

Patient Activity Analysis from Image and Video using YOLOv8 in Healthcare Settings

Malatesh KR¹, Shashikanta Modak², Darshan B³, Dr. Sreelatha R⁴

^{1,2,3}Information Science and Engineering BMS College of Engineering Bangalore, India

⁴Associate Professor, Information Science and Engineering BMS College of Engineering Bangalore, India

Abstract—This paper presents a computer vision-based system for patient activity analysis in healthcare environments that utilises the YOLOv8 object-detection model. The proposed system analyzes static images and pre-recorded video feeds to detect key patient states and events, such as 'Fallen Patient, Seizure, and 'Patient Sleeping, by classifying visual data with a fine-tuned YOLOv8n model. The methodology integrates a comprehensive pipeline encompassing data input (image/video), frame preprocessing, YOLOv8 inference, event classification, and subsequent visualisation. YOLOv8 was trained on a custom-labelled dataset with 12 classes, derived from publicly available hospital scene images (CC BY 4.0). Experimental results indicate that the system achieved approximately 0.80 for both precision and recall, an F1-score of approximately 0.80, and a mean Average Precision (mAP@0.5) of approximately 0.86 on the designated test data. The system exhibited processing speeds suitable for deployment on GPUs, CPUs, and edge devices. The architecture is designed to support deployment on resource-constrained hardware (leveraging the yolov8n.pt model variant) and is capable of producing annotated visual output. These findings demonstrate the feasibility of employing YOLOv8 for noninvasive patient activity analysis from images and videos, offering a robust approach to enhance patient safety assessment and situational awareness in ICUs, eldercare facilities, and home-based healthcare contexts.

Index Terms—Computer Vision, Patient Monitoring, YOLOv8, Object Detection, Healthcare, Fall Detection, Activity Recognition, Ethical AI, Tele-ICU, Eldercare, Deep Learning

I. INTRODUCTION

Monitoring of patient activities and states is of paramount importance in diverse healthcare settings, including Intensive Care Units (ICUs), general hospital wards, and specialised elderly care facilities, particularly for individuals at an elevated risk of adverse events such as falls or seizures [4], [11].

Traditional monitoring methodologies, which predominantly rely on manual supervision by healthcare personnel or the use of wearable physiological sensors, are often encumbered by significant limitations. These include the substantial costs associated with continuous staffing, challenges related to patient compliance and comfort with wearable apparatus, inherent lacunae in observational coverage, and the potentially intrusive nature of certain sensing technologies [2], [12]. Such constraints can culminate in delays in the detection of critical events, thereby adversely affecting patient outcomes.

In recent years, computer vision, substantially augmented by advancements in deep learning, has emerged as a highly promising non-contact and potentially more scalable paradigm for patient activity analysis [27]. Strategically positioned cameras can record visual data, which can then be analysed to facilitate the automated detection of behaviours (e.g. falls and epileptic seizures) or critical changes in a patient's physiological or behavioural state from images and videos [4], [6], [13], [24]. Deep learning models have demonstrated considerable efficacy in interpreting complex visual data even in dynamic and often cluttered medical environments [9], [28]. Consequently, vision-based systems have been increasingly investigated for various healthcare applications, including fall detection [4], [11], [14], [23], comprehensive patient activity recognition [12], [25], [26], [30], ICU-specific monitoring from recordings [5], and noninvasive pain assessment from visual cues [13]. Furthermore, AI-driven video analytics can significantly augment telemedicine and tele-ICU operations by enabling the review of rich visual data streams and AI-driven analyses that can flag critical incidents in recorded footage [10]. Although the deployment of such systems necessitates careful consideration of privacy and ethical concerns [20], [21], the performance of

AI-powered vision methods in specific medical tasks is rapidly approaching and, in some instances, exceeding human-level accuracy [9].

The "You Only Look Once" (YOLO) family of object detectors is recognised for its compelling balance of processing speed and detection accuracy, rendering it suitable for analysing image and video data [1]–[3], [29]. YOLO models are single-stage detectors that process full images once through a Convolutional Neural Network (CNN) to predict bounding boxes and class probabilities directly [1], facilitating efficient end-to-end training. YOLOv8, the version employed in this Study, incorporates several architectural enhancements, including an anchor-free detection head, novel CSPDarknet-based backbone, and improved Path Aggregation Network (PAN) neck structure. These modifications contribute to state-of-the-art performance while maintaining computational efficiency [4], [13], [15]. Notably, smaller variants such as YOLOv8n ("nano") are designed for deployment on resource-constrained edge devices, which is a critical factor for developing scalable and cost-effective healthcare analysis tools [5], [16]. This study presents the development and evaluation of a vision-based pipeline leveraging a lightweight YOLOv8n variant to identify and classify key patient states (e.g. Fallen Patient, Seizure, and Patient Sleeping) from static images and pre-recorded video streams, with the objective of enhancing patient safety assessment and augmenting caregiver situational awareness through data review.

II. RELATED WORK

Vision-based patient activity analysis systems that employ deep learning techniques, particularly YOLO models, have garnered considerable attention. Current research predominantly spans fall and activity detection, medical event and state detection from visual data, eldercare monitoring through recorded media, and applications within ICU and hospital environments, focusing on data review. Studies incorporating CNN-LSTM architectures [25], [30], pose estimation for action recognition [23], [24], and transformer networks [26] underscore the breadth of deep learning methodologies being explored for Human Activity Recognition (HAR) from images and videos.

