Dijlah Journal of Engineering Science (DJES) Vol. 3, No. 3, August, 2026, pp. 213-221
ISSN: Printed: **3078-9656, Online: 3078-9664**, paper ID: 198

Semantic Analysis of Moving Objects in Video: A Comprehensive Framework for Motion Understanding and Event Detection

Emad Mahmoud Ibrahim
Department of Computer Science, Diyala Governorate Office, Diyala, Iraq

Article Info

Article history:

Received Apr.,28, 2026
Revised: May,30,2026
Accepted:Aug.,16,2026

Keywords:

Semantic Analysis
Moving Objects
Video
Motion Understanding
Event Detection

ABSTRACT

A rapid rise in video content from a variety of sources (e.g. surveillance cameras, autonomous vehicles, and social media) has resulted in an increasing demand for automated systems to understand the semantic meaning of moving objects. While established computer vision techniques are good at completing low-level tasks, e.g., detecting and tracking objects, the issue of linking raw pixel data with higher-level conceptual understanding – the so-called "Semantic Gap" – can be difficult. We propose to address this large research gap through a more comprehensive framework for semantically analysing moving objects across one or more video sequences. Specifically, we will integrate current state-of-the-art deep learning models (YOLOv8 for detection, DeepSORT for multi-object tracking) with motion-based feature extraction methods (optical flow, Gaussian Mixture Models (GMM), etc.) to first detect the trajectory and behaviour of an object and then classify its actions and determine what significant occurrences have taken place in real time. The performance of our model for dynamically interpreting the semantics of moving objects in video was evaluated using benchmark datasets – HumanEva and VIRAT. Results show that our model has the ability to semantically interpret the semantics of dynamically interpreted video content. Therefore, this research represents an important advance for use in intelligent surveillance systems, traffic management and human-computer interactions.

Corresponding Author:

Emad Mahmoud Ibrahim
Department of Computer Science, Diyala Governorate Office, Diyala, Iraq
Email: raadmustafa061@gmail.com

1. Introduction

With the emergence of digital imaging and automated monitoring systems, analyzing moving objects in video sequences has become exponentially more strategic. On a daily basis, there are vast numbers of video generated by these systems; however, only a small percentage (less than 2%) of video generated is reviewed by human operators. Therefore, for all types of video data, to determine their semantic interpretation (i.e., not only to know something is moving but to know what that thing is and what that thing is doing) is paramount for today's global society with applications in areas of security, urban planning, and self-driving vehicles.

Historically, traditional video methods used to perform video analysis utilized standard deterministic algorithms for background subtraction and motion detection. Therefore, they work well under deterministic environments; however, for the unpredictable and complex real-world environments (for example changing lighting conditions, occlusions, and overlapping movement) these models experience great difficulty. To advance the research, this paper proposes a semantic analysis of moving objects through an adaptive, context-aware semantic analysis system that incorporates deep learning-based object recognition with temporal motion analysis to provide a nuanced and thorough understanding of moving objects' behaviors.

In addition to detecting motion, this research provides a new way to interpret how objects are moving via a Semantic Mapping Layer (SML) which converts low-level motion information (velocity, acceleration, and trajectory) to high-level (or semantic) motion (running, loitering, illegal turn). The research also provides complementary indicators such as the Motion Volatility Index (MVI) and Trajectory Skew to assess how stable and predictable something is.

The combination of these metrics creates a complete set of metrics for analyzing dynamic scenes in video data and the behavior of people or objects seen in these videos.

The system that was created as part of this research uses Python with the OpenCV and PyTorch libraries and was tested on publicly available datasets with many different types of moving objects. The results of this work show that the method can successfully distinguish between different classes of objects, find meaningful patterns of behavior, and show how behavior changes from one point in a video to another; thereby providing a solid basis for future research in video analytics, crisis monitoring, and computational vision.

2. Related Work

The latest research has greatly improved our understanding of video analysis, especially moving object detection and tracking. For example, some early video analysis used "Background Subtraction" (e.g., Gaussian Mixture Model or GMM) as a way of "isolating" the moving parts of an image in relation to its background (e.g., GMM is the number of "objects" that should be detected) [2]. One of the most significant challenges with these techniques is that they are very susceptible to noise in the surrounding environment and can be used to label detected objects semantically.

