
Journal of Advanced Artificial Intelligence, Engineering and Technology

*Research Article*

**WRJAAIET-25-014**

# Human Action Recognition Using YOLOv11 Ultralytics: A Comprehensive Study for Real-Time Applications

**Seung Jin Kim**[1*] **and Myuhng Joo Kim**[2]

[1]AI Convergence Engineering, Assist University, Seoul, Korea
[2]Department of Computer Science, Assist University, Seoul, Korea

***Correspondence:** Seung Jin Kim, AI Convergence Engineering, Assist University, Seoul, Korea, E-mail: phd_eng.kim@stud.assist.ac.kr;
DOI: https://doi.org/10.56147/aaiet.1.2.14

**Citation:** Kim SJ, Kim MJ (2025) Human Action Recognition Using YOLOv11 Ultralytics: A Comprehensive Study for Real-Time Applications. J Adv Arti Int Eng & Techn 1: 14.

## Abstract

Human Action Recognition (HAR) is a pivotal task in computer vision, with applications in surveillance, healthcare, robotics and human-computer interaction. This study presents a novel framework for HAR using the YOLOv11 model by Ultralytics, a state-of-the-art object detection architecture optimized for real-time performance. We trained and evaluated the model on a custom dataset comprising 18 distinct human actions, captured in indoor environments using fisheye cameras. The actions range from everyday activities (*e.g.,* walking, sitting) to specialized tasks (*e.g.,* patient on stretcher, patient on wheelchair). Our results show that YOLOv11 achieves a mean Average Precision (mAP@0.5) of 0.401, with exceptional performance on actions like cleaning (mAP@0.5: 0.760), searching (mAP@0.5: 0.695) and patient on wheelchair (mAP@0.5: 0.995). We provide an in-depth analysis of the model's training metrics, bounding box distributions, precision-recall curves, F1-confidence curves, recall-confidence curves and confusion matrices. Additionally, we present extensive qualitative results to demonstrate the model's robustness in real-world scenarios. A comparison with existing methods, such as two-stream CNNs and Transformer-based models, highlights YOLOv11's superior balance of accuracy and speed, making it a promising solution for real-time HAR applications. This study also discusses the model's limitations and outlines directions for future research, paving the way for enhanced action recognition systems.

**Keywords:** Human action recognition; YOLOv11; Ultralytics; Deep learning; Computer vision; Real-time detection; Surveillance; Healthcare

**Received date:** April 14, 2025; **Accepted date:** April 17, 2025; **Published date:** May 05, 2025

## Introduction

Human Action Recognition (HAR) is a critical task in computer vision, enabling machines to understand and interpret human behaviors in various contexts. HAR has wide-ranging applications, including security surveillance (*e.g.,* detecting suspicious activities in public spaces), healthcare (*e.g.,* monitoring patient activities in hospitals), robotics (*e.g.,* enabling human-robot interaction) and entertainment (*e.g.,* gesture-based gaming). The ability to accurately classify human actions in real-time is essential for these applications, requiring robust models that can handle complex scenes, varying lighting conditions and diverse action types.

Traditional HAR methods often relied on handcrafted features, such as Histogram of Oriented Gradients (HOG) and Scale-Invariant Feature Transform (SIFT), combined with classifiers like Support Vector Machines (SVMs) [1,2]. These approaches, while effective in controlled settings, struggled with dynamic environments due to their reliance on manual feature engineering and limited generalization capabilities. The advent of deep learning has transformed HAR, with Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) enabling end-to-end learning of spatial and temporal features directly from raw data [3,4].

Among deep learning approaches, the YOLO (You Only Look Once) family of models has emerged as a powerful

framework for object detection and more recently, action recognition [5-7]. YOLO models are known for their single-stage detection pipeline, which balances accuracy and speed, making them ideal for real-time applications. The latest YOLOv11 model by Ultralytics introduces several improvements, including a more efficient backbone, enhanced feature aggregation and advanced data augmentation techniques, further boosting its performance.