A. Fall and Activity Detection

Automated fall detection from visual data remains a significant research focus owing to the severe implications of falls, especially among the elderly and hospitalised individuals [11]. Bai et al. [4] utilized YOLOv8 in conjunction with OpenCV, reporting precision and recall values of approximately 80% and a mAP@0.5 of 0.86 for fall detection in images/videos. Other investigations have explored alternative YOLO versions, such as YOLOv7, often combined with pose estimation techniques, to enhance the fall detection accuracy from recorded footage [14], [24]. Hybrid approaches, integrating YOLO for initial person detection with specialised action recognition models such as Graph Convolutional Networks (GCNs) operating on skeletal data extracted from videos, have also been proposed for complex scenarios [12], [26], [31]. Earlier iterations, including YOLOv4 [2] and YOLOv7 [3], have also demonstrated efficacy in analysing recorded activities. Beyond acute events, broader patient activity recognition seeks to understand a wider spectrum of behaviours from visual records, including mobility levels and periods of inactivity [17], [28], [32]. A comprehensive overview of HAR across various modalities was provided by Shin et al. [28].

B. Medical Event and State Detection

Computer vision techniques are increasingly being applied to detect specific medical events and states from visual data.

Vision-based seizure detection systems, for instance, aim to identify characteristic motor manifestations in video feeds [6]. YOLO models have also been adapted for tasks such as patient posture detection from images to aid in pressure ulcer prevention [7]. Fiorini et al. [13] explored the use of YOLOv8 for pain level assessment through facial expression analysis in images. Wang et al. [27] offer a review of Deep Neural Networks (DNNs) in video-based HAR, and Kulkarni and Jadhav [33] discuss deep learning applications in HAR.

C. Eldercare and Home Monitoring

Vision-based systems provide an unobtrusive means of analysing recorded data to monitor the daily activities and safety of seniors [8], necessitating a balance between functional analysis and stringent privacy requirements [20], [21]. Recent developments include systems using YOLOv8 with tracking algorithms such as ByteTrack for

monitoring room entry/exit from video records, particularly relevant for patients with cognitive conditions such as Alzheimer’s disease [16]. The ethical implications of analysing such recorded data are of paramount importance [19]–[22].

D. ICU and Hospital Monitoring

AI-driven video analytics is being integrated into acute hospital settings, particularly ICUs, to review recorded data. Systems described by Gabriel et al. [5] utilise models such as YOLOv4 to detect patient presence, mobility, and staff interactions from video footage, supporting tele-ICU objectives through retrospective analysis [10]. YOLOv8 has also been employed to monitor compliance with safety protocols, such as face mask usage, in recorded videos [18]. Advanced techniques involving 3D human pose estimation [23] and Recurrent Neural Networks (RNNs) [34] further contribute to HAR capabilities from video. The present study extends this body of work by developing and evaluating a unified YOLOv8-based pipeline focused on key patient states identified from images and videos, emphasising the processing efficiency and potential for edge deployment for data analysis.

III. METHODOLOGY

The patient activity analysis system detailed herein is conceptualised as a sequential pipeline that transforms the input image or video data into classified events and informative visualisations, as illustrated in Fig. 1.

A. System Architecture

The system architecture encompasses the following stages:

- 1) Data Input: The system processes static image files or prerecorded video files.
- 2) Frame Preprocessing: Each frame extracted from the input (image or video) undergoes requisite preprocessing, typically including resizing to the model’s input dimensions (e.g., 640×640 pixels) and normalization of pixel values.
- 3) YOLOv8 Inference: The preprocessed frame is inputted into a fine-tuned YOLOv8n model (yolov8n.pt).

This model performs object detection to identify instances of predefined patient states or activities, and

predicts their corresponding bounding boxes and class labels based on the learned weights (best.pt) from our custom healthcare dataset.

- 4) Post-processing: Raw detections from YOLOv8 are refined by filtering based on a confidence threshold (e.g., 0.5) and applying Non-Maximum Suppression (NMS) to eliminate redundant bounding boxes.
- 5) Event Classification & Alerting: The system analyzes class labels of post-processed detections. If a detection corresponds to a predefined critical class (e.g., ‘Fallen Patient’, ‘Seizure’) in an image or video frame and its confidence score surpasses the established threshold, this event is flagged. Alerts can be generated for review.
- 6) Visualization: For operator feedback and verification, detected bounding boxes, class labels, and confidence scores are overlaid onto the original image or video stream.

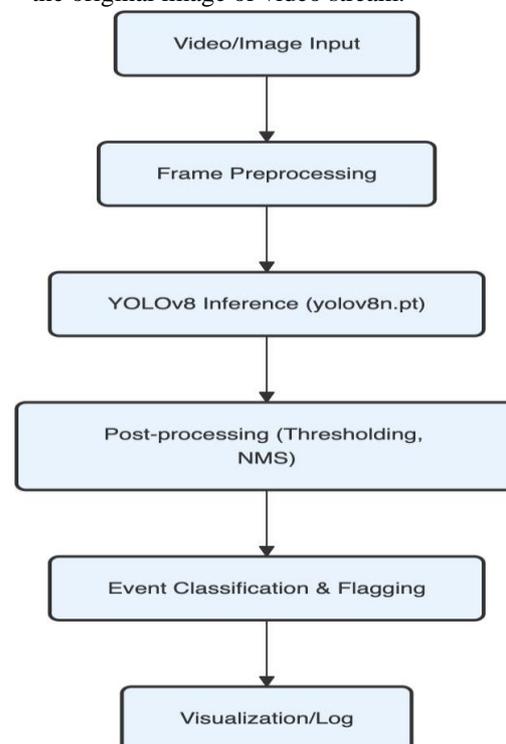


Fig. 1. High-level architecture of the YOLOv8-based patient activity analysis system, illustrating the data flow from image/video input to event classification and visualization.

