scientific reports

Check for updates

OPEN

EgoVision a YOLO-ViT hybrid for robust egocentric object recognition

Umm e Sadima[1], Yazeed Alkhrijah[2], Danish Hamid[1], Muhammad Ehatisham Ul Haq[3], Syed Muhammad Usman[4], Shehzad Khalid[5,6]✉ & Mohamad A. Alawad[2]

The rapid advancement of egocentric vision has opened new frontiers in computer vision, particularly in assistive technologies, augmented reality, and human-computer interaction. Despite its potential, object recognition from first-person perspectives remains challenging due to factors such as occlusion, motion blur, and frequent viewpoint changes. This paper introduces EgoVision, a novel and lightweight hybrid deep learning framework that fuses the spatial precision of YOLOv8 with the global contextual reasoning of Vision Transformers (ViT). This research presents EgoVision, a whole new hybrid framework combining YOLOv8 with Vision Transformers for object classification in static egocentric frames. The static images come from the HOI4D dataset. To the best of our knowledge, this is the first time that a fused architecture is applied for static object recognition on HOI4D, specifically for real-time use in robotics and augmented reality applications. The framework employs a key-frame extraction strategy and a feature pyramid network to efficiently handle multiscale spatial-temporal features, significantly reducing computational overhead for real-time applications. Extensive experiments demonstrate that EgoVision outperforms existing models across multiple metrics, achieving up to 99% accuracy on complex object classes such as 'Kettle' and 'Chair', while maintaining high efficiency for deployment on wearable and edge devices. The results establish EgoVision as a robust foundation for next-generation egocentric AI systems.

Object recognition serves as a basic task in multiple domains, including robotics, Augmented Reality (AR), surveillance, and Human-Computer Interaction (HCI) systems that aim to precisely identify and categorize each object in the data into their respective classes. The development of egocentric vision[1], i.e., first-person view through wearable head-mounted cameras, introduces a better understanding of the surroundings for identification of objects as well as studying human-object interactions. Egocentric vision delivers optimal results in applications that need an authentic and comprehensive understanding of user surroundings, specifically assistive technologies and AR solutions. This paradigm shift in vision brings major complications to object recognition through extreme viewpoint changes alongside motion blur effects, object blockages, and background distractions[2].

Traditional methods for object detection often involve the use of Scale-Invariant Feature Transform (SIFT) and Histogram of Oriented Gradients (HOG)[3,4], which perform well in controlled environments; however, they face challenges due to real-world variations, including lighting changes, occlusions, and deformations. The object classification process using Support Vector Machines (SVMs) and *K*-Nearest Neighbors (KNN)[5,6] required classical machine learning techniques that faced scalability issues when working with large datasets. Probabilistic graphical models, including Hidden Markov Models (HMMs)[7], demonstrated improved tracking abilities for sequential object recognition in video data to enhance the robustness of systems. Multiple deep learning methods have also been introduced in recent years for real-time object recognition in egocentric vision, and they use CNN-based architectures, including Faster R-CNN and Single Shot MultiBox Detector (SSD), and YOLO (You Only Look Once)[8–10]. YOLOv3 and its variant YOLOv4 run fast enough for real-time processing while also being suited for wearable devices and augmented reality applications[11,12]. Long-range dependencies in

[1]Department of Creative Technologies, Faculty of Computing and Artificial Intelligence (FCAI), Air University, Islamabad 44000, Pakistan. [2]Department of Electrical Engineering, Imam Mohammad Ibn Saud Islamic University (IMSIU), Riyadh, Saudi Arabia. [3]School of Childhood and Social Care, University of East London, London E15 4LZ, UK. [4]Department of Computer Science, Bahria University, Islamabad 44000, Pakistan. [5]Department of Computer Engineering, Bahria University, Islamabad 44000, Pakistan. [6]Computer and Information Sciences Research Center (CISRC), Imam Mohammad ibn Saud Islamic University (IMSIU), 11623 Riyadh, Saudi Arabia. ✉email: Shehzad@bahria.edu.pk

images remain challenging for CNN-based models to detect, especially in complicated egocentric environments marked by frequent obstructions alongside excessive background elements and image blurring.

The emergence of Vision Transformers (ViTs)[13] delivers an alternative solution that utilizes self-attention mechanisms to obtain global contextual information. The Swin Transformer-based framework[14] introduced into object identification strengthens performance quality under diverse lighting situations and presents improved resistance to obscured objects. The Detection Transformer (DETR), among other Transformer-based models, demonstrates outstanding performance in dealing with challenging background environments. Natural motions of the head and body cause motion blurs and affect the performance of object recognition methods; therefore, motion-aware networks are required. Recurrent Neural Network (RNN)[15] alongside 3D CNNs serve as temporal modeling approaches that boost tracking and recognition abilities between multiple frames. Optical flow-based methods have been developed to strengthen resistance against motion distortions and boost the stability of systems in egocentric vision[16].

Hybrid deep learning methods have outperformed independent models in recent years. Integration of Graph Neural Network (GNN) with CNNs has also developed better spatial relationship comprehension, which leads to improved actual object recognition performance in egocentric video scenarios[17]. Through meta-learning techniques, researchers have developed methods that train models to adapt autonomously in different settings with limited labeled training information[18]. The current strategy applies semi-supervised learning methods to egocentric vision tasks, which decreases data requirements while applying unlabeled video data to learn features autonomously[19]. Few-shot learning methods have been studied to identify unknown objects in egocentric videos since they address data limitation problems[20]. While such architectures have achieved promising results in conventional computer vision benchmarks, their application to egocentric datasets remains largely unexplored. This presents a valuable research opportunity, particularly given the unique challenges posed by first-person perspectives, such as frequent occlusions, dynamic camera motion, and hand-object interactions.

Egocentric vision presents unique challenges for object recognition due to extreme viewpoint variations, frequent occlusions, motion blur, and dynamic hand-object interactions. Traditional CNN-based object detectors like YOLO or Faster R-CNN struggle with long-range dependencies and global context understanding, limiting their performance in complex first-person scenarios. Vision Transformers (ViT), although effective in capturing global context, are computationally intensive and less optimized for real-time applications. Moreover, previous works have largely focused on video-based modeling, leaving a significant gap in static image-based object recognition using egocentric datasets. The proposed EgoVision framework addresses these challenges by introducing a novel hybrid model that combines the fast detection capabilities of YOLOv8 with the global reasoning of ViT. It also fills the research gap by being the first to adapt the HOI4D dataset for static image-based object recognition. The incorporation of a key-frame selection strategy and a Feature Pyramid Network (FPN) further mitigates issues related to computational efficiency and multi-scale feature compatibility, making it suitable for real-time deployment on edge devices. The key contributions of this research are as follows:

- We propose EgoVision, a novel hybrid deep learning framework that combines the real-time detection efficiency of YOLOv8 with the global contextual learning capabilities of Vision Transformers (ViT), tailored specifically for egocentric object recognition.
- We are the first to utilize the HOI4D dataset for static image-based object recognition, extracting and annotating keyframes to establish a new benchmark in egocentric vision research.
- We introduce a key-frame selection strategy that captures pre-interaction, interaction, and post-interaction phases from video streams, significantly improving computational efficiency while retaining contextual relevance.
- We design a Feature Pyramid Network (FPN) to harmonize multi-scale spatial features across different convolutional layers, enhancing their compatibility for ViT-based representation learning.
- We demonstrate that EgoVision achieves state-of-the-art performance, with up to 99% accuracy for selected object classes, while remaining lightweight enough for edge and wearable devices, making it suitable for real-time applications in AR, robotics, and assistive systems.

Related work

Egocentric view-based object recognition has been an active research field that has helped improve the accuracy and efficiency of deep learning-based real-time intelligent systems over the past few years. Given the inherent challenges in first-person vision systems due to factors like motion blur, occlusion, changes in viewpoint, and background clutter, researchers have thoroughly investigated various object recognition algorithms.

The early methods that used Scale-Invariant Feature Transform (SIFT) and Histogram of Oriented Gradients (HOG)[3,4] succeeded in controlled environments yet faced difficulties with real-world variations, including lighting changes, occlusions, and deformations. The object classification process using Support Vector Machines (SVMs) and *K*-Nearest Neighbors (KNN)[5,6] required classical machine learning techniques that faced scalability issues when working with large datasets. Probabilistic graphical models, including Hidden Markov Models (HMMs)[7], demonstrated improved tracking abilities for sequential object recognition in video data to enhance the robustness of systems. The existing methodologies experienced limitations in both processing speed and limited capability to manage intricate egocentric interaction sequences effectively.

Deep learning technology revolutionized the process of recognizing and detecting objects[21–24]. Real-time object recognition in egocentric vision uses CNN-based architectures, which include Faster R-CNN and Single Shot MultiBox Detector (SSD), and YOLO (You Only Look Once)[8–10]. YOLOv3 and its variant YOLOv4 run fast enough for real-time processing while also being suited for wearable devices and augmented reality applications[11,12]. Long-range dependencies in images remain challenging for CNN-based models to detect,

especially in complicated egocentric environments marked by frequent obstructions alongside excessive background elements and image blurring.

The emergence of Vision Transformers (ViTs)[13] delivers an alternative solution that utilizes self-attention mechanisms to obtain global contextual information. The Swin Transformer-based framework[14] introduced into object identification strengthens performance quality under diverse lighting situations and presents improved resistance to obscured objects. The DETR (Detection Transformer), among other Transformer-based models, demonstrates outstanding performance in dealing with challenging background environments. Natural motions of the head and body cause motion blurs that harm object recognition performance, thus requiring motion-aware networks as a solution. Recurrent Neural Network (RNN)[15] alongside 3D CNNs serve as temporal modeling approaches that boost tracking and recognition abilities between multiple frames. Optical flow-based methods have been developed to strengthen resistance against motion distortions and boost the stability of systems in egocentric vision[16].