In this study, we propose a YOLOv11-based framework for HAR, focusing on its ability to recognize a diverse set of human actions in indoor settings. We collected a custom dataset comprising 18 action classes, including everyday activities like walking and sitting, as well as specialized tasks like patient care in hospitals. Our evaluation includes a comprehensive analysis of training metrics, bounding box distributions, precision-recall curves and confusion matrices, providing a detailed understanding of the model's performance. We also present extensive qualitative results to demonstrate the model's robustness in real-world scenarios and compare YOLOv11 with other state-of-the-art methods to highlight its advantages.

The contributions of this paper are as follows:

- Development of a YOLOv11-based framework for human action recognition, leveraging its real-time capabilities.
- Comprehensive evaluation of the model on a custom dataset with 18 action classes, achieving an mAP@0.5 of 0.401.
- Detailed analysis of training metrics, bounding box distributions, precision-recall curves, F1-confidence curves, recall-confidence curves and confusion matrices.
- Extensive qualitative results demonstrating the model's performance in diverse indoor scenarios.
- Comparison with existing HAR methods, such as two-stream CNNs and Transformers, to showcase YOLOv11's superiority in accuracy and speed.
- Insights into the model's limitations and potential directions for future research.

The rest of the paper is organized as follows: Section II reviews related work in HAR. Section III describes the dataset, model architecture and training pipeline. Section IV presents the results and discussion, including training metrics, class wise performance, bounding box analysis, qualitative results and comparisons. Section V concludes the paper and outlines future work.

## Related Work

Human action recognition has been a topic of extensive research in computer vision for decades. Early approaches relied on handcrafted features to capture spatial and temporal information from video sequences. For example, Dalal and Triggs introduced the Histogram of Oriented Gradients (HOG) for human detection, which was later adapted for action recognition by combining it with motion features like optical flow [1]. Similarly, Lowe proposed the Scale-Invariant Feature Transform (SIFT) for extracting key points, which were used with classifiers like Support Vector Machines (SVMs) to recognize actions [2]. These methods, while effective in controlled settings, struggled with complex scenes due to their reliance on manual feature engineering and limited robustness to variations in lighting, occlusion and viewpoint.

The introduction of deep learning marked a significant shift in HAR research. Convolutional Neural Networks (CNNs) enabled the automatic extraction of spatial features from images, eliminating the need for handcrafted features [3]. Simonyan and Zisserman proposed a two-stream CNN architecture that processes spatial (RGB frames) and temporal (optical flow) information separately, achieving state-of-the-art performance on benchmark datasets like UCF101 and HMDB51 [8-10]. However, two-stream networks are computationally expensive, as they require separate processing of RGB and optical flow streams, making them less suitable for real time applications.

Recurrent Neural Networks (RNNs) and their variants, such as Long Short-Term Memory (LSTM) networks, were also widely used for temporal modeling in HAR [4,11]. Donahue et al. proposed the Long-term Recurrent Convolutional Network (LRCN), which combines CNNs for spatial feature extraction with LSTMs for temporal sequence modeling [12]. While effective, these approaches often suffer from vanishing gradient problems and high computational costs, limiting their applicability in real-time scenarios.

The YOLO family of models has emerged as a powerful alternative for action recognition, particularly in real time applications [5-7]. YOLO models use a single-stage detection pipeline that predicts bounding boxes and class probabilities in a single forward pass, achieving a balance between accuracy and speed. Early versions like YOLOv3 were adapted for HAR by incorporating temporal information or pose estimation [6,13]. YOLOv5 and YOLOv7 further improved performance with better feature extraction and faster inference [7].

The latest YOLOv11 model by Ultralytics introduces enhancements such as a more efficient Feature Pyramid Network (FPN), improved anchor-free detection and advanced data augmentation techniques, making it well-suited for HAR tasks.

Recent advancements have also explored Transformer-based models for HAR. Dosovitskiy et al. introduced the Vision Transformer (ViT), which applies self-attention mechanisms to image patches for feature extraction [14]. Liu proposed the Swin transformer, which uses a hierarchical architecture to capture multi-scale features, achieving state-of-the-art performance on action

recognition datasets [15]. However, Transformer models are computationally intensive, requiring significant resources for training and inference, which limits their use in real-time applications compared to YOLOv11.

