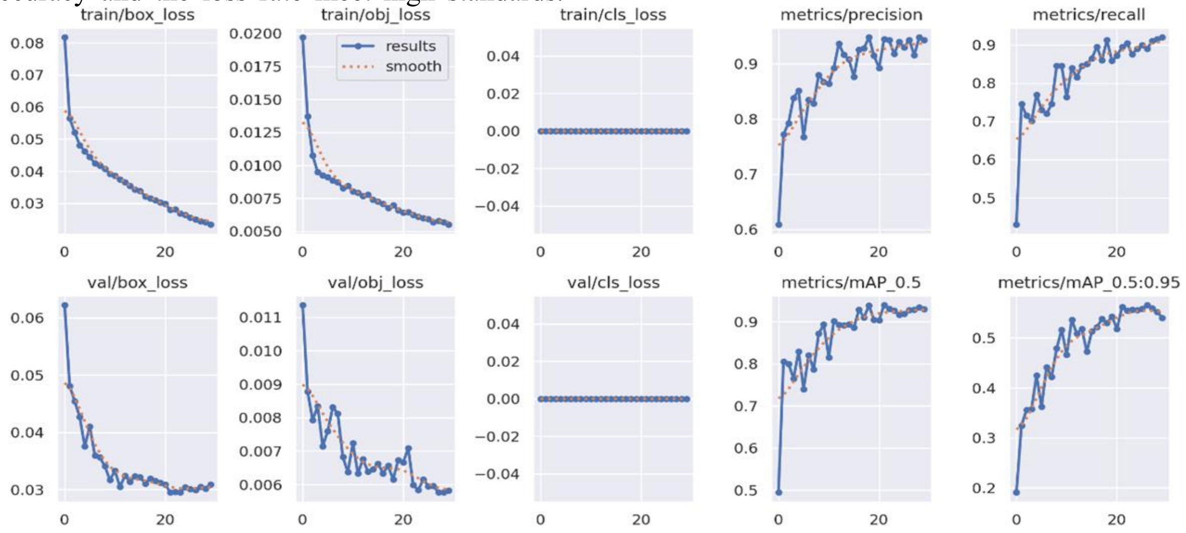

Figure 4. Parameter Change Trend

4.4 Integrating MediaPipe and YOLOv5 Models

We were able to obtain an ideal YOLO model after finishing all of the previously specified work. When applied to video detection, this model produced false positives for things that resembled cigarettes, despite its excellent performance in image recognition. To make a thorough assessment based on the relative positions of the cigarette and human body parts, the MediaPipe model was developed and integrated with YOLOv5x [15]. Figure 5 displays its process structure:

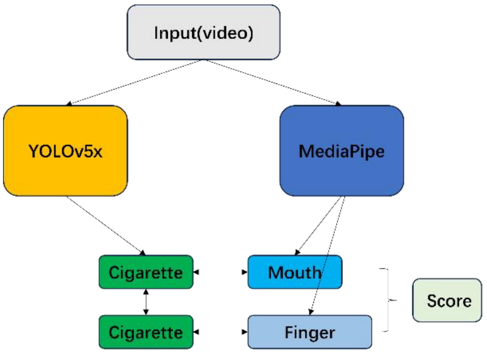

Figure 5. Diagram of System Operation Following Combination

The system's overall workflow is as follows: 21 hand key points are extracted using the MediaPipe Hands module for hand recognition. A cigarette-holding action is deemed to exist when the Euclidean distance between the tips of the index and middle fingers (landmarks 8 and 12) drops below a predetermined threshold. The mouth center point (landmark 13) is further located by obtaining 468 face key points using the MediaPipe FaceMesh module for facial detection.

It is possible to identify whether smoking is taking place by measuring the distance between the mouth point and the center of the cigarette detection box. The system's video processing flow is built using OpenCV. To obtain the cigarette target box, each frame is first sent to the YOLOv5 model. At the same time, the current frame's hand and face key points are extracted. Then, two fundamental functions, is _ mouth _ near _ cigarette () and is _ hand _ holding _ object (), ascertain whether the mouth is close to the cigarette target and whether a finger-holding cigarette action is present, respectively. The system recognizes smoking behavior in that frame and marks prompt information if at least one of these two requirements is satisfied. Figure 6 shows an example of a video frame.