Research into hybrid modeling has increased intensely over the last years because individual architecture presents their constraints. Researchers have established that GNN integration with CNNs develops better spatial relationship comprehension, which leads to improved actual object recognition performance in egocentric video scenarios[17]. Through meta-learning techniques, researchers have developed methods that train models to adapt autonomously in different settings with limited labeled training information[18]. The current strategy applies semi-supervised learning methods to egocentric vision tasks, which decreases data requirements while applying unlabeled video data to learn features autonomously[19]. Few-shot learning methods have been studied to identify unknown objects in egocentric videos since they address data limitation problems[20]. While such architectures have achieved promising results in conventional computer vision benchmarks, their application to egocentric datasets remains largely unexplored. This presents a valuable research opportunity, particularly given the unique challenges posed by first-person perspectives, such as frequent occlusions, dynamic camera motion, and hand-object interactions.

Our proposed framework, known as EgoVision, uses a hybrid YOLO-ViT model to create a real-time recognition system specifically for egocentric vision. EgoVision combines YOLO's spatial efficiency with Transformers' contextual reasoning to effectively deal with occlusion and viewpoint variations, which makes it well-suited for robotic assistance and augmented reality while also serving smart wearable device requirements. The detection of real-time objects through egocentric vision enables crucial benefits in augmented reality platforms and assistive systems for enhancing digital environment usability. Deep learning models created specifically for egocentric vision help robotic devices grasp objects more effectively through manipulation tasks, and thus they become useful for both wearable smart devices along assistive intelligent technologies.

Although there has been an increased tendency toward research on video-based egocentric analysis, there still exists a much less explored domain of static image-based recognition using first-person data. Especially, no previous work has used static frames from the HOI4D dataset for real-time object classification. This gap in the literature becomes salient, for several applications, such as robotics, augmented reality overlays, or smart assistive systems, rely on instantaneous decisions upon the presentation of single frames deprived of any temporal information. Recent advancements in Granular Computing (GrC)[25–27] have introduced semantic-level information granules that enhance fine-grained feature extraction. These methods have shown notable potential in addressing visual similarity and ambiguity challenges common in egocentric object recognition tasks.

Against this background, we develop in this paper a hybrid YOLO-ViT model, called EgoVision, tailored specifically for egocentric object detection in static imagery derived from HOI4D. The novelty of this approach lies in joining local spatial detection with global contextual reasoning specifically for first-person view data, which has not been previously tested against static egocentric datasets.

Specialized datasets for first-person vision have driven considerably more progress in the field of egocentric object recognition. These datasets are distinct from the more common third-person datasets, which record objects from fixed viewpoints, and are critical to deployment in applications such as augmented reality, robotics, and assistive technologies, because they consist of dynamic interactions[28–33]. These gaps are addressed through frameworks like EgoVision, enabling robust and efficient real-time egocentric object recognition for applications in augmented reality, assistive technologies, and beyond. A comparative summary of relevant egocentric datasets is presented in Table 1, illustrating their diversity in terms of data modalities, activities, and scale.

Proposed methodology

We propose EgoVision: a novel framework designed to enhance object recognition in egocentric settings while maintaining real-time efficiency. The flow diagram of the proposed method is shown in Fig. 1, which visually outlines the complete pipeline from input to classification. This diagram serves as a roadmap for the method and illustrates how spatial and contextual features are extracted and fused for robust object recognition.

The proposed hybrid model is tailored precisely to meet the challenges of egocentric vision, including frequent occlusions, fast motion, and nonstandard viewpoints for objects. YOLOv8 provides fast local object detection in cluttered first-person scenes, and the Vision Transformer computes the global context and spatial relationships necessary for the disambiguation of objects. Hence, this combination secures robust recognition in dynamically changing and context-dependent egocentric environments. Unlike models that were merely tested on egocentric data, our architecture has been structurally designed for egocentric perceptions.

In the first step, key-frame extraction from egocentric video streams has been done to retain only the most contextually relevant frames capturing pre-interaction, interaction, and post-interaction phases. These frames are then preprocessed and manually annotated to generate high-quality training data using the HOI4D dataset. After data annotations, YOLOv8 is employed for local spatial feature extraction, leveraging its multi-scale convolutional backbone to detect fine-grained object characteristics. To harmonize these multi-resolution features, a Feature Pyramid Network (FPN) is applied, aligning them into a unified representation suitable for

Dataset	Year	Description
HOT3D	2024	HOT3D represents the standard dataset for monitoring egocentric vision-based three-dimensional interactions between hands and objects. The dataset combines multi-view image streams of RGB and monochrome video while 19 subjects work with 33 rigid objects.
IndustReal	2023	The dataset consists of 84 video clips, which demonstrate 27 subjects performing maintenance and assembly jobs with a construction-toy assembly set.
EvIs-Kitchen	2024	EvIs-Kitchen serves as a dataset that evaluates Video-Sensor-Sensor (V-S-S) interactions when detecting ego-HAR activities. The sequences of kitchen activities contain inertial sensors mounted on each wrist to track efficient human interaction for research purposes.
EgoExoLearn	2024	People record their demonstrated tasks following instructions from demonstration videos into this dataset. The database contains both egocentric video recordings and demonstration video recordings that extend across 120 hours.
IKEA Ego 3D	2024	A new ego-view 3D point cloud action data set exists for human action recognition. The dataset presents 493k frames along with 56 classes divided into four furniture categories.
EgoObjects	2023	To collect this dataset, 250 contributors uploaded 9k videos through their four wearable cameras in 50+ countries, which contained 368 categories.
EgoWholeBody	2023	The synthetic dataset contains 840K well-curated egocentric images which cover diverse motions of the whole body.
HOI-Ref	2024	HOI-Ref serves as a training platform that provides 3.9 million question-answer pairs to assess Visual Language Model performance. The dataset provides three types of questions that examine object recognition and hand-object movements, and environment understanding.

Table 1. Summary of Egocentric datasets and their characteristics.

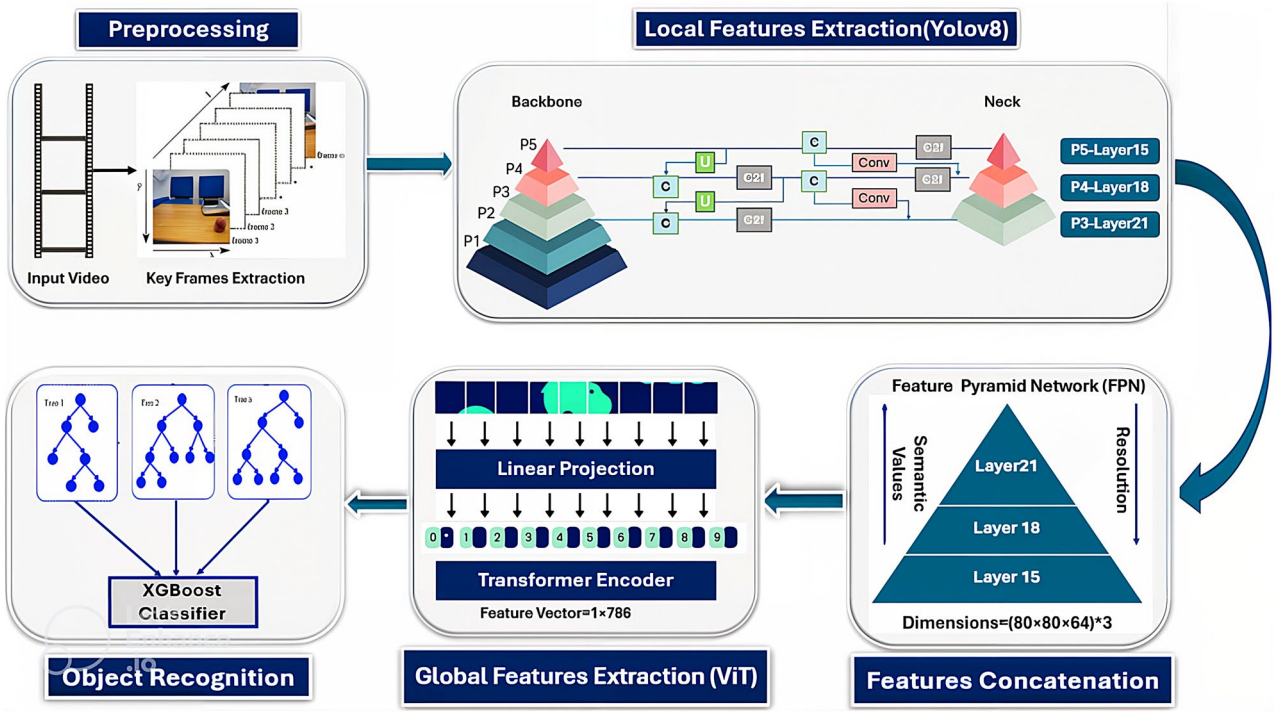


Fig. 1. EgoVision: Proposed framework for object recognition integrating YOLOv8 and ViT..

transformer input. This representation is then fed into a Vision Transformer (ViT) to model global contextual dependencies and improve recognition robustness in the presence of occlusion and motion blur. In the final step, a Random Forest classifier is used on the fused features from YOLO and ViT to perform the final object classification. This hybrid architecture ensures a balance between accuracy, generalization, and computational efficiency, making it ideal for real-time applications on edge and wearable devices. A detailed description of each step is presented in the following subsections.