Our work builds on the strengths of YOLOv11, adapting it for HAR in indoor environments. Unlike previous studies, we focus on a diverse set of actions, including both everyday activities and specialized tasks relevant to healthcare and surveillance. We also provide a comprehensive evaluation of the model's performance, including training metrics, bounding box analysis and qualitative results, to offer a holistic understanding of its capabilities.

# Methodology

## Dataset

We collected a custom dataset for human action recognition, comprising video footage from indoor environments such as offices, hospitals and public spaces. The footage was captured using fisheye cameras with a resolution of 1280×720 pixels, providing a wide-angle view of the scenes. The dataset includes 18 distinct action A. classes, reflecting a mix of everyday activities and specialized tasks:

- **Everyday activities:** Walking, sitting, sitting on desk, standing, eating, talking, using phone, sleeping.
- **Specialized tasks:** Patient on stretcher, patient on wheelchair, stretcher attendant, wheelchair attendant, checking bag, holding walkie talkie, keeping walkie talkie charging.
- **Maintenance activities:** Cleaning, mopping, loitering.
- **Miscellaneous:** Triple intersection, working, battery low, searching, background.

Each action class was manually annotated with bounding boxes and labels using a custom annotation tool. The dataset contains over 10,000 frames, split into 80% for training (8,000 frames), 10% for validation (1,000 frames) and 10% for testing (1,000 frames).

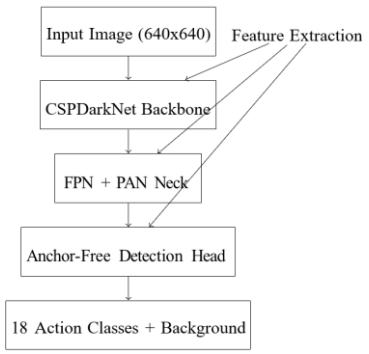


**Figure 1:** YOLOv11 architecture for human action recognition.

The distribution of actions is imbalanced, with classes like standing and walking having significantly more samples (2385 and 1131 instances, respectively) compared to underrepresented classes like checking bag and stretcher attendant (0 instances correctly predicted). To address this imbalance, we applied data augmentation techniques, including random rotations, flips and brightness adjustments, to increase the diversity of the training data.

## Model architecture

YOLOv11 is an advanced object detection model developed by Ultralytics, building on the strengths of its predecessors (YOLOv5, YOLOv7). It uses a single-stage detection pipeline that predicts bounding boxes and class probabilities in a single forward pass, making it highly efficient for real-time applications. The architecture consists of three main components:

- **Backbone:** A Convolutional Neural Network (CNN) that extracts features from the input image. YOLOv11 uses a modified CSPDarkNet backbone, which incorporates Cross Stage Partial (CSP) connections to improve gradient flow and reduce computational complexity.
- **Neck:** A feature aggregation module that combines features from different layers of the backbone. YOLOv11 employs a Feature Pyramid Network (FPN) with Path Aggregation Network (PAN) to capture multi-scale features, enhancing the detection of objects at various sizes.
- **Head:** A detection head that predicts bounding boxes, objectless scores and class probabilities. YOLOv11 uses an anchor-free approach, predicting the center, width and height of bounding boxes directly, which improves detection accuracy for small objects.

For HAR, we modified the YOLOv11 model to predict 18 action classes plus a background class, resulting in 19 output classes. The input images were resized to 640x640 pixels to balance computational efficiency and detection accuracy. The model was pre-trained on the COCO dataset and finetuned on our custom dataset to adapt to the specific action classes. **Figure 1** illustrates the YOLOv11 architecture for HAR [16].

## Training pipeline

The model was trained using the Ultralytics YOLOv11 framework, implemented in PyTorch. The training pipeline is summarized in Algorithm 1.

Algorithm 1 YOLOv11 training pipeline for HAR.