B. YOLOv8 Integration and Architecture

The Ultralytics YOLOv8 framework [15] was employed, specifically the yolov8n variant, selected for its advantageous trade-off between detection accuracy and computational efficiency. This

characteristic facilitates deployment across diverse hardware platforms to analyse image and video data [5], [16].

The YOLOv8 architecture incorporates several key improvements over its predecessors.

- **Backbone:** It utilizes a new CSPDarknet-based backbone (e.g., a variant of CSPDarknet53), optimized for efficient feature extraction.
- **Neck:** An enhanced Path Aggregation Network (PANet) or a similar structure is employed in the neck to effectively fuse features from different scales of the backbone, thereby improving the detection of objects of various sizes.
- **Head:** YOLOv8 employs an anchor-free detection head. This design directly predicts object centres and dimensions, potentially simplifying the detection process and reducing post-processing overhead. It often incorporates a decoupled head for classification and regression tasks, which can enhance performance.

The model, initially pre-trained on the large-scale Common Objects in Context (COCO) dataset, was subsequently fine-tuned on our custom patient-monitoring dataset. Python scripts for image and video testing load this fine-tuned model (for example, `textttmodel = YOLO('path/to/best.pt')`). Image or video frames were then processed by passing them to the model: `results = model(frame, conf=0.5, verbose=False)`. Argument `textttconf = 0.5` represents an empirically determined confidence threshold.

C. Classes for Patient Monitoring

The system is specifically trained and configured to recognise 12 key classes relevant to patient monitoring and health-care environments. These classes encompass a range of personnel, patient states, critical medical events, and environmental objects.

- **Personnel:** 'Doctor', 'Nurse', 'Visitor'. These classes identify the presence of different individuals within the monitored scene.
- **Patient States:** 'Patient Sleeping', 'Patient Awake', 'Patient Sitting', 'Patient Coughing'. These classes aim to capture various common patient activities and conditions.
- **Critical Medical Events:** 'Heart Attack',

'Seizure', 'Fallen Patient'. These classes are designed to detect urgent situations that require immediate attention.

- **Environmental Context:** 'Monitor' (medical monitoring equipment), 'Bed'. These classes provide contextual information regarding the patient's immediate surroundings.

These classes were selected based on their clinical significance in ensuring patient safety, monitoring patient well-being, and understanding the operational context of a healthcare setting, such as an ICU.

D. Dataset and Augmentation

The model was trained and evaluated using a custom dataset structured according to the specifications in `data.yaml` (see Section III.E). This dataset, derived from publicly available hospital scene images (licensed under CC BY 4.0, potentially sourced or curated via platforms such as Roboflow Universe), consists of images annotated with bounding boxes for the 12 target classes defined above. The dataset was partitioned into standard subsets: training (approximately 70%), validation (approximately 20%), and testing (approximately 10%). During the training phase, standard data augmentation techniques integrated into the Ultralytics YOLOv8 training framework were used. These techniques, including geometric transformations, photometric adjustments, mosaic augmentation, MixUp, and random flips, are crucial for improving model generalisation and robustness against real-world visual variations [19]. The annotations are provided in the standard YOLO format.

E. Data.yaml Configuration

The structure of the dataset, paths to the image directories and class information are defined in a `data.yaml` configuration file. This file is essential for the YOLOv8 training pipeline to correctly locate and interpret the data, particularly for the 12 classes involved in this study. An example of this configuration is presented in Listing 1.

```

1 # Dataset configuration for patient monitoring
2 train: ./train/images
3 val: ./valid/images
4 test: ./test/images # Optional
5
6 # Classes
7 nc: 12 # Number of classes
8 names:
9   - Doctor
10  - Nurse
11  - Visitor
12  - Patient Sleeping
13  - Patient Awake
14  - Patient Sitting
15  - Patient Coughing
16  - Heart Attack
17  - Seizure
18  - Fallen Patient
19  - Monitor
20  - Bed

```

Listing 1. Illustrative data.yaml configuration for YOLOv8 training with 12 classes.

This data.yaml file acts as a manifest, linking the YOLOv8 training scripts to the respective data splits and defining the semantic meaning of the class indices used in the annotation files and predicted by the model. The nc field specifies the total number of distinct classes (12), and the names list provides human-readable labels for each class index.

IV. IMPLEMENTATION

The patient activity analysis system was implemented in Python, primarily utilising the ultralytics library for YOLOv8 model operations and opencv-python (cv2) for image/video handling and visualisation tasks. The implementation focuses on scripts for model testing of static images and prerecorded videos.

A. Core Python Scripts for Testing and Analysis

The system utilises dedicated Python scripts to evaluate the trained model on different types of visual media. The trained YOLOv8 model (for example, last.pt or best.pt, located in ./runs/detect/train/weights/) was loaded into each script for inference.