Dataset

In this research, we have used the Human-Object Interaction 4D (HOI4D) dataset, which is very extensive and combines various object interaction possibilities and its high-definition egocentric video frame content. We select HOI4D, mainly because it provides context-rich first-person images during hand-object interactions. In contrast to generic video frames, HOI4D supplies semantically meaningful views fraught with realistic egocentric challenges, such as occlusion, varied angles, and motion. Such characteristics render HOI4D especially advantageous to train object recognition models under egocentric constraints, wherein object understanding must occur in very natural user-centric contexts. HOI4D also addresses key limitations observed in earlier egocentric datasets, such as low resolution, limited interaction diversity, minimal occlusion handling, and restricted environmental variety. With its high-resolution RGB-D frames, diverse interaction contexts, realistic occlusions, and coverage of 610 indoor environments, HOI4D combines the strengths of datasets like

EPIC-Kitchens and Ego4D while overcoming their flaws. This makes strong performance on HOI4D a reliable indicator of generalization to other egocentric datasets. Sample frames of the HOI4D dataset are shown in the Fig. 2.

The research team developed a basic data acquisition system comprising a bicycle helmet connected to a Kinect v2 RGB-D Sensor and an Intel RealSense D455 RGB-D Sensor to construct their data. Thus, the dataset contains enhanced 3D scene coverage from two integrated sensors. The dataset holds 2.4M RGB video frames that include 4000 videos at 1024 x 768 resolution,n created from 9 interacting participants performing 800 object instances across 610 indoor environments. The framework integrates a range of object classes that come with a selection and position functionality, followed by multiple function-based operations that serve for detecting object movement alongside functionality engagements. The complete number of classes reaches 16, which can be further sorted into two sections: rigid and articulated objects. Object classes from 0 to 15 consist of chairs, knives, toy cars, and scissors, alongside additional objects. As shown in Fig. 3, classes like Toy Car, Trashcan, and Mug have higher counts, while classes such as Safe and Chair appear less frequently. This distribution reflects the natural object occurrence patterns in egocentric settings.

Data preprocessing

Enhancement of HOI4D data through preprocessing represents an important step that allows better training effectiveness when utilizing the model. Data quality enhancement allows better identification capabilities for object recognition tasks by the model. Before training, no extensive preprocessing steps were conducted on the dataset. The model utilized raw RGB frames from the HOI4D video sequences as input. The frames were automatically resized to the expected model input size of 640 $\times$ 640 pixels, as handled internally by the Ultralytics YOLOv8 framework. The pixel values were normalized in the range [0, 1] according to default model requirements. Three video frames (as shown in Fig. 4) were selected from videos of 20 seconds duration for analyzing static images showing different human-object interaction periods. We collected the base frame during the period when the object was not touched by any person. The sequence in human-object interaction appears in the middle frame, and the final frame shows the object after human contact. A frame-selection protocol was standardized for all videos to minimize the chance of bias.

An extraction of keyframes uniformly across all video samples in the HOI4D dataset added great diversity to the dataset, presenting a wide range of subjects, scenes, object categories, and interaction types. This technique also alleviated the repetition problem and guaranteed the representative and balanced distribution of 6911 images over a variety of egocentric scenarios. This frame-static approach fits real-world needs in egocentric applications that feature wearable devices and assistive robotics, where instantaneous decisions have to be made on a frame-by-frame basis. By sampling key frames that define the interaction context, the model maintains critical temporal cues while avoiding the computational burden associated with processing entire videos. This further makes the current system scalable and efficient in the context of time on resource-starved devices. By integrating this input, the model receives extensive knowledge about the human-object relationship shown in video clips. After extracting the frames, we are left with images. The dataset consists of a total of 6,911 extracted images. Of these, 80% (5,529 images) were used for training, 10% (691 images) were allocated for testing, and the remaining 10% (691 images) were reserved for validation.

Fig. 2. Sample frames from HOI4D showing diverse egocentric human-object interactions.

Fig. 3. Distribution of samples per object category in the HOI4D.

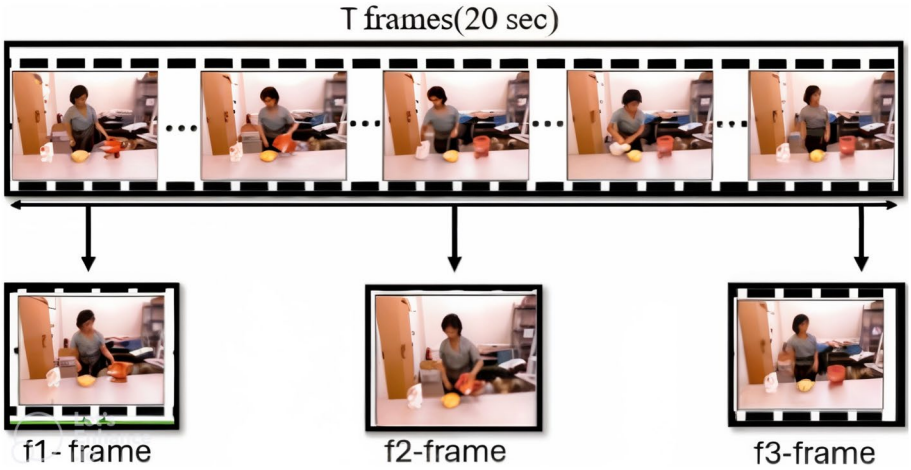
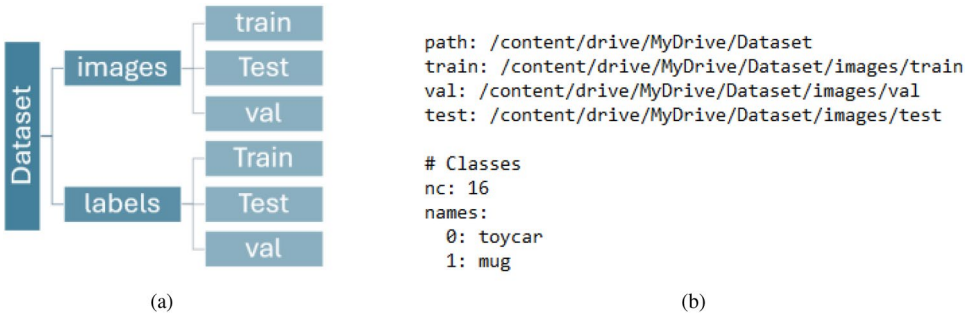


Fig. 4. Key-frame selection capturing pre-, during-, and post-interactions from input videos.

For our continuing examination, we selected the work of creating a YAML-formatted configuration file. The main role of this file involves determining where data will be processed according to the model understanding. The YAML file contains the paths to the root directory and also to training and testing, and validation data folders (as shown in Fig. 5). The dataset class labels have been described in two parts, where numbers represent them and textual identifiers explain their meaning. The correct identification of objects depends on applying numeric and name values to prevent misclassifications by the model.

Data annotation

The image labeling process for the proposed HOI4D dataset generated its video content using the Computer Vision Annotation Tool (CVAT)[34] during the object recognition problem. CVAT provides users with a convenient user interface to create annotations for images in an open-source application built for computer vision annotation needs. The annotation tool allows users to detect objects precisely and define their spatial boundaries while understanding how various objects connect to humans. The image annotation procedure involved detailed label marking of objects, which included both a unique identifier and specific class type assignment (as shown in Table 2). Such a labeling system allowed us to acquire essential details about object positions and their relationship patterns. Manual annotations were used as our method to prevent multiple errors that automation-



(a) (b)

Fig. 5. **a** Dataset distribution for YOLO fine-tuning, **b** YAML file configuration listing the dataset path and class labels.

Class Label	X Center	Y Center	Width	Height
3	0.335323	0.762954	0.445573	0.474093
3	0.743820	0.776282	0.423453	0.432583
9	0.453089	0.705120	0.103229	0.237833

Table 2. YOLO annotation format.

based annotation introduces and obtain superior robustness in data delivery. This improvement stands as an excellent addition to object recognition research, which enhances fundamental training data quality for machine algorithms. The precise annotations we developed during annotation creation now provide researchers studying object recognition with a strong foundational framework due to their close resemblance to real-world object recognition scenarios. The accurate data analysis method stands as a fundamental requirement for advancing object recognition work because our research adds value towards continuous field development.

Feature extraction

After preprocessing, both YOLO and ViT extracted specific features for our task that involved both local and global features. YOLOv8[35] served as the tool for extracting local features since it provides excellent real-time performance in application settings. We extracted the local features from the convolutional layers of the YOLOv8 backbone structure as follows:

$$y(x,y) = \iint_{-\infty}^{\infty} x(a,b) \cdot h(x-a, y-b)\, da\, db \tag{1}$$

$$\mathrm{SiLU}(x) = x \cdot \frac{1}{1+e^{-x}} \tag{2}$$

$$\mathrm{Bottleneck} = X + \mathrm{Conv}_{1\times1}(\mathrm{Conv}_{3\times3}(\mathrm{Conv}_{1\times1}(X))) \tag{3}$$

Where X is the input feature vector dimension. The layers executed spatial operations to detect features containing object structures like edges and textures. Feature propagation received improvement made possible by the C2f modules, which relied on split-transform-merge operations to maintain efficiency during extraction. Features of differing scales were extracted through Spatial Pyramid Pooling Fast(SPPF) modules by performing pooling operations at multiple kernel dimensions, so the model maintained adequate spatial data information as shown in the Fig. 6.

We extracted features from layers 15, 18, and 21 of the YOLOv8 model[35] features. The model organizes layers into a particular sequence so that it automatically obtains multiple details and abstraction levels. All feature dimensions change across different model layers when advancement occurs. The lower-level attributes of edges and textures appear in layer 15, but layer 18 starts recognizing middle-level patterns, and layer 21 identifies semantic features at the highest level. Table 3 shows the layer-wise dimensions of the extracted features. Cloud-based computational resources from Google Colab and Kaggle notebooks, including high-performance GPUs, managed the training process to achieve optimal efficiency. The model trained for 40 sessions used continuous weight adjustments to boost its functionality. Training stability was preserved through normalization of inputs and resizing images to 640×640 pixels and applying pixel intensity scaling from 0 to 1 before passing images to the feature extraction layers.

For the extraction of global features, we fed the features obtained from YOLO into the vision transformer. Since these layers have features of different dimensions and levels of abstractions, a Feature Pyramid Network (FPN) was required to build a consistent representation that can be fed as input to the transformer.