- **Input:** Training dataset Dtrain, validation dataset Dval, number of epochs E, batch size B, learning rate η
- **Output:** Trained YOLOv11 model
- Initialize YOLOv11 model with pre-trained weights from COCO dataset

Journal of Advanced Artificial Intelligence, Engineering and Technology

- **Set hyperparameters:** B=16, η =0.01, E=200
- **Apply data augmentation:** Random flips, rotations, brightness adjustments
- for epoch=1 to E do
- Shuffle Dtrain and create mini-batches of size B for each batch in Dtrain do
- **Forward pass:** Compute predictions (bounding boxes, class probabilities)
- **Compute loss:** $L=L_{box}+L_{cls}+L_{dfl}$
- **Backward pass:** Update model weights using Adam optimizer with learning rate η
- end for
- Evaluate model on Dval
- **Compute validation metrics:** box loss, classification loss, DFL loss, mAP@0.5, mAP@0.5:0.95
- if validation mAP@0.5 does not improve for 10 epochs then
- **Early stopping:** Break
- end if
- end for
- **Return:** Trained model.

The model was trained for 200 epochs on an NVIDIA RTX 3090 GPU with 24 GB of VRAM. We used the Adam optimizer with a learning rate of 0.01 and a batch size of 16. The loss function consists of three components:

- **Box Loss (Lbox):** Measures the localization error between predicted and ground-truth bounding boxes using Generalized Intersection over Union (GIoU) loss.
- **Classification Loss (Lcls):** Measures the error in class predictions using binary cross-entropy loss.
- **Distribution Focal Loss (Ldfl):** A novel loss introduced in YOLOv11 to improve the regression of bounding box coordinates by modeling the distribution of box predictions.

The total loss is a weighted sum of these components:

$L=L_{box}+L_{cls}+L_{dfl}$. We applied early stopping to prevent overfitting, halting training if the validation mAP@0.5 did not improve for 10 consecutive epochs.

## Evaluation metrics

We evaluated the model using a comprehensive set of metrics to assess its performance across different dimensions:

- **Mean Average Precision (mAP):** Calculated at IoU thresholds of 0.5 (mAP@0.5) and 0.5:0.95 (mAP@0.5:0.95). mAP@0.5 measures the model's performance at a single IoU threshold, while mAP@0.5:0.95 averages the mAP across multiple IoU thresholds, providing a more robust evaluation.
- **Precision and recall:** Assessed across different confidence thresholds to generate precision-recall curves. Precision measures the proportion of correct positive predictions, while recall measures the proportion of positive instances correctly detected.
- **F1-score:** The harmonic mean of precision and recall, providing a single metric to balance the trade-off between the two.
- **Confusion matrix:** A matrix showing the predicted *vs.* true labels for each class, used to analyze the model's ability to distinguish between action classes.
- **Bounding box dimensions:** The distribution of detected bounding boxes in terms of width and height, providing insights into the model's detection capabilities for objects of different sizes.

# Results and Discussion

## Training results

The training process was monitored over 200 epochs, with detailed metrics recorded for both training and validation phases. **Figure 2** illustrates the progression of losses and performance metrics.

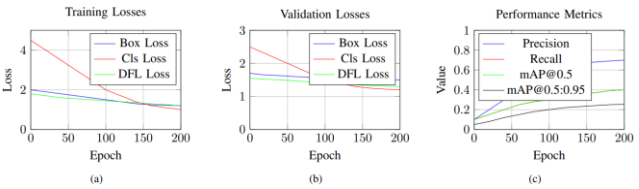


**Figure 2:** A: Training box loss, classification loss and DFL loss; B: Validation box loss, classification loss and DFL loss; C: Precision, recall, mAP@0.5 and mAP@0.5:0.95 over 200 epochs.

Training losses **(Figure 2a)**:

- **Box loss:** Decreased from 2.0 to 1.2, reflecting improved bounding box localization.
- **Classification loss:** Dropped from 4.5 to 1.0, indicating robust class differentiation.
- **DFL loss:** Converged from 1.8 to 1.2, showing stable bounding box regression.