1) *Image Testing Script ('imagetest.py')*: This script is designed to apply the trained YOLOv8 model to individual static images to detect patient activities or states. It performs the following operations: loads the pre-trained model, reads a specified image file (e.g. test.jpg), runs the YOLOv8 inference to identify objects, and classifies them according to the defined patient states. The script then displays the original image with overlaid bounding boxes and class labels for each detection. The Ultralytics YOLO library typically saves the

annotated image to a designated prediction directory. The fundamental logic of this script is presented in Table 2.

```

1 # imagetest.py
2 from ultralytics import YOLO
3 import cv2
4 import os
5
6 # Load the trained YOLOv8 model
7 model_path = os.path.join('.', 'runs', 'detect', 'train',
8   'weights', 'last.pt')
9 model = YOLO(model_path)
10
11 # Path to the image you want to test
12 image_path = "test.jpg" # Replace with your image filename
13
14 # Run detection
15 # The 'save=True' argument directs YOLO to save the output
16 # image
17 results = model(image_path, save=True, conf=0.5)
18
19 # Display the output image
20 # YOLO typically saves prediction outputs in a
21 # 'runs/detect/predictN' directory
22 predicted_image_folder = os.path.join('runs', 'detect')
23 predict_folders = []
24 if os.path.exists(predicted_image_folder):
25     predict_folders = [d for d in
26       os.listdir(predicted_image_folder) if
27       d.startswith('predict') and
28       os.path.isdir(os.path.join(predicted_image_folder,
29         d))]
30
31 if predict_folders:
32     predict_folders.sort(key=lambda x:
33       int(x.replace('predict','')) if
34       x.replace('predict','').isdigit() else -1,
35       reverse=True)
36     latest_predict_folder_name = predict_folders[0]
37     latest_predict_folder_path =
38       os.path.join(predicted_image_folder,
39         latest_predict_folder_name)
40     predicted_path =
41       os.path.join(latest_predict_folder_path,
42         os.path.basename(image_path))
43
44 if os.path.exists(predicted_path):
45     img = cv2.imread(predicted_path)
46     if img is not None:
47         cv2.imshow("Detection Result", img)
48         cv2.waitKey(0)
49     else:
50         print(f"Error: Could not read predicted image
51           at {predicted_path}")
52 else:
53     print(f"Prediction image not found at
54       {predicted_path}")
55     print(f"Detected objects: {len(results[0].boxes)
56       if results and results[0].boxes else 0}")
57
58 else:
59     print(f"No 'predict' directory in
60       {predicted_image_folder}.")
61
62
63 cv2.destroyAllWindows()

```

Listing 2. Python script for model testing on a single image ('imagetest.py').

2) *Video Testing Script ('videotest.py')*: This script enables the evaluation of the model's performance on pre recorded video files (e.g., icu_test_video.mp4). It loads the trained YOLOv8 model and the specified video. The script then processes the video frame by frame: for each frame, it performs inference to detect patient states/activities. Detections that meet a defined confidence threshold (e.g., 0.5) are visualized by drawing bounding boxes and corresponding class

labels directly onto the frame. The processed video, with these annotations, is then displayed. User interaction, typically a key press (e.g., 'q'), allows for the termination of the video processing and display. Listing 3 provides an outline of this script's functionality. The CRITICAL_CLASSES list was updated to reflect the relevant classes from the 12-class dataset.

```

1 # videotest.py
2 from ultralytics import YOLO
3 import cv2
4 import os
5
6 # Load trained model
7 model_path = os.path.join('.', 'runs', 'detect', 'train',
8   'weights', 'last.pt')
9 model = YOLO(model_path)
10
11 # Load video file
12 video_path = "icu_test_video.mp4" # Replace with your
13   video filename
14 cap = cv2.VideoCapture(video_path)
15
16 if not cap.isOpened():
17     print(f"Error: Could not open video {video_path}")
18     exit()
19
20 CONFIDENCE_THRESHOLD = 0.5 # Define confidence threshold
21 # Define critical classes based on the 12-class model.names
22 CRITICAL_CLASSES = ['Fallen Patient', 'Seizure', 'Heart
23   Attack']
24
25 whereas cap.isOpened():
26     ret, frame = cap.read()
27     if not ret:
28         break
29
30     results = model(frame)[0]
31
32     for the box in results.boxes.data.tolist():
33         x1, y1, x2, y2, score, class_id = box
34         if score > CONFIDENCE_THRESHOLD
35             class_name = model.names[int(class_id)].upper()
36             label = f"{class_name} {score:.2f}"
37
38             color = (255, 0, 0) # Default Blue for
39               non-critical
40             # Standardise class name for comparison (for
41               example, 'FALLEN PATIENT')
42             current_class_standardised =
43               class_name.replace(' ', '_').upper()
44             critical_classes_standardised = [c.replace('
45               ', '_').upper() for c in CRITICAL_CLASSES]
46
47             if current_class_standardised in
48               critical_classes_standardised.
49                 color = (0, 0, 255) # Red for critical
50
51             cv2.rectangle(frame, (int(x1), int(y1)),
52               (int(x2), int(y2)), color, 2)
53             cv2.putText(frame, label, (int(x1), int(y1) -
54               10),
55
56                 cv2.FONT_HERSHEY_SIMPLEX, 0.9,
57                 color, 2)
58
59             if current_class_standardised in
60               critical_classes_standardised and score >
61               0.7.
62                 print(f"ALERT: Critical event detected -
63                   {class_name}")
64
65
66 cv2.imshow("Video Detection", frame)
67
68 if cv2.waitKey(1) & 0xFF == ord("q"):
69     break
70
71 cap.release()
72 cv2.destroyAllWindows()

```

Listing 3. Python script for model testing on a pre-recorded video ('videotest.py').