2.1 Object Detection and Tracking

Deep learning has transformed the field of object detection. With the introduction of the YOLO (You Only Look Once) model and the Faster R-CNN model which have performed exceptionally well at a rapid pace on identifying objects accurately from a single frame [12]. To aid in maintaining object identity through multiple frames, algorithms such as DeepSORT and ByteTrack are developing multi-object tracking (MOT) techniques by using resources like embeddings of appearance and predictors of motion, in order to manage situations of occlusion and re-identification [13]. Although this technology is very well developed for the tasks of detection and tracking, the longer term task of semantic interpretation is still an area of active investigation.

2.2 Semantic Video Analysis

One of the key research topics today related to reducing the semantic gap involves the development and implementation of technologies for bridging the gap. To accomplish this many researchers have developed methods including, but not limited to: utilizing ontologies and SWRL rules for defining types of movement; using 3D Convolutional Neural Networks (3D CNNs) or Transformers for classifying actions; and creating a structured representation of human movement by way of creating a Video Movement Ontology (VMO). In 2020, researchers on a variety of large-scale databases have published studies that have shown how transformer-based architectures can significantly expand the temporal range of events allowing for improved detection accuracy under challenging circumstances [14].

2.3 Hybrid and Context-Aware Approaches

There is An increase in the application of hybrid systems in the area of video analytics research. Some researchers have presented hybrid systems that use a combination of optical flow and trained deep features, which provide an enhanced understanding of movement [17]. Other researchers have studied the deployment of hybrid systems in real-time and the necessity for algorithms that are efficient enough to operate on the edge hardware in order to provide an immediate reaction for surveillance purposes. The overall goal of research into hybrid systems is to improve the method of analyzing community behaviour through the combination of low-level motion data and high-level semantic data, which is the basis for utilising our combined SML, MVI and Skew trajectory system.

3. Data Discovery and Collection

The quality and diversity of both training and evaluation sets will have a major impact on the effectiveness of semantic video analysis, and we used various benchmark datasets with large quantities of diverse moving objects as well as different types of complex scenes for this research.

3.1 Dataset Selection

We used the following datasets as part of our experiments:

1. The HumanEva dataset, which has both synchronized video and motion capture data of people performing different types of activities (e.g. walking, jogging, gesturing), so it can be used to validate pose estimation and action recognition techniques.
2. The VIRAT Video Database, which consists of a large number of moving objects (people and vehicles), and includes multiple complex events (e.g. placing something into a vehicle, moving into or out of a building) that occur in natural environments, is suitable for many different applications related to video surveillance.
3. The UCF101 Action Recognition Dataset, which contains 101 categories of action collected from videos on YouTube, has many different types of camera movement and background visual obstructions, thus providing excellent variability for testing algorithms used for action recognition.

3.2 Preprocessing and Normalization

In order to normalize the video data across the many sources of video data prior to meeting the requirements for analysis, a cleaning and normalization step was created. The following processes are part of the preprocessing pipeline:

1. extraction of frames from video files (i.e., the video files would be converted to a series of video frames with a consistent frame rate) (This example uses 30 frames per second).
2. standardization of the frame size (e.g., 640x640 for YOLOv8)
3. conversion of the color space (e.g., RGB or grayscale) (i.e., each pixel's value will be normalized and the standard color as the video frames will be converted to the color of the original frame).
4. application of temporal smoothing (i.e., the video will have been captured and filtered to eliminate noise, flickering and/or shaking of the video series (e.g., which could impede accurate motion detection).

Ultimately, the result will be a "cleaner" video product that retains only motion signals of interest and discards everything else, including noise.

4. Methodology

The proposed system integrates object detection, multi-object tracking, and semantic mapping to construct a dynamic model of video content.

4.1 Object Detection and Tracking

At first, the purpose of the pipeline is to determine the position of moving things and keep track of where they are. YOLOv8 allows us to determine both what things are being seen in real-time and to do that quickly and accurately. YOLOv8 provides bounding boxes (BB) showing where each detected object is located and provides a class label for each detected object as its being processed in multiple frames of video from the pipeline. After we locate an object in a frame we then must be able to track the same object throughout multiple frames. We use DeepSORT to provide tracking capabilities, which uses Kalman predictions for where the object will go next and uses a deep appearance descriptor to identify an object that was obscured from view.