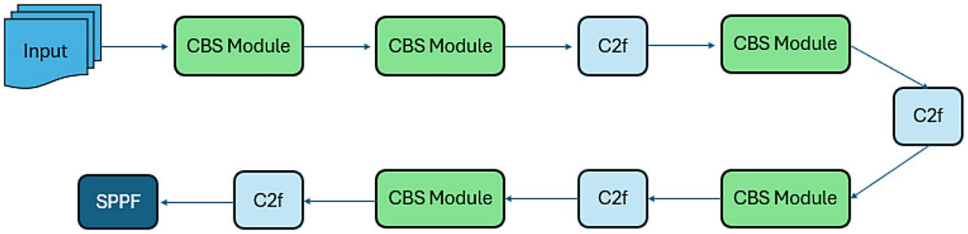


Fig. 6. Backbone architecture of YOLOv8 for multiscale feature extraction.

Layers	Feature Vector Size
Layer15	80×80×192
Layer18	40×40×384
Layer21	20×20×576

Table 3. Features dimensions at each Layer.

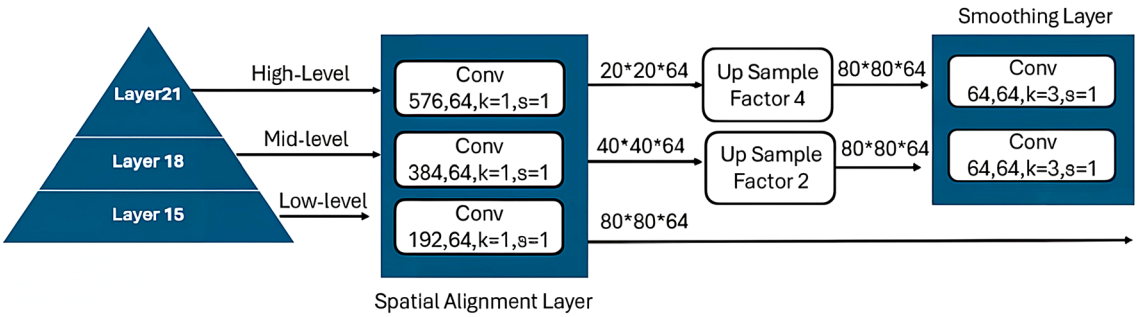


Fig. 7. Feature Pyramid Network(FPN) structure for aligning multidimensional YOLOv8 features.

Feature pyramid network

This approach describes the complete workflow that was used to scale both channel and spatial resolution features in the maps during the process. Figure 7 shows the complete mechanism of this process.

To extract hierarchical features at different levels of the backbone, a linear transformation coupled with a non-linear activation is applied to the output from specific layers. The channel dimensions of feature maps selected from layers 15, 18, and 21 in the YOLOv8 backbone are standardized for effective fusion of multilevel features in a manner suitable for subsequently merging them. This is done through 1×1 convolution on each feature map to lower each channel dimension to 64, as per Eqs. (4), (5), and (6):

$$X_{15} = \sigma\left(W_{15} \cdot X_{\text{input}} + b_{15}\right) \tag{4}$$

$$X_{18} = \sigma\left(W_{18} \cdot X_{\text{mid}} + b_{18}\right) \tag{5}$$

$$X_{21} = \sigma\left(W_{21} \cdot X_{\text{high}} + b_{21}\right) \tag{6}$$

Where:

$X_{\text{input}}, X_{\text{mid}}, X_{\text{high}}$ Input feature vectors from YOLOv8 layers 15, 18, and 21 respectivel

W_{15}, W_{18}, W_{21} Learnable weight matrices corresponding to layers 15, 18, and 2

b_{15}, b_{18}, b_{21} Bias terms associated with each respective laye

X_{15}, X_{18}, X_{21} Transformed feature representations obtained after applying the linear transformatio

σ Activation function (e.g., ReLU)

Convolution operation between the weight matrix and input feature map

The spatial alignments of feature maps took place before the process of up-scaling their spatial resolution. The sampling operations included the following sequence of actions for achieving this procedure: We sampled the feature map with a factor of 4 in this layer, thus achieving an initial layer enhancement. The bilinear interpolation technique was used to perform the up-sampling operations, which resulted in a smooth transitional connection between adjacent pixels. Another convolutional layer connected to the interpolation operation was employed to refine the sampled feature map. Adding this step allowed the spatial characteristics to enhance without

losing dimensional compatibility with upcoming model layers. The up-sampling parameter for the second stage remained set at 2.

To set the same spatial dimension for all feature maps (80*80), we performed upsampling on Layer 21 and Layer 18 only. This was because both of these layers had been reduced in size: Layer 21 was 20*20 and Layer 18 was 40*40. Thus, Layer 21 was upsampled by a factor of 4, and Layer 18 was augmented by a factor of 2. Layer 15 already had the expected spatial size of 80*80; therefore, it did not require further upsampling.

Equations (7) and (8) demonstrate this step by mathematical model. Bilinear interpolation was used to boost the spatial density of the feature map shortly before the convolution performed additional manipulations on the interpolated map. The hierarchical up-sampling methodology used in this network enables the system to detect elements at multiple scales and sizes.

$$X_{15\_\text{upsampled}} = U_4\left(X_{15}\right) \tag{7}$$

$$X_{18\_\text{upsampled}} = U_2\left(X_{18}\right) \tag{8}$$

where U_n represents upsampling by a factor of n.

Gaussian filters were utilized after up-sampling to reduce artifacts that potentially emerge during this process. The feature map quality requires Gaussian smoothing (as shown in Eqs. (9) and (10)) because the interpolation process can introduce noise along with abrupt edges. The selected features require this operation so they remain spatially accurate and relevant to the original work input.

$$S(x,y) = \sum_{i=-k}^{k}\sum_{j=-k}^{k} G(i,j)\cdot X(x+i,y+j) \tag{9}$$

Where:

$S(x, y)$ is the smoothed output at position (x, y)

$X(x+i, y+j)$ is the input pixel at offset (i, j)

$G(i, j)$ is the Gaussian kernel weight at offset (i, j)

σ controls the spread (standard deviation) of the Gaussian

k defines the kernel size as $(2k+1)\times(2k+1)$.

where the Gaussian kernel is defined as:

$$G(i,j) = \frac{1}{2\pi\sigma^2} e^{-\frac{i^2+j^2}{2\sigma^2}} \tag{10}$$

$G(i, j)$: The value of the Gaussian kernel at position (i, j)

σ: The standard deviation of the Gaussian distribution. It controls the spread or "width" of the kernel.

i, j: The pixel's coordinates (or displacement) from the center of the kernel in the horizontal and vertical directions.

$\frac{1}{2\pi\sigma^2}$: The normalization factor to ensure that the total sum of the kernel elements equals 1.

$e^{-\frac{i^2+j^2}{2\sigma^2}}$: The exponential part that gives higher weight to pixels near the center and lower weight to pixels farther away.

Due to previous operations, all processed feature maps underwent normalization until they reached a predefined pixel size of 80 by 8,0 with each map maintaining 64 channels. Spatial dimension, along with 'channel depth', must show high uniformity between pyramid levels for successful feature integration between levels 286 (as shown in Eq. (11)). The straightforward nature of post-processing becomes possible due to these operations, thus facilitating object detection and segmentation. This baseline does not include a complete ablation to isolate contributions from YOLO, ViT, and FPN completely. In broad strokes, YOLO should be thought of as providing precise local features, ViT as providing context in long ranges, and FPN as unifying multi-scale outputs.

$$X_{\text{final}} = \text{concat}_{\text{channel}}(X_{15}, X_{18}, X_{21}) \tag{11}$$

Feature embedding in ViT

Consequently, FPN's processed feature maps were applied in a way to feed them into ViT. In this feature map, the spatial dimension of the features is reduced to a standardized spatial dimension of 80×80 pixels such that the 64 channels operate via the channel way and the final representation is 80×80×192 (i.e., the 64 channels are summed up). By integrating with this, the Vision Transformer was able to work with a variety of multi-scale features without any modification. Therefore, since the Vision Transformer requires input, we divided these features into patches of size 16 x 16, which cover an input of 80 x 80. The four hundred patches from the feature map resulted from these extents of partitioning. Replicate it the same for all the patches, and resize their dimension to 256 dimensions, meaning 16×16×1 channel, which is 256 in total (as shown in Fig. 8).

For the visualization model to understand spatial information, it required a flattening process that transformed spatial data into an appropriate form for transformers. After flattening, the output vectors were inserted into the Vision Transformer. Self-attention (see Eq. (12)) within the ViT allows it to understand exact patch locations, which supports learning of local, along with global contexts. The issue of spatial relationships loss in work stems from transformer architecture usage, so each patch representation received positional embeddings to resolve this problem.