These scripts provide the primary means for applying the trained model to analyse static images and prerecorded video data for patient activity assessment.

B. Hardware/Software Setup

The development and evaluation were performed on a system equipped with an NVIDIA GPU (for example, Tesla T4, RTX series) utilising CUDA for acceleration. The YOLOv8n model can also be run on CPUs. The software environment comprised Python (version 3.9 or newer), PyTorch (version 1.10 or later), the ultralytics package (version 8.x), OpenCV-Python (version 4.5 or newer), and NumPy. For deployment on dedicated edge devices (for example, NVIDIA Jetson platforms), YOLOv8n can be further optimised using tools such as ONNX and TensorRT [16].

C. Thresholds and Processing Flow

A confidence threshold of 0.5 was empirically established for general object detections. For flagging critical events such as 'Fallen Patient' or 'Seizure' in analysed data, a more stringent confidence threshold (e.g., 0.7 to 0.75) may be applied. In addition, temporal consistency checks (e.g. those requiring detection across several consecutive frames in video analysis) can be incorporated to mitigate false positives [11], [14]. The processing pipeline (load → preprocess → infer → postprocess → visualize/flag) is optimised for efficient operation on static images and video files.

D. Visualization and Alerts

Visual feedback is provided by rendering bounding boxes around detected patient states/activities and overlaying labels with class names and confidence scores using OpenCV, which is applicable to both image and video outputs. Flagged events in analysed data can be presented through on-screen visual warnings during review or logged for audit trails.

V. EVALUATION

The performance of the fine-tuned YOLOv8n model was rigorously assessed using a held-out test set that constituted approximately 10% of the custom 12-class dataset.

A. Metrics

The evaluation employed the following standard

Fig. 2. Sample detection outputs from the YOLOv8n-based system, illustrating various detected patient activities and personnel.

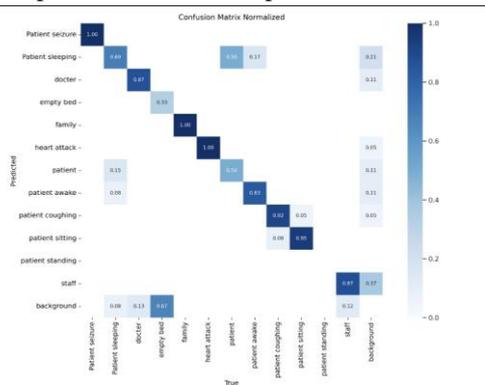


Fig. 3. Normalized confusion matrix illustrating the classification performance of the YOLOv8n model across the 12 defined patient activity and personnel classes.

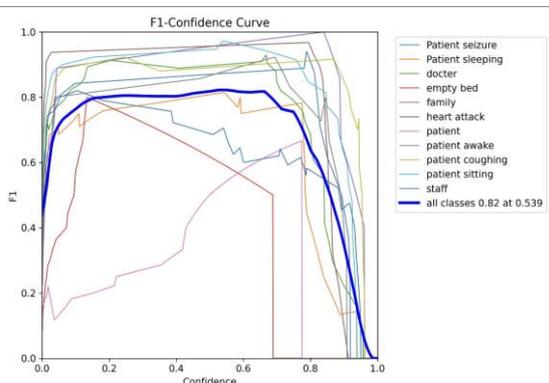


Fig. 4. F1-Confidence curve for the YOLOv8n model, showing the F1 score variation with changing confidence thresholds for all classes and individual classes.

pose estimation [14], [23] or temporal modelling (for example, LSTMs/transformers [12], [25], [26], [30]) may yield higher accuracy on specific subtasks. The yolov8n model offers a compelling and robust balance for general patient state analysis from visual data, particularly for deployment on diverse hardware, including resource-constrained edge devices.

D. Limitations

The proposed system, when applied to image and video analyses, is subject to several inherent limitations.

- **Environmental Factors:** System efficacy can be compromised by suboptimal environmental conditions in the source images/videos, such as inadequate or inconsistent illumination, strong shadows, or excessive background clutter.
- **Occlusion:** Partial or complete occlusion of the

patient by objects (e.g., blankets, medical equipment, other individuals) in the visual data presents a significant challenge to detection [16].

- **Similar Poses and Subtle Activities:** Distinguishing between visually similar postures or subtle activities (e.g., ‘Patient Sleeping’ vs. ‘Patient Awake’ in certain poses) based solely on single-frame visual cues can be inherently difficult.
- **Dataset Bias and Generalization:** The model’s performance is intrinsically linked to the diversity, quality, and representativeness of the training dataset [28]. Insufficient coverage of patient demographics, environmental variations, or activity nuances can lead to suboptimal performance when analysing novel scenarios or under-represented patient groups [19], [20].
- **Privacy Concerns:** The analysis of video data in health-care settings raises substantial ethical and privacy considerations [6], [19]–[22], necessitating meticulous system design and operational protocols for data handling (further discussed in Section VI.A).
- **Reliance on Visual Cues:** The system is fundamentally limited to detecting events and states that possess clear, discernible visual manifestations in the provided images or videos.
- **Alerting Logic Sophistication:** Translating raw detections from analyzed data into clinically reliable flags or alerts requires carefully engineered post-processing and alerting logic to minimize false positives during review.