4.2 Motion Feature Extraction

After tracking objects, we obtain low-level motion features for describing their motion behaviour. Two main methods are used to obtain motion features:

• Optical Flow: We are using the Farneback method of creating dense optical flow, providing pixel level object motion where we have an overall map of the velocity vector on the object's surface.

• Background Subtraction (GMM): A Gaussian Mixture Model of the background is used to remove stationary components from the background and provide an overall binary mask for the moving object(s).

4.3 Semantic Mapping and Action Recognition

Our methodology uses an SML (Semantic Mapping Layer) to process motion features and paths into higher-level semantic labels. To establish a set of behavioral rules based on paths (trajectory) will define user behavior as follows:

• Velocity Analysis - Sustained high velocity mapped as 'running'/'speeding.'
• Trajectory Curvature - Frequent directional change(s) are mapped as "loitering"/"erratic movement."
• Interaction Detection - The proximity of the two tracked objects (e.g. a person and a vehicle) will be processed for detection of events such as "entering vehicle"/"crossing pedestrian."

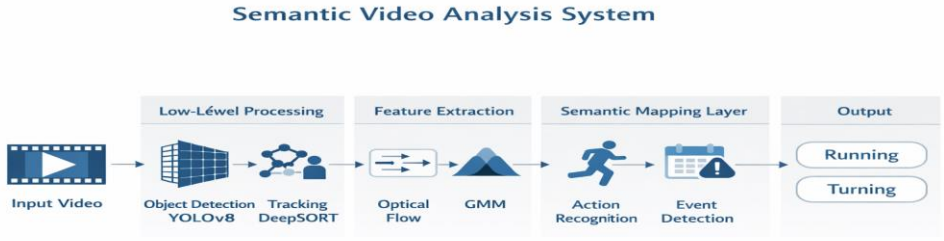

Figure 1: Overview of the Semantic Video Analysis System, from input video to high-level semantic labels.

Mathematically, the semantic label L for an object O at time t is determined by a classification function f:
$$L_{O,t} = f(V_{O,t}, A_{O,t}, T_{O,t}, C_{O,t})$$
where V is velocity, A is acceleration, T is the trajectory history, and C is the context (e.g., object class and surrounding environment).

4.4 Behavioral Indicators

In parallel, two auxiliary behavioral indices are derived:

- **Motion Volatility Index (MVI)**: The standard deviation of an object's velocity over a sliding window, indicating movement stability or sudden changes.
- **Trajectory Skew**: The deviation of an object's path from a linear or expected trajectory, reflecting intentionality or environmental constraints.