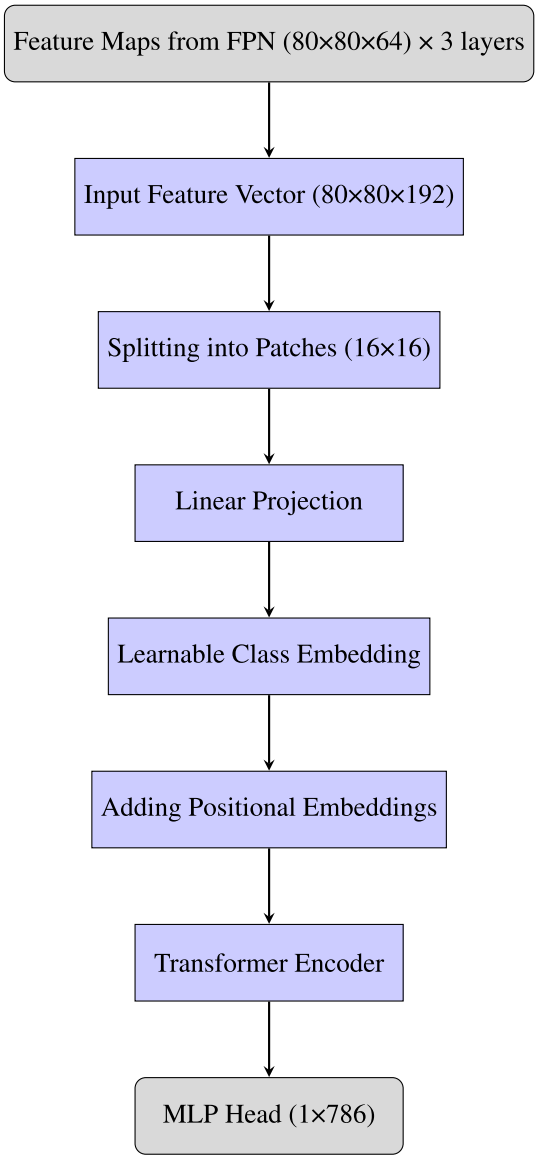


Fig. 8. Vision Transformer pipeline for global feature extraction via patch splitting, embedding, and encoding.

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \tag{12}$$

Where Q, K, and V are the query, key, and value matrices, respectively.

Transformer-driven representation learning with MLP head
After processing input patches through multiple transformer blocks with their self-attention layers and feedforward layers, the system directed the output toward the MLP head. All the features from the transformer layers feed into the MLP head. The final feature vector generated by the MLP head possesses a dimension of 1× 786 as its output vector. The output produces high-level features together with compact representations that assist subsequent model-based tasks with expanded contextual information. The MLP head serves instance segmentation in the classification task to produce a 1×786 feature vector applicable for classification and segmentation applications. The representation proves adequate to shorten the needed information coming from the inputs for subsequent model analysis. Through the union of FPN's multi-scale input collection mechanism and Vision Transformer's powerful representational abilities, this research improves considerably how the model interprets and recreates complex visual information, leading to better performance in downstream tasks.

Classification

An object recognition task was solved through a Random Forest classifier (as shown in Fig. 9 that extracted its features from the YOLO model combined with Vision Transformer outputs at the end of the pipeline.

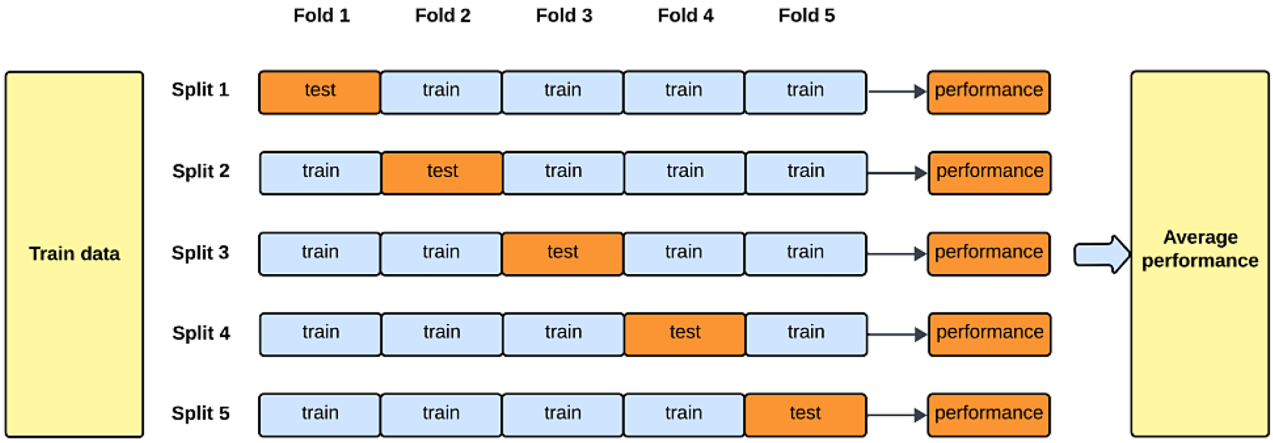


Fig. 9. 5-fold cross-validation for evaluating classification performance with Random Forest.

The Random Forest classifier was selected because of its internal capacity for coping proficiently with high-dimensional fused features and its maintainable performance on imbalanced and moderately-sized datasets like HOI4D. Its modular architecture allowed for robust classification without increasing computational complexity. Although not considered end-to-end trainable, the RF stands as a lightweight and interpretable solution for real-time egocentric applications. The input data description was effectively described by these features since an effective classification relies heavily on thorough description features. The received feature vectors needed their true labels attached to them so the training process could be performed. The classification model depends on these labels to obtain supervision while training because they ensure proper identification of input image contents. This research employs 5-fold cross-validation[36] to evaluate the Random Forest classifier's performance. Each model training step utilizes four segments of data to train its parameters while using the untrained fifth segment for testing. Each fold in this procedure functions as a test set once during its five repeated cycles. The final performance is calculated through the average of evaluation metrics measured across multiple iterations (as depicted in Eq. (13)), minimizing overfitting and guaranteeing unbiased results. The ensemble technique achieves higher accuracy while decreasing the problems associated with overlearning and thus performs well for multiclass outputs.

$$\text{Averaged 5-fold} = \frac{1}{k}\sum_{i=1}^{k}\text{Performance}^{(i)} \quad \text{where} \quad k=5 \tag{13}$$

Model parameters received evaluation through prediction versus truth comparisons, which resulted in error reduction throughout the training period. The classifier developed its prediction abilities through multiple iterations under the supervision of a training framework. Through training, the Random Forest classifier[37] becomes capable of detecting and categorizing unseen objects in fresh images by utilizing their feature representations. The combination of an efficient feature extraction module with the Random Forest classifier leads to improved recognition performance during the object recognition process. The system implements this method to establish the framework's efficiency at raising recognition accuracy and establishing strengthened robustness through generalizable scale features, which pair with ensemble learners based on Random Forest classifiers.

Experimental results and analysis

The EgoVision framework extracted features at both local and global scales for improved object detection capability. The YOLOv8 module extracted fine details from local regions while the Vision Transformer (ViT) processed complete image relationships to extract global characteristics. For classification purposes, Random Forest (RF) served as the processing framework, together with the scikit-learn software tool at 25 maximum depth. The training was performed in 100 epochs using the SGD optimizer on the YOLOv8m model variant, with a batch size of 16 and an input image resolution of 640 × 640. Those settings were chosen based on defaults considered best practices in the YOLOv8 implementation by Ultralytics. The conducted experiments utilized a Dell laptop featuring an Intel(R) Core i7-7500U processor operating at 2.70 GHz and reaching up to 2.90 GHz through turbo boost mode and containing 8 GB of RAM while running Windows 10 Pro Version 22 Hz OS Build 19045.5131. A Python framework included OpenCV along with PyTorch and scikit-learn libraries for executing the implementation. We evaluate the proposed method by using several visualization tools, including confusion matrices, performance curves, and additional suitable graphical representations. These visualizations present a quantitative and clear analysis showing the model's accuracy levels and performance alongside its error distribution. Table 4 provides detailed class-wise evaluation metrics including accuracy, precision, recall, and F1-score to support analysis of model robustness across all object categories. The reduced performance on classes such as toy car, mug, and trashcan can be attributed to class imbalance, as these categories had limited

Class	Accuracy	Precision	Recall	F1-Score	TPs	FNs	FPs	TNs
Bottle	0.93	0.96	0.93	0.95	71	5	3	1137
Bowl	0.97	1.00	0.97	0.99	74	2	0	1140
Bucket	0.72	0.89	0.72	0.80	55	21	7	1133
Chair	0.99	0.84	0.99	0.91	75	1	14	1126
Kettle	0.99	0.85	0.99	0.91	75	1	13	1127
Knife	0.93	0.83	0.93	0.88	71	5	15	1125
Lamp	0.92	0.88	0.92	0.90	70	6	10	1130
Laptop	0.95	0.69	0.95	0.80	72	4	32	1108
Mug	0.47	0.57	0.47	0.52	36	40	27	1113
Pliers	0.97	0.93	0.97	0.95	74	2	6	1134
Safe	0.99	0.69	0.99	0.82	75	1	33	1107
Scissor	0.76	0.89	0.76	0.82	58	18	7	1133
Stapler	0.96	0.86	0.96	0.91	73	3	12	1128
Furniture	0.95	0.76	0.95	0.84	72	4	23	1117
Toy Car	0.22	0.39	0.22	0.28	17	59	27	1113
Trashcan	0.25	1.00	0.25	0.40	19	57	0	1140

Table 4. Average Classification Metrics for 5-Fold Cross-Validation.