Addressing these limitations is crucial for the responsible and effective application of such analytical systems in real-world healthcare environments.

VI. DISCUSSION

The evaluation results indicated that YOLOv8n exhibits substantial potential for patient activity analysis based on images and videos. However, the successful and ethical integration of such AI-driven visual data analysis into sensitive healthcare environments necessitates a thorough consideration of several factors, including profound ethical and privacy implications, specific use-case suitability, and avenues for future refinement and improvement.

A. Ethical and Privacy Considerations

The application of AI-driven video analysis systems in healthcare settings requires rigorous ethical oversight and a proactive approach to privacy protection [20]–[22]. Key considerations include the following:

- **Privacy and Data Security:** Adherence to data protection regulations, such as HIPAA and GDPR, is imperative when handling patient visual data [19], [21]. Essential strategies encompass obtaining informed consent for data collection and analysis [20], practising data minimisation, employing anonymisation or de-identification techniques where feasible [19], prioritising on-device or edge processing for analysis to reduce data transmission [5], [16], and implementing robust encryption and access control mechanisms for data storage and transmission [19], [21].
- **Bias and Fairness:** AI models can perpetuate biases present in training data, potentially leading to performance disparities across demographic groups when analyzing their data [19], [20], [22], [28]. Mitigation strategies include curating diverse and representative datasets, conducting regular bias audits using an analysis tool [19], and exploring fairness-aware algorithmic techniques.
- **Transparency and Explainability:** Efforts should be made to enhance transparency regarding system operation, capabilities, and limitations for both clinicians and patients when reviewing analyzed data [19], [20], [22].
- **Accountability and Clinical Oversight:** Clear lines of accountability for system performance and interpretation of results must be established. AI systems are intended to augment, not replace, clinical judgment in reviewing analysed data.
- **Potential for Misuse:** Strong governance frameworks are necessary to prevent misuse of analyzed data beyond intended clinical scope [19].

Addressing these ethical challenges proactively is fundamental to building trust and ensuring responsible technological adoption for patient data analysis.

B. Potential Use Cases

The proposed patient activity analysis system leveraging YOLOv8n has a wide spectrum of applications for reviewing image and video data.

- **Hospitals (ICUs, General Wards):** Enhancing patient safety via automated review and detection of critical events (e.g., ‘Fallen Patient’, ‘Seizure’, ‘Heart Attack’) in recorded videos [5]; analyzing patient mobility and activity levels (e.g., ‘Patient Sitting’, ‘Patient Sleeping’) from footage; potentially assisting in auditing protocol compliance (e.g., presence of ‘Doctor’ or ‘Nurse’ [18]) from recordings.
- **Eldercare Facilities:** Offering a method for unobtrusive review of recorded data for high-risk elderly residents, focusing on fall detection, inactivity alerts, and recognition of unusual behavior patterns from video logs [4], [11], [14].
- **Home Healthcare and Telemedicine:** Supporting remote analysis of recorded data for individuals living independently or patients with chronic conditions, facilitating review and timely alerts based on flagged events in submitted videos [8], [10], [27].
- **Rehabilitation Centers:** Assisting therapists by analyzing recorded sessions to track patient movements and adherence to prescribed exercises.
- **Specialized Care Units:** Providing an additional tool for reviewing recordings in units managing patients prone to specific events, such as epilepsy monitoring units.

System implementation for data analysis must be tailored to the unique requirements of each environment, ensuring data integrity, appropriate review protocols, and integration with existing clinical information systems.

C. Future Directions

Future research endeavours could focus on several avenues to augment system capabilities and clinical utility for image and video analysis.

- **Model Enhancement and Hybrid Architectures:** Exploration of hybrid models combining YOLOv8 with pose estimation techniques [14], [23], [24] or temporal modeling frameworks (e.g., LSTMs, transformers [12], [25], [26], [30]) for richer analysis of video data. An investigation of newer YOLO versions (e.g. YOLOv9 [29]) is also warranted.
- **Improving Robustness to Real-World Challenges:** Development of techniques to enhance model resilience against occlusions, dynamic lighting conditions in source media, and to improve generalization across varied

environments and patient demographics, potentially through domain adaptation strategies applied to analysis of diverse datasets [27].

- **Expanded Range of Detectable States and Activities:** Increasing the granularity of detectable states from images/videos, such as specific types of agitation, subtle pain indicators [13], or discernible changes in respiratory patterns, which would necessitate the creation of new, meticulously annotated datasets.
- **Multi-Modal Data Integration:** Fusion of visual data analysis with information from other recorded sensor modalities (e.g., audio from videos, wearable sensor logs [11], [28], [30], depth sensor data [24]) to create more comprehensive retrospective monitoring solutions.
- **Advanced Edge AI Optimization:** Further optimization of YOLOv8n and the analysis pipeline for efficient processing of image/video data on low-power edge AI devices [5], [16].
- **Clinical Validation, Usability, and Ethical Refinement:** Conducting extensive real-world clinical trials of the analysis system in diverse clinical settings to validate its performance, safety, and utility against established benchmarks and current review practices [33], [34]. Continuous assessment of biases [19], [20] and refinement of data governance and ethical protocols are essential.
- **Federated Learning:** Investigating federated learning for privacy-preserving model training on distributed datasets without sharing raw patient visual data.