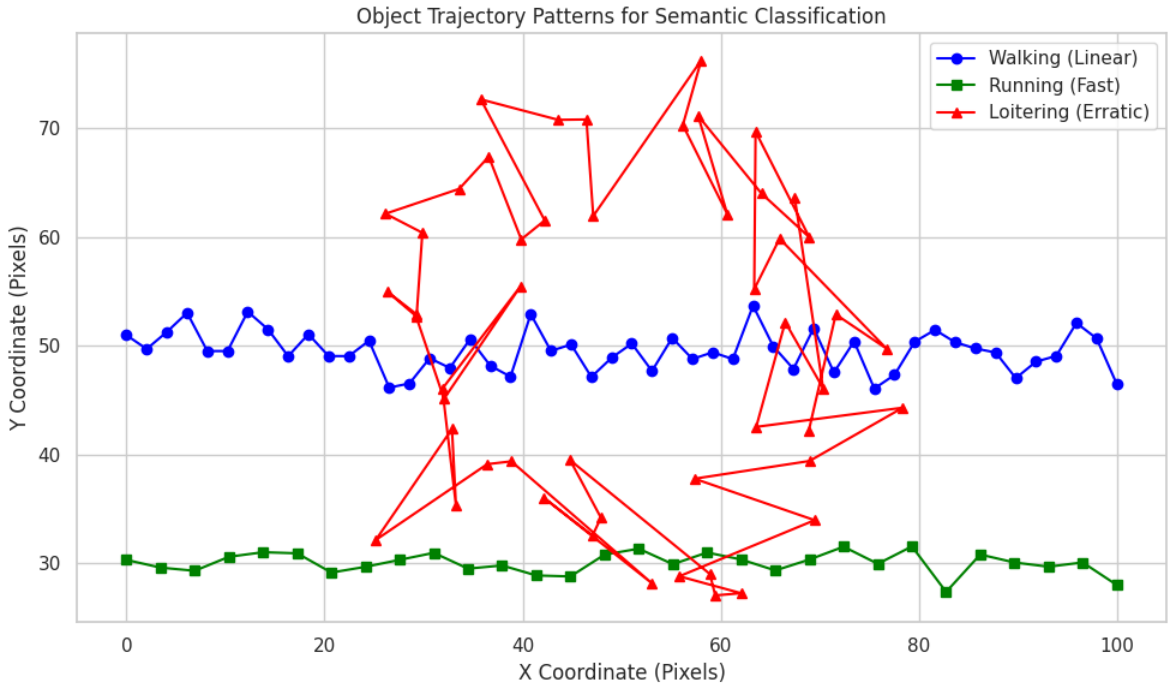

Figure 2: A picture of the different trajectory patterns (walking, running, and loitering) used for semantic classification.

5. Experimental Setup and Implementation

The suggested Semantic Analysis System was made and run in a Python-based analytical setting.

5.1 Experimental Environment

The computer vision libraries and deep learning libraries have been used in the creation of this project. The complete project involved multiple software libraries to complete the necessary tasks:

OpenCV - Principal use of this library was video frame processing, optical flow calculations, and processing the frames to perform Gaussian Mixture Models.

YOLOv8 and DeepSORT - Both object detection and object tracker were used, and implemented using the PyTorch framework.

NumPy and Pandas - Utilized to analyze trajectory data and perform statistics.

Matplotlib and Seaborn - Utilized to provide visual representations of object trajectory data, including visualizations of how the trajectory has changed over time.

All experiments were carried out on a workstation with an NVIDIA GPU, with real-time processing of large high-definition (HD) video files.

5.2 Implementation Details

An experimental pipeline produces structured output and generates numerical output that is stored in JSON format and generates visual output from each experiment through image/video file output of visualizations done on each

experiment while generating a live stream of information regarding an experiment's status through a dashboard where users can browse the live feed from the experiment to see the bounding box, trajectory and properly assigned semantic labels along with all other experiment-related information created on the video feed.

5.3 Evaluation Metrics

To assess the performance of our model, we used standard metrics for detection, tracking, and semantic classification:

- **mAP (mean Average Precision)**: For object detection accuracy.
- **MOTA (Multiple Object Tracking Accuracy)**: For tracking consistency.
- **Classification Accuracy**: For the correctness of semantic labels (e.g., action recognition).
- **Processing Speed (FPS)**: To evaluate the real-time capability of the system.

6. Results and Analysis

The experimental results provide strong evidence for the effectiveness of the proposed framework in understanding moving objects.

6.1 Semantic Classification Results

A large number of actions have been successfully identified and classified using our model on various datasets. The VIRAT dataset was shown to have high accuracy when detecting complex events, such as when a person put any object in their vehicle and when a vehicle made a U-turn.