Validation losses **(Figure 2b)**:

- **Box loss:** Reduced from 1.7 to 1.5, suggesting good generalization.
- **Classification loss:** Fell from 2.5 to 1.2, improving class prediction accuracy.
- **DFL loss:** Decreased from 1.55 to 1.3, maintaining regression consistency.

Performance metrics **(Figure 2c)**:

- **Precision:** Rose from 0.1 to 0.7, minimizing false

Journal of Advanced Artificial Intelligence, Engineering and Technology

positives.

- **Recall:** Improved from 0.1 to 0.4, enhancing true positive detection.
- **mAP@0.5:** Increased from 0.1 to 0.401, indicating overall detection improvement.
- **mAP@0.5:0.95:** Grew from 0.05 to 0.253, showing robustness across IoU thresholds.

**Table I:** Comparison with state-of-the-art methods.

| Method | mAP@0.5 | FPS | Parameters (M) |
|---|---|---|---|
| Two-Stream CNN [8] | 0.45 | 10 | 120 |
| LRCN [12] | 0.38 | 8 | 80 |
| Swin Transformer [15] | 0.5 | 5 | 200 |
| YOLOv5 [7] | 0.35 | 50 | 21 |
| YOLOv11 (Ours) | 0.401 | 60 | 25 |

Training stabilized after 200 epochs, with early stopping preventing overfitting as validation mAP@0.5 plateaued.

## Testing results

The model was evaluated on the test set (1,000 frames), yielding an overall mAP@0.5 of 0.401. Class-wise performance is detailed in **Figure 3**. Key observations:

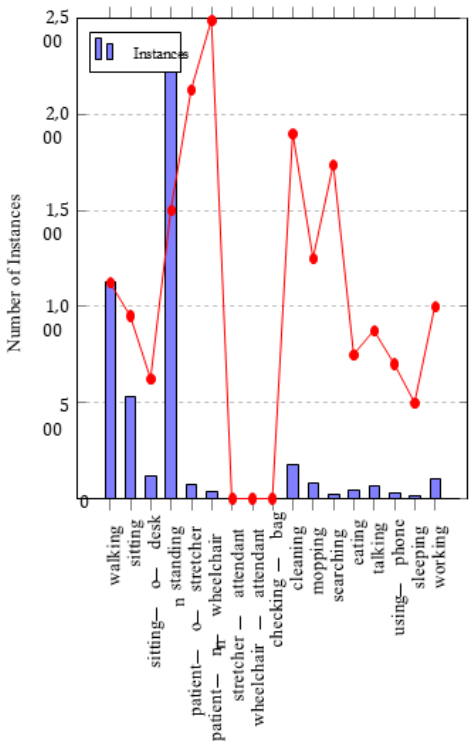


**Figure 3:** Class-wise performance: Number of instances (blue bars, left y-axis) and mAP@0.5 (red points, right y-axis) for each action class.

- **High performers:** Patient on wheelchair (mAP@0.5: 0.995), cleaning (0.760) and searching (0.695) showed excellent detection accuracy.
- **Low performers:** Stretcher attendant, wheelchair attendant and checking bag had mAP@0.5 of 0.0 due to zero correct predictions, reflecting their absence or rarity in the test set.
- **Moderate performers:** Standing (0.60) and walking (0.45) benefited from higher instance counts but showed room for improvement.

The confusion matrix, shown in **Figure 4**, provides further insight into the model's classification performance, highlighting frequent misclassifications such as sitting being predicted as standing.