VII. CONCLUSION

This investigation presents the development and evaluation of a patient activity analysis system employing the YOLOv8n deep-learning model. Through fine-tuning on a custom-labelled dataset with 12 classes representative of healthcare scenarios, the system demonstrated effective detection and classification of a range of patient states and personnel from images and videos, including critical events like ‘Fallen Patient’ and ‘Seizure’, alongside activities such as ‘Patient Sitting’ and ‘Patient Sleeping’. The resultant performance metrics, notably a mean Average Precision (mAP@0.5) of approximately 0.86 and Precision, Recall, and F1-scores around 0.80,

combined with efficient processing speeds for visual data, affirm the considerable potential of YOLOv8n as an effective tool for noninvasive analysis of patient activity in diverse healthcare contexts.

This study elucidates the capacity of contemporary lightweight object detection models to contribute significantly to tools for enhancing patient safety assessment and situational awareness through a review of visual data. The system architecture, designed for efficient processing and potential deployment on edge devices for analysis tasks, offers a scalable approach to augment traditional patient data review methods. Nevertheless, the practical and widespread adoption of such technology is contingent upon the diligent mitigation of inherent limitations, particularly those pertaining to environmental variability in source media, patient occlusion, and the critical ethical and privacy considerations attendant to the analysis of patient visual data in healthcare. Proactive implementation of robust data security measures, adherence to patient consent protocols for data usage, and strategies for mitigating algorithmic bias are paramount.

While the obtained results are promising, future research should strategically focus on enhancing the model robustness for analysing challenging visual data (e.g. with occlusions and varied illumination), expanding the repertoire of detectable clinically relevant activities and states, and exploring multi-modal sensor data fusion for more comprehensive analytical solutions. Rigorous clinical validation in authentic health-care environments, coupled with the continuous refinement of ethical guidelines and user-centric design principles for analytical tools, will be indispensable for translating the technological capabilities demonstrated in this study into tangible improvements in patient care, safety assessments, and clinical decision-making in hospitals, elderly care facilities, and home healthcare environments through the effective analysis of image and video data. The ongoing evolution of artificial intelligence and computer vision holds significant promise for transforming patient activity analysis paradigms, and this study contributes to expanding the corpus of evidence supporting this transformative potential.

REFERENCES

- [1] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You Only Look Once: Unified, Real-Time Object Detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, 2016, pp. 779–788.
- [2] A. Bochkovskiy, C.-Y. Wang, and H.-Y. Liao, "YOLOv4: Optimal Speed and Accuracy of Object Detection," *arXiv preprint arXiv:2004.10934*, 2020.
- [3] C.-Y. Wang, A. Bochkovskiy, H.-Y. M. Liao, "YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors," *arXiv preprint arXiv:2207.02696*, 2022.
- [4] M. Rama Bai et al., "Real-Time Human Fall Detection using YOLOv8 and OpenCV," *Int. J. Adv. Res. Comput. Commun. Eng.*, vol. 13, no. 5, pp. 321–327, May 2024.
- [5] P. Gabriel et al., "Continuous patient monitoring with AI: real-time analysis of video in hospital care settings," *Front. Imaging*, vol. 4, 2025, Art. no. 1547166. doi: 10.3389/fimag.2025.1547166.
- [6] B. M. Brown et al., "Computer vision for automated seizure detection and classification: A systematic review," *Epilepsia*, vol. 65, no. 5, pp. 1176–1202, May 2024.
- [7] B. Aldughayfiq et al., "YOLO-Based Deep Learning Model for Pressure Ulcer Detection and Classification," *Sensors*, vol. 23, no. 6, p. 2853, Mar. 2023.
- [8] M. Buzzelli, A. Albe', and G. Ciocca, "A Vision-Based System for Monitoring Elderly People at Home," *Appl. Sci.*, vol. 10, no. 1, p. 374, Jan. 2020.
- [9] A. Esteva et al., "Deep learning-enabled medical computer vision," *npj Digit. Med.*, vol. 4, no. 1, p. 5, Jan. 2021.
- [10] C. Udeh, B. Udeh, and N. Rahman, "Telemedicine/Virtual ICU: Where Are We and Where Are We Going?," *J. Intensive Care Medicine.*, vol. 33, no. 7, pp. 446–452, Jul. 2018.
- [11] F. X. Gaya-Morey, C. Manresa-Yee, and J. M. Buades-Rubio, "Deep learning for computer vision based activity recognition and fall detection of the elderly: a systematic review," *Appl Intell*, vol. 54, pp. 8982–9007, 2024. doi: 10.1007/s10489-024-05645-1.
- [12] H. Liu et al., "A Hybrid Multi-Person Fall Detection Scheme Based on Optimized YOLO and ST-GCN," *Int. J. Interact. Multimed. Artif. Intell.*, vol. 9, no. 4, pp. 28–38, 2024.
- [13] L. Fiorini et al., "Employing the Artificial Intelligence Object Detection Tool YOLOv8 for Pain Assessment Through Facial Expression Analysis," *J Pain Res.*, vol. 17, pp. 4385–4398, Nov. 2024. doi: 10.2147/JPR.S478819.
- [14] A. A. Alanezi et al., "Enhanced Fall Detection Using YOLOv7-W6-Pose for Real-Time Elderly Monitoring," *Electronics*, vol. 16, no. 12, p. 472, Dec. 2024. doi: 10.3390/electronics16120472.
- [15] Ultralytics, "AI in Healthcare: YOLOv8 Models," Ultralytics Blog, Oct. 2024. [Online]. Available: <https://www.ultralytics.com/blog/the-role-of-ai-in-healthcare>
- [16] S. Piramanayagam et al., "Robust Alzheimer's Patient Detection and Tracking for Room Entry Monitoring Using YOLOv8 and Cross Product Analysis," *Comput. Mater. Contin.*, vol. 81, no. 1, pp. 1033-1051, 2025. doi: 10.32604/cmc.2024.051123.
- [17] M. A. Al-khafajiy et al., "Patient Monitoring by Abnormal Human Activity Recognition Based on CNN Architecture," *Electronics*, vol. 9, no. 12, p. 1993, Dec. 2020. doi: 10.3390/electronics9121993.
- [18] M. Z. Islam et al., "Towards Contactless Safety Compliance: Face Mask Monitoring with YOLOv8 Algorithm," *Int. J. Adv. Comput. Sci. Appl.*, vol. 15, no. 12, 2024.
- [19] Ultralytics, "Responsible AI with YOLOv8: Ethics & Security," Ultralytics Blog, Sep. 2024. [Online]. Available: <https://www.ultralytics.com/blog/approaching-responsible-ai-with-ultralytics-yolov8>
- [20] S. Ahmad et al., "Ethical Considerations in the Use of Artificial Intelligence and Machine Learning in Health Care: A Comprehensive Review," *Cureus*, vol. 16, no. 6, p. e62417, Jun. 2024. doi: 10.7759/cureus.62417.
- [21] HITRUST Alliance, "The Ethics of AI in Healthcare," HITRUST Blog, Nov. 2023. [Online]. Available: <https://hitrustalliance.net/blog/the-ethics-of-ai-in-healthcare>
- [22] KMS Healthcare, "5 Ethical Considerations for Implementing Machine Learning in Healthcare," KMS Healthcare Blog, Mar. 2023. [Online]. Available: <https://kms->