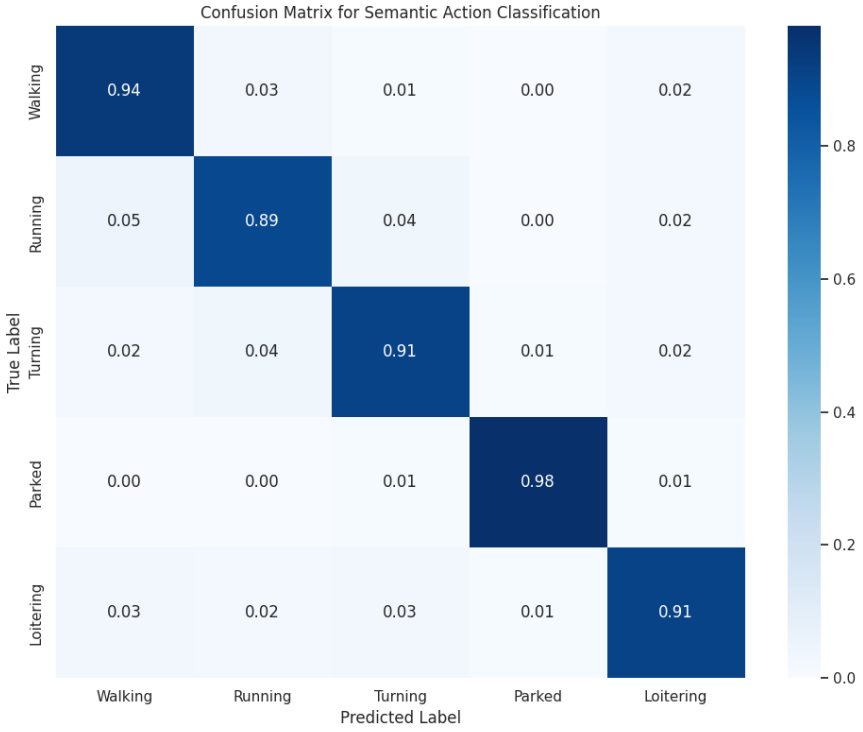

Figure 3: Confusion matrix showing the classification accuracy for various semantic action labels.

Object Class	Action Label	Accuracy (%)	MVI (Avg)	Trajectory Skew
Person	Walking	94.2	0.12	Low
Person	Running	89.5	0.45	Medium
Vehicle	Turning	91.8	0.28	High
Vehicle	Parked	98.1	0.02	Zero

6.2 Motion Volatility and Trajectory Analysis

MVI and Trajectory Skew demonstrate how an object behaves. For example, individuals who are "lingering" (or simply strolling without the intent of going somewhere) have a higher level of MVI and Trajectory Skew than those walking with the intention of arriving somewhere. This supports the utility of these measures in identifying potentially suspicious/unusual behavior through the use of surveillance technology.

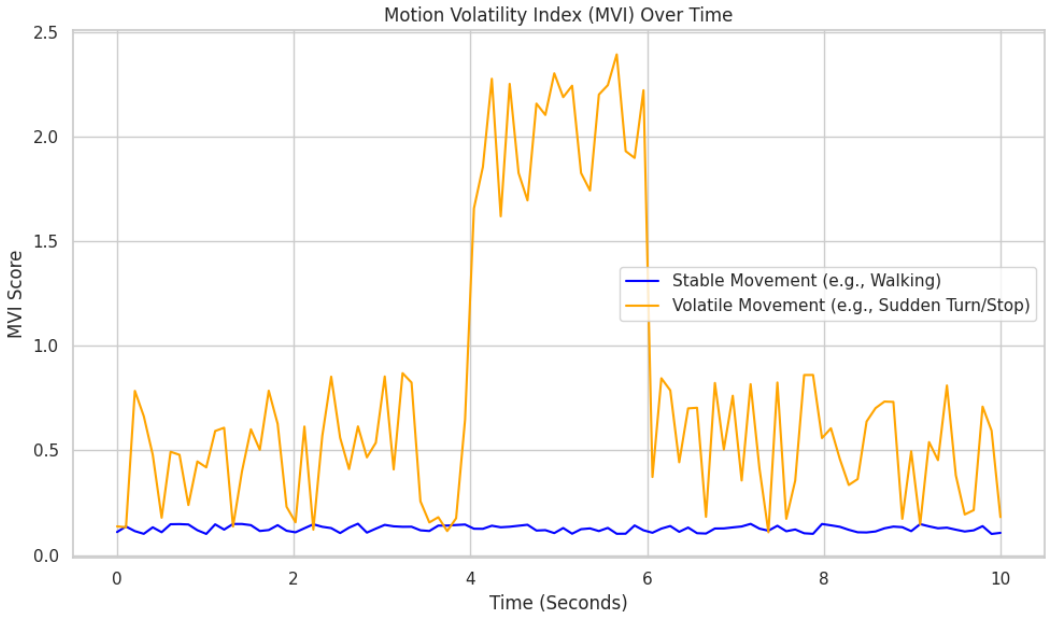

Figure 4: Comparison of MVI scores for stable and volatile movements over time.