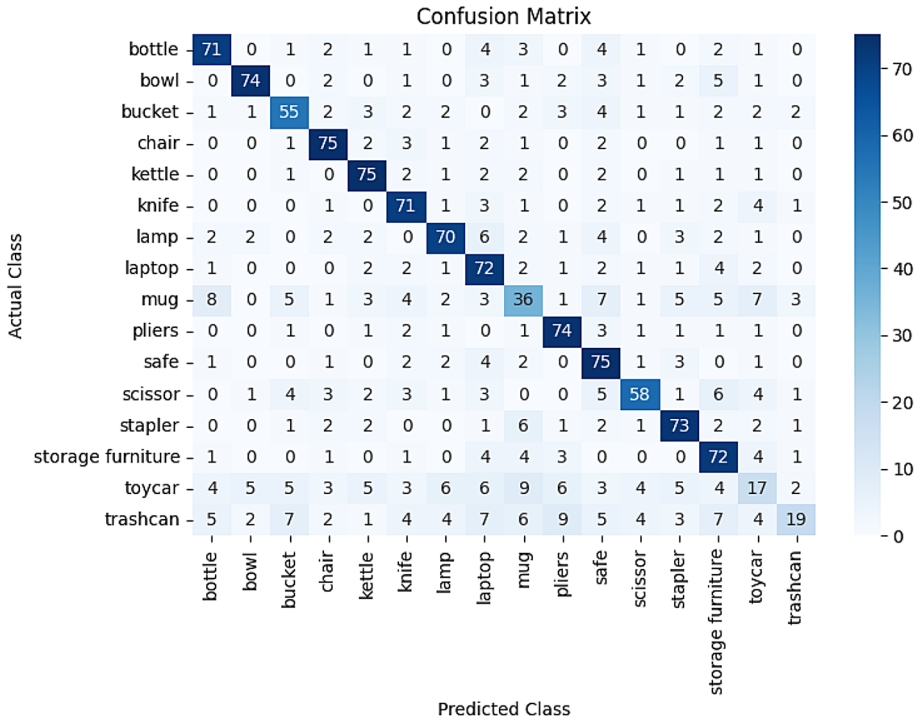


Fig. 10. Confusion matrix for object recognition performance.

representation in the training set. While general augmentations were applied, no class-specific augmentation or balancing strategies were used. In the future, work may involve targeted augmentation, such as geometric transforms, to improve recognition of the minority classes, including Toy Car and Trashcan.

The proposed hybrid architecture effectively addresses inter-class similarity and occlusion–two common challenges in egocentric vision–by combining the fine-grained spatial precision of YOLOv8 with the global contextual reasoning of Vision Transformers (ViT). YOLOv8 enables precise localization of object boundaries, while ViT captures surrounding context, aiding in the recognition of partially occluded or visually similar objects. This design helps reduce confusion between overlapping categories, as reflected in the class-wise metrics and confusion matrix. Additionally, the key-frame selection strategy further improves robustness by providing diverse visual cues across different interaction phases. The confusion matrix presented in Fig. 10 provides an in-depth analysis of how well the model performs classifications. Most values are positioned on the diagonal section, which demonstrates that the model is successful at identifying most instances in their proper categories.

The distribution of values along the main diagonal demonstrates how well the classifier performs at recognizing objects accurately.

Certain off-diagonal elements demonstrate substantial classification errors by the model. Bottles are frequently mistaken for trash cans or mugs by the classifier because of their shared shape and texture characteristics. The misclassifications among "scissors" with "knife" and "pliers" are influenced by shared structural similarity between these objects. Improvements to particular class detection will become possible through better methods for recognizing distinct features in those categories. The model shows exceptional performance in identifying the "bucket" and "trashcan" object classes because these classes demonstrate very high correct recognition rates. The confusing separation of visually comparable object pairs, such as knives and scissors, needs higher-quality data augmentation along with advanced representation learning to improve model classification precision. The accuracy visualization, as shown in Fig. 11, exhibits class-specific detection performance variations. Some object classes show reduced accuracy levels, although most classes reach high detection accuracy rates. This suggests possible difficulties in recognizing objects with similar visual characteristics. The model demonstrates general proficiency yet faces specific challenges when detecting certain object types accurately.

Some classes show lower accuracy performance because the dataset contains uneven representations of these objects, and the used features lack sufficient distinction ability. The combination of more diverse datasets and enhanced feature techniques would work together to solve performance problems in categories with low accuracy. The model demonstrates outstanding retrieval capability when identifying objects which include "bowl," "chair," "kettle," "pliers," "safe," and "stapler." The model demonstrates reduced precision in identifying the categories mug, toy car, and trashcan, which signifies problems during their classification processes. The execution proves to be successful in general, but some objects show persistent difficulties in classification. Figure 12 shows object-wise F1-score measurements, which condense information about the model's precision-recall balance across each classification type. Higher F1-scores for particular objects demonstrate the classifier's ability to achieve optimal false-positive and negative results for reliable identification. The F1-scores of particular objects indicate either high rates of incorrect positives or incorrect negatives. It shows that objects with clear, distinct features achieve better F1-scores. The performance of objects decreases when they exhibit shared visual characteristics with alternative classification categories. Figure 13 shows Precision-Recall (PR), which serves as a critical analysis tool for evaluating model performance across varying classification thresholds. The curve illustrates the changing precision levels as the recall value rises, thus providing knowledge about these dual metrics. An effective model should maintain high precision standards during increases in recall because it enables detection capability without yielding false positives.

The PR curve indicates that model performance shows exceptional precision-recall balance across most object categories because it attains a high area under the curve value. All classes reach a measurement of 1.00 area under the curve (AUC), showing the best precision as well as recall levels. The model demonstrates strong capability in differentiating objects without any considerable misclassification errors. The model demonstrates precise classification performance because the recall value approaches 1.0 while the curve remains vertical. The model demonstrates excellence in reducing incorrect classifications, which results in superior performance in all categories. Figure 14 shows the key measurement tool to evaluate how well the classifier identifies positive instances from negative ones. Model performance quality rises with the area under the ROC curve (AUC), reaching higher values and approaching perfection when AUC approaches 1.0. Most object classes achieve high AUC values within the ROC curve analysis, which proves the model's effectiveness for separating different object categories from one another.

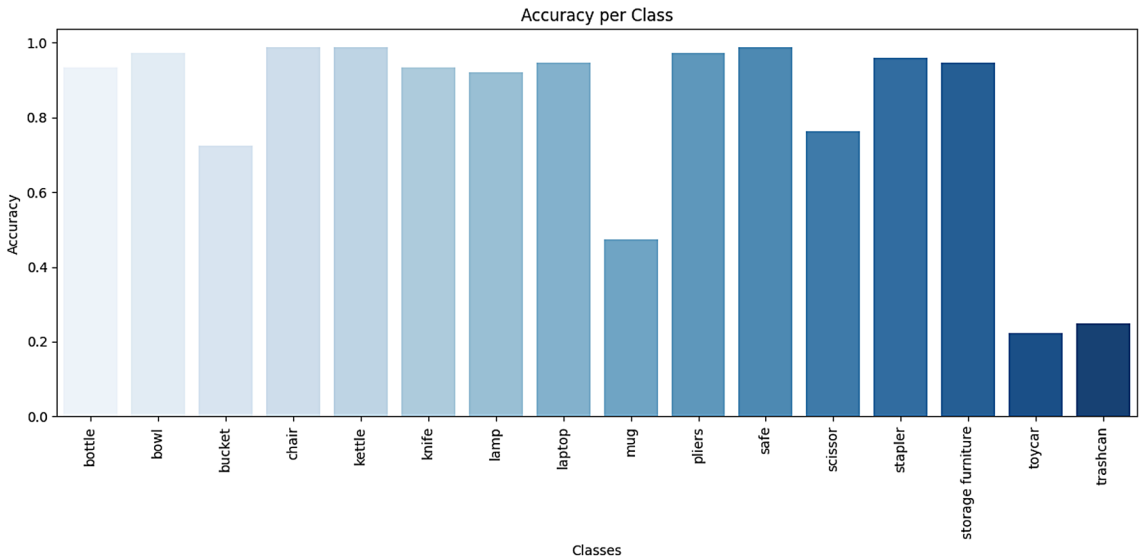


Fig. 11. Bar Plot Representation for class-wise Recognition Accuracy.

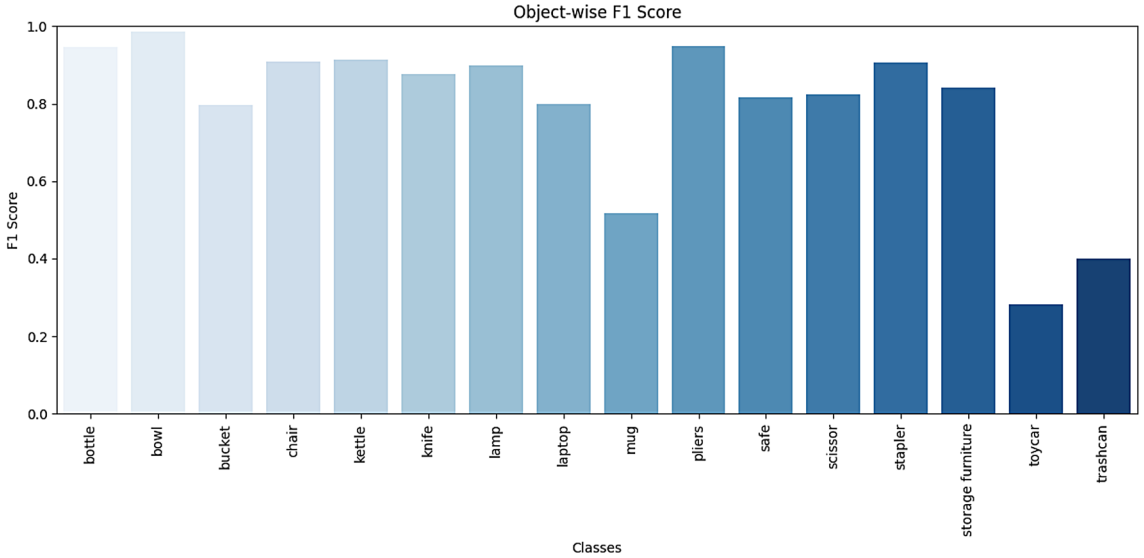


Fig. 12. Object-Wise F1-Score for Recognition Task.