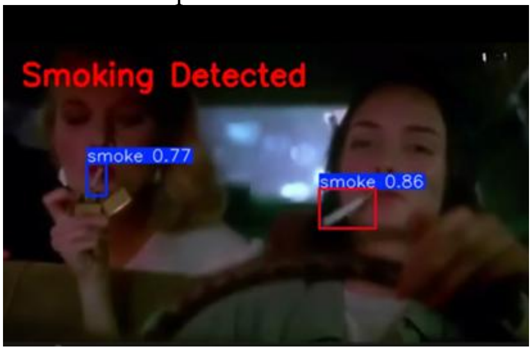

Figure 6. Example Diagram for Detection

The method also incorporates a frame-count-based confidence-building strategy for smoking behavior to lessen sporadic miscalculations. The system outputs the overall recognition result after processing every video frame and calculating the percentage of frames where smoking behavior was recognized. This approach strikes a compromise between stability and real-time performance.

5. Conclusion

YOLOv5 and MediaPipe were integrated in the experimental effort to create a reliable and effective smoking behavior detection system. The YOLOv5 model is the main object identification framework. It quickly finds and recognizes cigarette targets. To address the drawback that using bounding boxes alone makes it challenging to appropriately assess behavioral semantics, MediaPipe was added concurrently to achieve high-precision extraction of important hand and facial points. By combining these two visual modules, the system is able to recognize and comprehend smoking behavior more deeply by analyzing if the mouth is close to the cigarette in addition to determining whether a cigarette is being carried by the hand.

The system showed satisfactory accuracy and real-time performance in testing, which makes it appropriate for real-world applications, including intelligent health intervention and surveillance video analysis. Future studies may further add behavioral temporal analysis models (like LSTM or Transformer) to enhance the system's comprehension of intricate action sequences. Meanwhile, it is anticipated that the system's generalization and adaptability under various conditions and individual differences will be enhanced through multi-source data fusion and cross-domain transfer learning.

In conclusion, the experiment confirmed the viability and efficiency of multi-modal visual technology in challenging behavior detection tasks, offering fresh perspectives and a useful basis for research on smoking behavior recognition.

References

[1] Liu Z D. Building a Smoke-Free Environment Together, Creating a Healthy Life Together. Population & Health, 2024, (09): 81-82.

[2] Li L, Wang S L. Research on the Implementation of a Smoking Monitoring System Based on Digital Technology. Electronic Engineering & Product World, 2024, 31(10): 32-35.

[3] Cheng C, Liu Q H, Chen Y T, et al. Analysis and Implementation of a Smoking Behavior Recognition and Detection System Based on YOLOv5. Computer Knowledge and Technology, 2024, 20(30): 17-20.

[4] Jiang X, Hu H, Liu X, et al. A smoking behavior detection method based on the YOLOv5 network. Journal of Physics: Conference Series. IOP Publishing, 2022, 2232(1): 012001.

[5] Maia F W, Lopes M A, David A S. Automatic sign language to text translation using MediaPipe and transformer architectures. Neurocomputing, 2025, 642: 130421.

[6] Guo H, Sun C, Zhang J, et al. MMYFnet: Multi-Modality YOLO Fusion Network for Object Detection in Remote Sensing Images. Remote Sensing, 2024, 16(23): 4451.

[7] Wang Z, Wu L, Li T, et al. A smoke detection model based on improved YOLOv5. Mathematics, 2022, 10(7): 1190.

[8] Zheng Y P, Xu B Y, Wang Z Y. An Improved YOLOv5 Smoke Detection Model. Journal of Computer Engineering and Applications, 2023, 59(7).

[9] Zhang J, Yin L, Gong X F, et al. Small Target Smoke Detection Algorithm Based on Improved YOLOv5. China Safety Science Journal, 2023, 33(11): 75.

[10] Wang R B, Dou Q L, Zhang Q, et al. MediaPipe-Based Hand Gesture Recognition for Teleoperated Control of Excavators. Journal of Information Technology in Civil Engineering and Architecture, 2022, 14(4): 9-16.

[11] Ni G X, Xu H, Wang C. Research on Hand Gesture Recognition Integrating Improved

YOLOv5 and Mediapipe. Journal of Computer Engineering and Applications, 2024, 60(7).

[12]Cao W, Lu P, Cao W. Multimodal Gesture Recognition with Spatio-Temporal Features Fusion Based on YOLOv5 and MediaPipe. International Journal of Pattern Recognition and Artificial Intelligence, 2024, 38(08).

[13]Shi F, Zhou H, Ye C, et al. Faster detection method of driver smoking based on decomposed YOLOv5. Journal of Physics: Conference Series. IOP Publishing, 2021, 1993(1): 012035.

[14]Paramita C, Supriyanto C, Putra K R. Comparative analysis of YOLOv5 and YOLOv8 cigarette detection in social media content. Scientific Journal of Informatics, 2024, 11(2): 341-352.

[15]Cui Z W, Liu X F, Xia S N, et al. Research on Hand Gesture Recognition Based on Deep Learning. Journal of Sensor Technology and Application, 2024, 12: 570.