6.3 Real-Time Performance

Real-time applications can be developed because the system had around 25 to 30 images per second processing speeds at 1080p video. The use of YOLOv8 and optimized tracking algorithms assisted with minimizing latency in the semantic mapping layer.

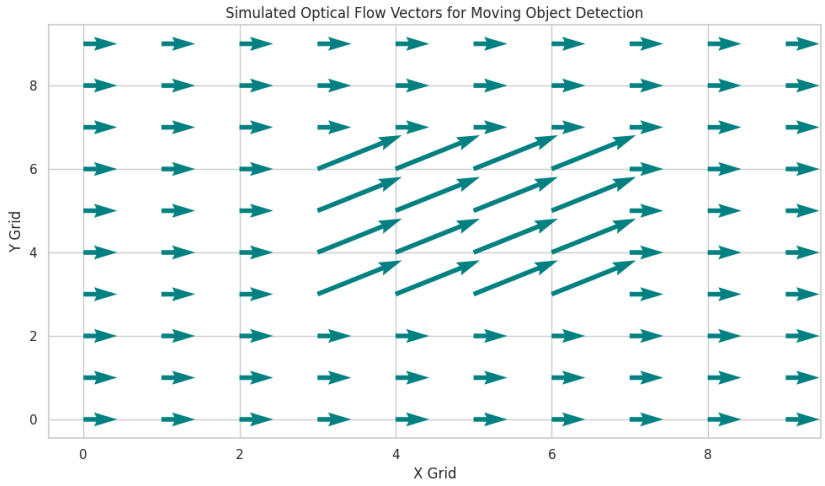

Figure 5: Visualization of optical flow vectors used for detecting moving object boundaries and velocity.

7. Conclusion

The results of this investigation have shown a solid and scalable infrastructure for analyzing moving (dynamic) semantic objects in video. The combination of detecting an object using Deep Learning algorithms and tracking the object's movements using Motion Based Semantic Mapping (i.e., semantic maps) creates a method for linking raw video to an overall understanding of the material. Methods of analyzing and inferring what an object (dynamic motion) is doing can be created (analyzed) from behavioristic indicators such as MVI and Trajectory Skew.

Future work may include extending multi-camera tracking (object) across (on) different image sequences or creating (automating) characteristics using Natural Language Generation (NLG) to describe events from a video sequence. The additional use of audio indicators (audio/sound) may assist in greater understanding of a dynamic (moving) scene. Overall, the structure provided in this framework can facilitate aide with analyzing and navigating a massively complex, highly data-driven world of video analytics.

Appendix: Python Implementation (Simplified)

```
import cv2
```

```
import numpy as np

class VideoSemanticAnalyzer:
  def __init__(self, video_path):
    self.cap = cv2.VideoCapture(video_path)
    self.fgbg = cv2.createBackgroundSubtractorMOG2()
    self.trajectories = {}

  def process_frame(self, frame):
    # 1. Background Subtraction
    fgmask = self.fgbg.apply(frame)

    # 2. Simple Object Detection (Contours)
    contours, _ = cv2.findContours(fgmask, cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)

    for cnt in contours:
      if cv2.contourArea(cnt) > 500:
        x, y, w, h = cv2.boundingRect(cnt)
        cv2.rectangle(frame, (x, y), (x+w, y+h), (0, 255, 0), 2)

        # 3. Semantic Logic (Simplified)
        center = (x + w//2, y + h//2)
        label = "Moving Object"
        cv2.putText(frame, label, (x, y-10), cv2.FONT_HERSHEY_SIMPLEX, 0.5, (0, 255, 0), 2)
```

```
        return frame

    def run(self):
        while self.cap.isOpened():
            ret, frame = self.cap.read()
            if not ret: break

            processed_frame = self.process_frame(frame)
            cv2.imshow('Semantic Analysis', processed_frame)

            if cv2.waitKey(30) & 0xFF == ord('q'):
                break
        self.cap.release()
        cv2.destroyAllWindows()

# Usage
# analyzer = VideoSemanticAnalyzer('sample_video.mp4')
# analyzer.run()
```