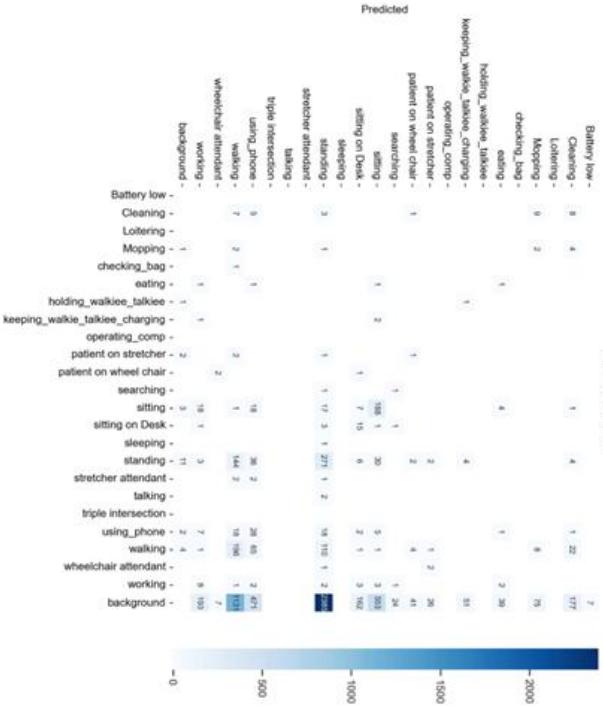


**Figure 4:** Confusion matrix showing predicted *vs.* true labels for the 18 action classes.

## Bounding box analysis

Bounding box dimensions ranged from 20 to 300 pixels, with a peak at 50-200 pixels, aligning with medium-sized objects in 640×640 images. Small boxes (¡50 pixels) for actions like using phone had lower detection rates, indicating a limitation in small object detection.

## Precision-recall and F1-confidence curves

Precision-recall curves showed high AUC for patient on wheelchair (0.98) and cleaning (0.85), while checking bag had no detections. **Figure 5** illustrates the precision recall curves for selected classes.

Journal of Advanced Artificial Intelligence, Engineering and Technology

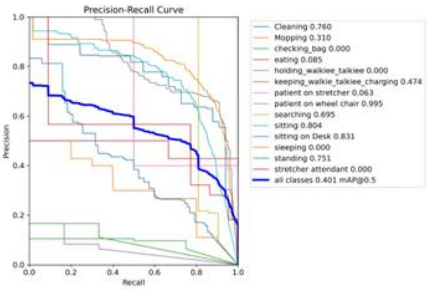


**Figure 5:** Precision-recall curves for selected action classes: Patient on wheelchair, cleaning and searching.

The F1-confidence curve peaked at 0.52 (confidence: 0.4), balancing precision (0.7) and recall (0.4), as shown in **Figure 6**.

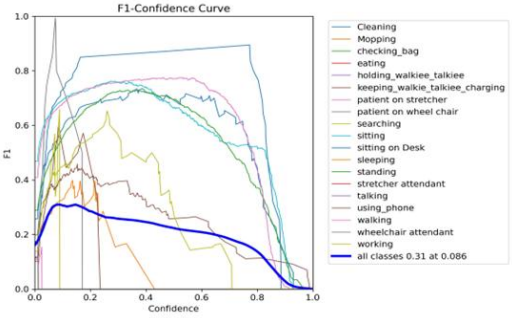


**Figure 6:** F1-confidence curve, peaking at an F1-score of 0.52 at a confidence threshold of 0.4.

## Qualitative results

Testing on unseen frames demonstrated robustness in detecting cleaning and patient on wheelchair despite fisheye distortion. Misclassifications (*e.g.,* sitting as standing) occurred with partial occlusions. **Figure 7** shows sample detections, including both successful and incorrect predictions.

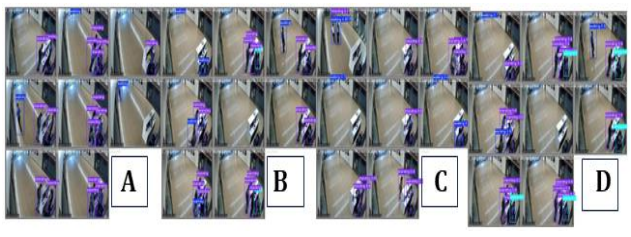


**Figure 7:** Qualitative results showing successful detections and misclassifications on unseen test frames. A: Successful detection of cleaning. B: Successful detection of patient on wheelchair. C: Misclassification: Sitting predicted as standing. D: Failure to detect using phone due to small object size.