- healthcare.com/blog/5-ethical-considerations-for-implementing-machine-learning-in-healthcare/
- [23] D. C. Luvizon, D., Picard, H. Tabia, "Multi-Task Deep Learning for Real-Time 3D Human Pose Estimation and Action Recognition", *textitIEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 8, pp. 2752- 2765, Aug. 2021.
- [24] L. Song, G. Yu, J. Yuan, and Z. Liu, "Human pose estimation and its application to action recognition: A survey," *J. Vis. Commun. Image R.*, vol. 76, p. 103055, 2021.
- [25] R. Mutegeki and D. S. Han, "A CNN-LSTM Approach to Human Activity Recognition", in *textit2020 International Conference on Information and Communication Technology Convergence (ICTC)*, Jeju, Korea (South), 2020, pp. 1136- 1140.
- [26] C. Plizzari, M. Cannici, and M. Matteucci, "Skeleton-based action recognition via spatial and temporal transformer networks," *Comput. Vis. Image Underst.*, vol. 208-209, p. 103219, 2021.
- [27] Z. Wang, Y. Zheng, Y. Yang, M. Zhang, and Y. Li, "Deep Neural Networks in Video Human Action Recognition: A Review," *arXiv preprint arXiv:2305.15692*, 2023.
- [28] J. Shin, N. Hassan, A. S. M. Miah, and S. Nishimura, "A Comprehensive Methodological Survey of Human Activity Recognition Across Divers Data Modalities," *arXiv preprint arXiv:2409.09678*, 2024.
- [29] K. Jannet, M. S. Sunar, M. A. bin As'ari, and M. M. I. Molla, "A Deep Learning Approach to Badminton Player Footwork Detection Based on YOLO Models: A Comparative Study," in *2024 IEEE 8th International Conference on Signal and Image Processing Applications (ICSIPA)*, 2024, pp. 1-6.
- [30] C. Shiranthika, H. Samani, N. Premakumara, C. Shyalika, H.-L. Chiu, and C.-Y. Yang, "Human Activity Recognition Using CNN & LSTM," in *2020 IEEE International Conference on Consumer Electronics-Taiwan (ICCE-Taiwan)*, Taoyuan, Taiwan, 2020, pp. 1-2.
- [31] L. Lu, H. Di, Y. Lu, L. Zhang, and S. Wang, "Spatio-temporal attention mechanisms based model for collective activity recognition," *Signal Process. Image Commun.*, vol. 74, pp. 162-174, 2019.
- [32] L. SaiRamesh, B. Dhanalakshmi, and K. Selvakumar, "Human Activity Recognition Through Images Using a Deep Learning Approach," *Research Square preprint*, 2024. DOI: 10.21203/rs.3.rs-4443695/v1
- [33] S. S. Kulkarni and S. Jadhav, "Insight on Human Activity Recognition Using the Deep Learning Approach," in *2023 International Conference on Emerging Smart Computing and Informatics (ESCI)*, Pune, India, 2023, pp. 1-5.
- [34] A. W. Muhamad and A. A. Mohammed, "Improved RNN Model for Real-Time Human Activity Recognition", *textitIraqi Journal of Science*, vol. 65, no. 9, pp. 5220-5238, 2024.