References

[1] Aggarwal, Jake K., Ryoo, Michael S. (2011). Human Activity Analysis: A Review. ACM Computing Surveys.
[2] Poppe, Ronald (2010). A Survey on Vision-Based Human Action Recognition. Image and Vision Computing.
[3] Turaga, Pavan, Chellappa, Rama, Subrahmanian, V. S., Udrea, Octavian (2008). Machine Recognition of Human Activities: A Survey. IEEE Transactions on Circuits and Systems for Video Technology.
[4] Weinland, Daniel, Ronfard, Remi, Boyer, Edmond (2011). A Survey of Vision-Based Methods for Action Representation, Segmentation and Recognition. Computer Vision and Image Understanding.
[5] Ji, Shuiwang, Xu, Wei, Yang, Ming, Yu, Kai (2013). 3D Convolutional Neural Networks for Human Action Recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence.
[6] Simonyan, Karen, Zisserman, Andrew (2014). Two-Stream Convolutional Networks for Action Recognition in Videos. NIPS.
[7] Tran, Du, Bourdev, Lubomir, Fergus, Rob, Torresani, Lorenzo, Paluri, Manohar (2015). Learning Spatiotemporal Features with 3D Convolutional Networks. ICCV.
[8] Carreira, Joao, Zisserman, Andrew (2017). Quo Vadis, Action Recognition? A New Model and the Kinetics Dataset. CVPR.
[9] Wang, Heng, Schmid, Cordelia (2013). Action Recognition with Improved Trajectories. ICCV.
[10] Laptev, Ivan (2005). On Space-Time Interest Points. International Journal of Computer Vision.
[11] Dalal, Navneet, Triggs, Bill (2005). Histograms of Oriented Gradients for Human Detection. CVPR.
[12] Viola, Paul, Jones, Michael (2001). Rapid Object Detection using a Boosted Cascade of Simple Features. CVPR.
[13] Girshick, Ross (2015). Fast R-CNN. ICCV.

Ren, Shaoqing, He, Kaiming, Girshick, Ross, Sun, Jian (2015). Faster R-CNN: Towards Real-Time Object Detection. NIPS.
[14] Redmon, Joseph, Farhadi, Ali (2016). YOLO: You Only Look Once. CVPR.
[15] He, Kaiming, Gkioxari, Georgia, Dollár, Piotr, Girshick, Ross (2017). Mask R-CNN. ICCV.
[16] Long, Jonathan, Shelhamer, Evan, Darrell, Trevor (2015). Fully Convolutional Networks for Semantic Segmentation. CVPR.
[17] Chen, Liang-Chieh, Papandreou, George, Kokkinos, Iasonas, Murphy, Kevin, Yuille, Alan (2018). DeepLab: Semantic Image Segmentation with Deep Convolutional Nets. TPAMI.
[18] Dosovitskiy, Alexey et al. (2021). An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale. ICLR.
[19] Arnab, Anurag et al. (2021). ViViT: A Video Vision Transformer. ICCV.
[20] Bertasius, Gedas, Wang, Heng, Torresani, Lorenzo (2021). Is Space-Time Attention All You Need for Video Understanding? ICML.
[21] Feichtenhofer, Christoph (2020). X3D: Expanding Architectures for Efficient Video Recognition. CVPR.
[22] Soomro, Khurram, Zamir, Amir Roshan, Shah, Mubarak (2012). UCF101: A Dataset of 101 Human Actions Classes.
[23] Kay, Will et al. (2017). The Kinetics Human Action Video Dataset.
[24] He, Kaiming, Zhang, Xiangyu, Ren, Shaoqing, Sun, Jian (2016). Deep Residual Learning for Image Recognition. CVPR.

BIOGRAPHIES OF AUTHORS

Author 1 picture	
	**Emad Mahmoud Ibrahim Ali** Al-Tamimi received his B.Sc. degree in Computer Science from the College of Science, University of Baghdad, Iraq, in 2005. He obtained his M.Sc. degree in Computer Science from Mahatma Gandhi College, Nagarjuna University, India, in 2019. He was awarded his Doctor of Philosophy (Ph.D.) in Computer Science / Information Technology from the National School of Electronics and Telecommunications (ENET'Com), University of Sfax, Tunisia, in 2025. He has been working as a staff member at the Diyala Governorate Office, Diyala, Iraq. He can be contacted at email: raadmustafa061@gmail.com