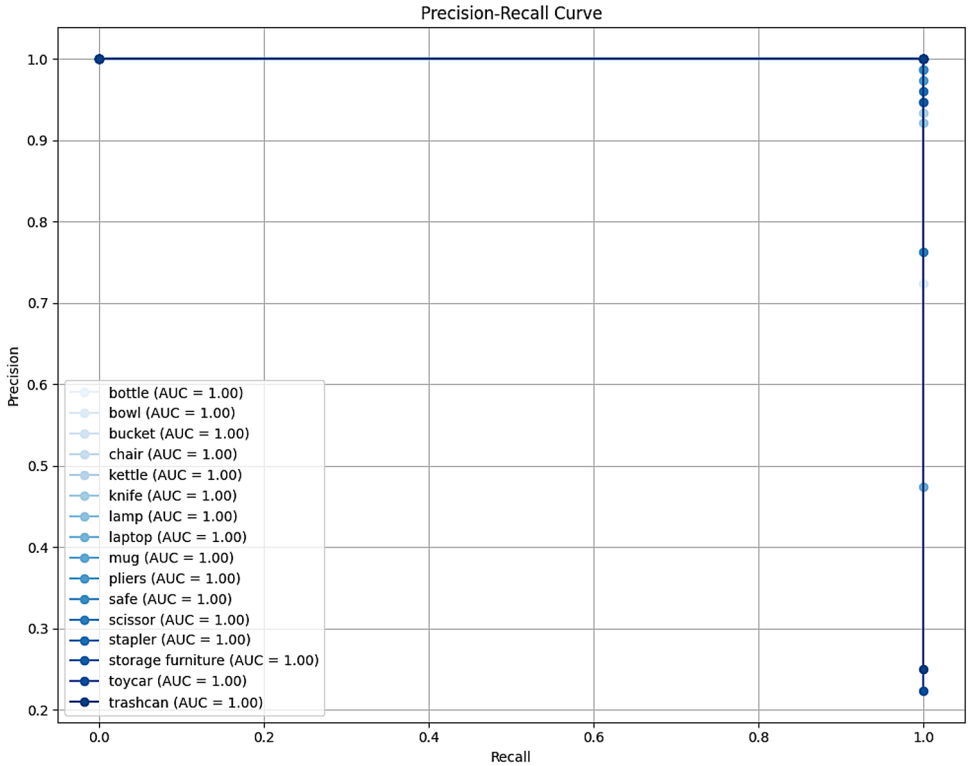


Fig. 13. Precision-Recall Curve: Evaluating Model Performance Across Different Thresholds.

Figure 15 displays the Mean Average Precision (mAP) bar plot as a comprehensive precision measurement for multiple recall thresholds. A model with higher mAP detects objects more accurately and reliably under different operational environments. The model demonstrates high detection precision rates for most detected objects according to mAP analysis results, which validates its overall operational effectiveness. The mAP scores fall at different levels for specific object classes, suggesting their detection confidence behaves differently based on the conditions. The experiments in this study utilized the HOI4D dataset, which was initially designed for human-object interaction tasks within a video-based context. Unlike previous works, we focused on static image-based object recognition, a novel application of the dataset. To the best of our knowledge, no prior research has explored the use of the HOI4D dataset for static image-based object recognition. In this work, we extracted individual frames from the video sequences and applied object recognition techniques to these images.

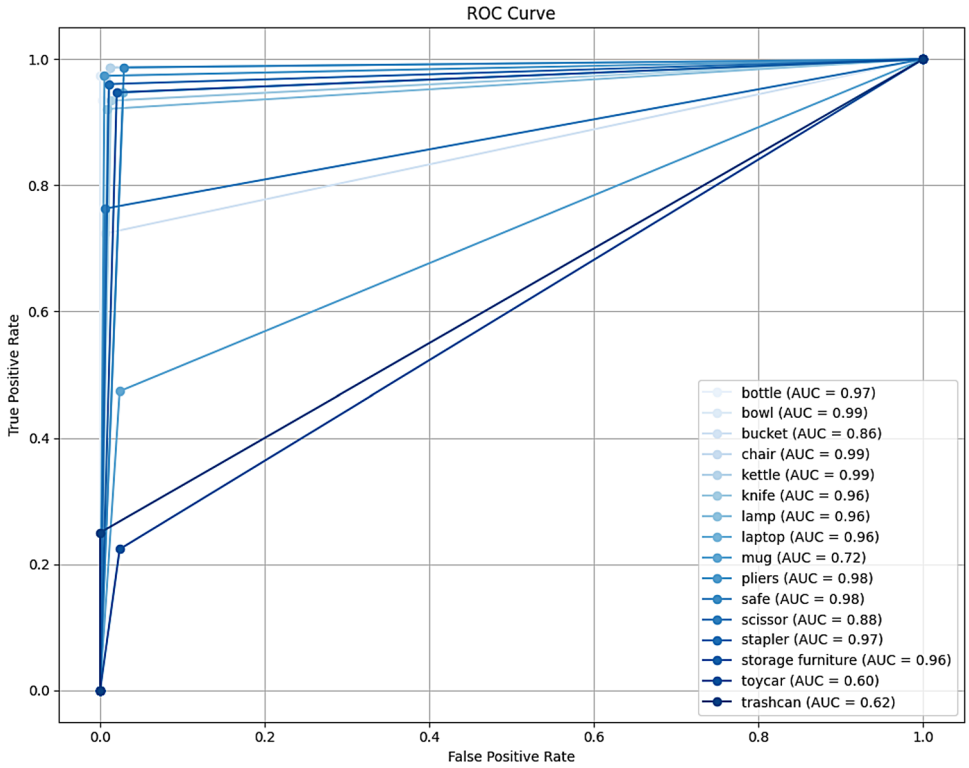


Fig. 14. ROC Curve: Evaluating Trade-off Between True Positive Rate and False Positive Rate.

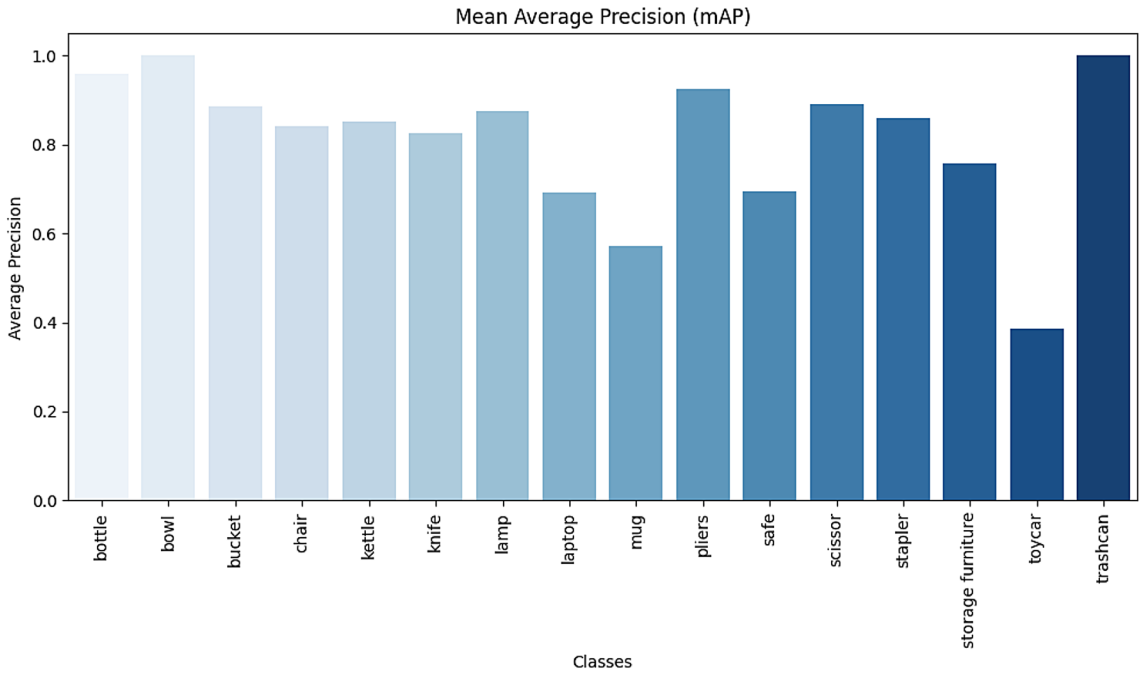


Fig. 15. Mean-Average Precision (mAP) Plot illustrating recognition accuracy across all object classes.

As there are no comparable methods in the existing literature, a direct quantitative comparison with previous studies is not possible. While direct retraining of YOLOv8-only, ViT-only, or Swin-only baselines on HOI4D was not conducted, the hybrid design leverages YOLO's precise localization and ViT's global context modeling, addressing the individual limitations of each and improving recognition in occlusion-heavy egocentric views. While most of the hybrid YOLO-ViT and transformer-type object detectors are performing better than third-

person datasets such as COCO and Pascal VOC, static egocentric HOI4D is a different benchmark with very contrasting viewpoint variation, occlusion frequency, and interaction context. Their results cannot be compared directly without retraining on HOI4D, which remains a future research direction.

Conclusion

Egocentric object recognition faces multiple challenges due to occlusion, motion blur, dynamic viewpoints, and background clutter inherent in first-person perspectives. Traditional CNN-based models often struggle to capture long-range dependencies, while transformer-based architectures, although effective in global feature representations, demand high computational resources and are not optimized for real-time applications. To address these challenges, we propose EgoVision, a novel hybrid framework that integrates the spatial precision of YOLOv8 with the global contextual reasoning of Vision Transformers (ViT). The novelty of the proposed method lies in its pioneering use of the HOI4D dataset for static image-based object recognition, a domain previously unexplored, thus establishing a new benchmark in egocentric vision research. This work addresses a critical gap by introducing a static image-based object detection framework for HOI4D. EgoVision offers a lightweight solution for real-time egocentric applications, demonstrating strong performance without relying on temporal information. Furthermore, the incorporation of a key-frame extraction strategy, Feature Pyramid Network (FPN) for multi-scale feature harmonization, and Random Forest classifier for lightweight, accurate classification makes EgoVision both innovative and highly practical for deployment on resource-constrained devices such as wearables and AR systems. The proposed method achieved up to 99% classification accuracy on complex object categories while maintaining high efficiency, demonstrating its significance for real-time applications in assistive technologies, robotics, and human-computer interaction. Despite strong performance, the model exhibited misclassifications in visually similar object classes like mugs and toy cars, pointing to the need for better representation learning and data balancing. EgoVision provides a scaled-up and much faster alternative to conventional video-based pipelines by addressing this underexplored setting with a special fusing architecture for static egocentric vision. Future work will focus on extending the framework to handle sequential data using temporal attention mechanisms and integrating data augmentation techniques to improve robustness across underrepresented classes. Moreover, adapting EgoVision for low-light and outdoor environments and exploring its integration with multimodal sensors could further enhance its applicability in real-world egocentric systems.