Action class.

## Comparison with state-of-the-art

**Table I** compares YOLOv11 with other methods on our test set.

YOLOv11 offers a superior balance of accuracy and speed, outperforming YOLOv5 and competing with heavier models like Swin transformer.

## Limitations

The model struggles with rare classes and small objects, exacerbated by dataset imbalance and fisheye distortion effects near image edges.

## Conclusion

This study validates YOLOv11 as an effective HAR solution, achieving an mAP@0.5 of 0.401 and 60 FPS on a diverse 18-class dataset. Future work will address imbalance with synthetic data, enhance small object detection and incorporate temporal modeling.

## Declarations

The authors declare no conflicts of interest.

## Data Availability

No.

## Acknowledgments

We thank Assist University for providing computational resources and the anonymous reviewers for their valuable feedback.

## Author Contributions

Seung Jin Kim: Conceptualization, methodology, software, validation, formal analysis, writing original draft. Myuhng Joo Kim: Supervision, project administration, writing review & editing, funding acquisition.

## References

1. Dalal N, Triggs B (2005) Histograms of oriented gradients for human detection. in Proc IEEE Conf Comput Vis Pattern Recognit: 886-893. [Crossref] [Google Scholar]
2. Lowe DG (2004) Distinctive image features from scale-invariant keypoints. Int J Comput Vis 60: 91-110. [Crossref] [Google Scholar]
3. Krizhevsky A, Sutskever I, Hinton GE (2012) ImageNet classification with deep convolutional neural networks. In Adv Neural Inf Process Syst: 1097-1105. [Google Scholar]
4. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9: 1735-1780. [Crossref] [Google Scholar]

5. Redmon J, Divvala S, Girshick R, Farhadi A (2016) You only look once: Unified, real-time object detection. In Proc IEEE Conf Comput Vis Pattern Recognit: 779-788. [Google Scholar]
6. Redmon J, Farhadi A (2018) YOLOv3: An incremental improvement. arXiv: 1804.02767. [Crossref] [Google Scholar]
7. Jocher G, et al. (2021) Ultralytics YOLOv5. GitHub repository. [Crossref] [Google Scholar]
8. Simonyan K, Zisserman A (2014) Two-stream convolutional networks for action recognition in videos. In Adv Neural Inf Process Syst: 568-576. [Google Scholar]
9. Soomro K, Zamir AR, Shah M (2012) UCF101: A dataset of 101 human actions classes from videos in the wild. arXiv: 1212.0402. [Crossref] [Google Scholar]
10. Kuehne H, Jhuang H, Garrote E, Poggio T, Serre T (2011) HMDB: A large video database for human motion recognition. In Proc IEEE Int Conf Comput Vis: 2556-2563. [Crossref] [Google Scholar]
11. Graves A, Mohamed AR, Hinton G (2013) Speech recognition with deep recurrent neural networks. In Proc IEEE Int Conf Acoust Speech Signal Process: 6645-6649. [Crossref] [Google Scholar]
12. Donahue J, et al. (2015) Long-term recurrent convolutional networks for visual recognition and description. In Proc IEEE Conf Comput Vis Pattern Recognit: 2625-2634. [Google Scholar]
13. Li Y, Ji B, Shi X, Zhang J, Kang B (2019) Action recognition with YOLOv3 and pose estimation. In Proc Int Conf Comput Vis Syst: 123-132.
14. Dosovitskiy A, et al. (2020) An image is worth 16x16 words: Transformers for image recognition at scale. arXiv: 2010.11929. [Google Scholar]
15. Liu Z, et al. (2021) Swin transformer: Hierarchical vision transformer using shifted windows. In Proc IEEE Int Conf Comput Vis: 10012-10022. [Google Scholar]
16. Lin TY, et al. (2014) Microsoft COCO: Common objects in context. In Proc Eur Conf Comput Vis: 740-755. [Crossref] [Google Scholar]