This model is intended to perform particularly well on regular hardware. This work, however, has not looked into the deployment of the model in dedicated edge devices. In benchmarking FPS, latency, and memory usage on edge systems (e.g., Jetson Nano, Raspberry Pi), possible future work could be a live test on wearable or mobile systems to prove the efficiency in real-world modeling. YOLOv8m, while optimized for real-time inference, will, however, feature future iterations benchmarking performance on embedded platforms such as Jetson Nano or Raspberry Pi to understand its applicability in real-world resource-constrained environments. At the moment, the implemented framework only allows for inference on static video frames for classification tasks. The model architecture, by nature, is capable of taking stream-based inputs; thus, future work aims at real-time deployment for continuous video streams for live egocentric vision applications such as robotic perception and wearable AI systems.

This study assumes that static egocentric frames contain sufficient visual context for reliable object classification, even without temporal continuity. While the proposed model achieves high accuracy in several object categories, its performance drops on visually similar classes due to feature ambiguity. Real-world deployment challenges–such as varying lighting, motion artifacts, or sensor noise–were not addressed in this version. Furthermore, the current implementation does not include real-time stream processing or edge-device deployment, which are planned for future work.

Data availability

The datasets generated and/or analyzed during this study are available at https://hoi4d.github.io/

Received: 20 May 2025; Accepted: 1 September 2025
Published online: 06 October 2025

References

1. Nunez-Marcos, A. Azkune, G. & Arganda-Carreras, I. Egocentric vision-based action recognition: A survey. *Neurocomputing* **472**, 175–197. https://doi.org/10.1016/j.neucom.2021.11.081 (2022).
2. Li, X. *et al.* Challenges and trends in egocentric vision: A survey. *arXiv preprint* arXiv:2503.15275 (2025).
3. Lowe, D. G. Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vis.* **60**, 91–110 (2004).
4. Dalal, N. & Triggs, B. Histograms of oriented gradients for human detection. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1, 886–893. https://doi.org/10.1109/CVPR.2005.177 (2005).
5. Bishop, C. M. Pattern recognition and machine learning. In *Inf. Sci. Stat.* 1st edn (Springer, New York, NY, 2006).
6. Cover, T. & Hart, P. Nearest neighbor pattern classification. *IEEE Trans. Inf. theory* **13**, 21–27 (1967).
7. Hornegger, J., Niemann, H., Paulus, D. & Schlottke, G. Object recognition using hidden Markov models. In *Machine Intelligence and Pattern Recognition* Vol. 16, 37–44 (Elsevier, NY, 1994).
8. Jiang, D. et al. Semantic segmentation for multiscale target based on object recognition using the improved faster-rcnn model. *Future Gener. Comput. Syst.* **123**, 94–104 (2021).
9. Lyu, Z., Jin, H., Zhen, T., Sun, F. & Xu, H. Small object recognition algorithm of grain pests based on ssd feature fusion. *IEEE Access* **9**, 43202–43213 (2021).
10. Francies, M. L., Ata, M. M. & Mohamed, M. A. A robust multiclass 3d object recognition based on modern yolo deep learning algorithms. *Concurr. Comput.: Pract. Exp.* **34**, e6517 (2022).
11. Kai, T., Lu, H. & Kamiya, T. Object recognition from spherical camera images based on yolov3. In *2020 20th International Conference on Control, Automation and Systems (ICCAS)*, 419–422 (IEEE, 2020).

12. Dewi, C., Chen, R.-C., Jiang, X. & Yu, H. Deep convolutional neural network for enhancing traffic sign recognition developed on yolo v4. *Multimed. Tools Appl.* **81**, 37821–37845 (2022).
13. Dosovitskiy, A. *et al.* An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint* arXiv:2010.11929 (2020).
14. Liu, Z. *et al.* Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF international conference on computer vision*, 10012–10022 (2021).
15. Liang, M. & Hu, X. Recurrent convolutional neural network for object recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 3367–3375 (2015).
16. Watanabe, M., Takeda, N. & Onoguchi, K. A moving object recognition method by optical flow analysis. In *Proceedings of 13th International Conference on Pattern Recognition*, vol. 1, 528–533 (IEEE, 1996).
17. Escorcia, V., Guerrero, R., Zhu, X. & Martinez, B. Sos! self-supervised learning over sets of handled objects in egocentric action recognition. In *European Conference on Computer Vision*, 604–620 (Springer, 2022).
18. Hatano, M., Hachiuma, R., Fujii, R. & Saito, H. Multimodal cross-domain few-shot learning for egocentric action recognition. In *European Conference on Computer Vision*, 182–199 (Springer, 2024).
19. Wang, H. et al. Distilling interaction knowledge for semi-supervised egocentric action recognition. *Pattern Recognit.* **157**, 110927 (2025).
20. Majumder, S., Nagarajan, T., Al-Halah, Z. & Grauman, K. Switch-a-view: Few-shot view selection learned from edited videos. *arXiv preprint* arXiv:2412.18386 (2024).
21. Amin, S. U., Hussain, A., Kim, B. & Seo, S. Deep learning based active learning technique for data annotation and improve the overall performance of classification models. *Expert Syst. Appl.* **228**, 120391 (2023).
22. Ul Amin, S., Kim, B., Jung, Y., Seo, S. & Park, S. Video anomaly detection utilizing efficient spatiotemporal feature fusion with 3d convolutions and long short-term memory modules. *Adv. Intell. Syst.* **6**, 2300706 (2024).
23. Khan, H., Usman, M. T. & Koo, J. Bilateral feature fusion with hexagonal attention for robust saliency detection under uncertain environments. *Inf. Fusion* **121**, 103165 (2025).
24. Khan, H., Khan, S. U., Ullah, W. & Baik, S. W. Optimal features driven hybrid attention network for effective video summarization. *Eng. Appl. Artif. Intell.* **158**, 111211 (2025).
25. Ghosh, S., Paral, P., Chatterjee, A. & Munshi, S. Rough entropy-based fused granular features in 2-d locality preserving projections for high-dimensional vision sensor data. *IEEE Sens. J.* **23**, 18374–18383 (2023).
26. Ghosh, S., Paral, P., Chatterjee, A. & Munshi, S. A novel 2d robust lpp-based approach using density-based neighborhood granulation for challenging visual cue detection. *IEEE Sens. J.* **25**(5), 8665–8673 (2025).
27. Paral, P., Ghosh, S., Pal, S. K. & Chatterjee, A. Adaptive non-homogeneous granulation-aided density-based deep feature clustering for far infrared sign language images. *IEEE Trans. Emerg. Top. Comput. Intell.* **9**(2), 1269–1280 (2024).
28. Banerjee, P. *et al.* Introducing hot3d: An egocentric dataset for 3d hand and object tracking. *arXiv preprint* arXiv:2406.09598 (2024).
29. Schoonbeek, T. J., Houben, T., Onvlee, H., Van der Sommen, F. *et al.* Industreal: A dataset for procedure step recognition handling execution errors in egocentric videos in an industrial-like setting. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 4365–4374 (2024).
30. Huang, Y. *et al.* Egoexolearn: A dataset for bridging asynchronous ego-and exo-centric view of procedural activities in real world. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 22072–22086 (2024).
31. Ben-Shabat, Y., Paul, J., Segev, E., Shrout, O. & Gould, S. Ikea ego 3d dataset: Understanding furniture assembly actions from ego-view 3d point clouds. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, 4355–4364 (2024).
32. Zhu, C. *et al.* Egoobjects: A large-scale egocentric dataset for fine-grained object understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 20110–20120 (2023).
33. Bansal, S., Wray, M. & Damen, D. Hoi-ref: Hand-object interaction referral in egocentric vision. *arXiv preprint* arXiv:2404.09933 (2024).
34. CVAT.ai. Computer vision annotation tool (accessed 08 April 2025) (2023).
35. Yaseen, M. What is yolov8: An in-depth exploration of the internal features of the next-generation object detector. arxiv 2024. *arXiv preprint* arXiv:2408.15857.
36. Wong, T.-T. & Yeh, P.-Y. Reliable accuracy estimates from k-fold cross validation. *IEEE Trans. Knowl. Data Eng.* **32**, 1586–1594 (2019).
37. Rigatti, S. J. Random forest. *J. Insur. Med.* **47**, 31–39 (2017).

Author contributions

Conceptualization, U.S., Y.A., and S.M.U.; methodology, U.S., Y.A., D.H., M.E.H., and S.M.U.; software, U.S., Y.A., D.H., and M.E.H.; validation, S.M.U., S.K., and M.A.A.; formal analysis, D.H., S.K., and S.M.U.; investigation, U.S., Y.A., D.H., and M.E.H.; resources, S.M.U. and S.K.; data curation, U.S., D.H., Y.A., and M.E.H.; writing–original draft preparation, U.S., Y.A., and S.M.U.; writing–review and editing, S.K., S.M.U., and M.A.A.; visualization, U.S., D.H., and S.K.; supervision, S.M.U. and S.K.; project administration, S.M.U. and M.A.A.

Funding

This work was supported and funded by the Deanship of Scientific Research at Imam Mohammad Ibn Saud Islamic University (IMSIU) (grant number IMSIU-DDRSP2503).

Declarations

Competing interests

The authors declare no competing interests.

Additional information

Correspondence and requests for materials should be addressed to S.K.

Reprints and permissions information is available at www.nature.com/reprints.

Publisher's note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.