Intel Unnati Industrial Training Program 2024

Project Report

On

PS 13:Innovative Monitoring System for TeleICU Patients using Video Processing and Deep Learning

By

Team Initiators

Department of Computer Engineering

Marathwada Mitra Mandal's

COLLEGE OF ENGINEERING

Under the guidance of:

Dr. Girija Chiddarwar

Index

Sr. no	Title of chapter	Page no
01	**Introduction**	**3-4**
1.1	Motivation	
1.2	Problem Definition	
1.3	Title	
1.4	Project Domain	
1.5	Problem Statement	
02	**Dataset Description**	**5-6**
2.1	Dataset Source	
2.2	Description	
2.3	Features of Dataset	
03	**Methodology**	**7-8**
3.1	Tool Used	
3.2	Methods used with explanation	
04	**Result And Discussion**	**9-15**

CHAPTER 1: INTRODUCTION

1.1. MOTIVATION:

The "Innovative Monitoring System for TeleICU Patients using Video Processing and Deep Learning" is driven by the imperative to enhance patient care, improve ICU management, and ensure the safety of critically ill patients. By leveraging advanced video processing and deep learning techniques, the project aims to provide real-time monitoring of ICU patients remotely, facilitating timely interventions and reducing the workload on healthcare professionals. This approach not only enhances operational efficiency and resource allocation but also supports clinical research and demonstrates the commitment to integrating cutting-edge technology in healthcare. Ultimately, the project seeks to create a safer, more efficient, and responsive TeleICU environment, improving the overall experience for patients, healthcare staff, and families.

1.2. PROBLEM DEFINITION:

TeleICU, where multiple ICUs are monitored remotely, faces several challenges that affect patient care and resource management. Traditional monitoring techniques, which rely heavily on manual observation and central point processing of data, are slow and prone to errors. The absence of real-time tracking and analytical information exacerbates these problems, making it difficult to ensure patient safety, respond promptly to emergencies, and efficiently manage ICU resources.

Key challenges include:
Patient Monitoring: Inadequate real-time monitoring of patients can lead to delayed responses to critical situations, compromising patient safety and care.
Data Management: Manual data processing and recording increase the risk of errors, leading to inaccurate patient records and inefficient resource utilization.
Emergency Response: The lack of a real-time system to detect and alert medical staff to emergencies makes it difficult to provide timely intervention and support.

Resource Allocation: Inefficient management of ICU resources, such as medical equipment and staff, due to the absence of actionable insights and real-time data, hampers the overall effectiveness of the ICU.

1.3. PROJECT TITLE:

Innovative Monitoring System for TeleICU Patients using Video Processing and Deep Learning

1.4. PROJECT DOMAIN:

DEEP LEARNING & MACHINE LEARNING

1.5. PROJECT STATEMENT:

To develop an "Innovative Monitoring System for TeleICU Patients using Video Processing, Machine Learning, and Deep Learning," we aim to enhance patient care and ICU management through real-time remote monitoring. By leveraging machine learning and deep learning techniques, we ensure immediate responses to patient anomalies, optimizing resource allocation and reducing staff workload. This approach supports clinical research and integrates cutting-edge technology in healthcare. Ultimately, it creates a safer, more efficient, and responsive TeleICU environment.

CHAPTER 2: DATASET DESCRIPTION

2.1. DATASET SOURCE:

In this project, we have meticulously gathered and curated a comprehensive dataset of 2000 images from various reputable sources, aimed at developing an advanced monitoring system for TeleICU patients using video processing and deep learning. These images encompass a wide spectrum of scenarios encountered in TeleICU settings, capturing diverse patient conditions and environmental factors. By leveraging this extensive dataset, our objective is to enhance the accuracy and reliability of remote ICU monitoring through cutting-edge technology. This initiative not only supports the development of robust deep learning models but also fosters innovation in healthcare, facilitating more effective patient care management and clinical decision-making.

2.2. DATASET DESCRIPTION:

The dataset used for developing the "Innovative Monitoring System for TeleICU Patients using Video Processing and Deep Learning" project is a self-built collection augmented with images sourced from reputable Google sites. It comprises a diverse set of 2000 images selected to represent various scenarios encountered in TeleICU environments. These images include different lighting conditions, patient poses, and room configurations, ensuring comprehensive coverage of real-world situations.

Each image is accompanied by patient details and relevant metadata, facilitating the training and evaluation of deep learning models for accurate patient detection and monitoring. This curated dataset forms a crucial foundation for advancing remote ICU monitoring capabilities through state-of-the-art technology integration.

2.3. DATASET FEATURES:

Folder: Name of the folder where the image is located (e.g., State-wise_OLX, google_images, video_images).
Filename: Name of the image file along with its extension (e.g., image0001.jpg, image0002.png).
Path: Absolute path of the image, adaptable to user requirements for flexibility in data management.
Width: Width of the image in pixels.
Height: Height of the image in pixels.
Depth: Depth of the image, typically 3 for RGB color images.
Xmin: Minimum x-coordinate value of relevant features or objects within the image.
Ymin: Minimum y-coordinate value of relevant features or objects within the image.
Xmax: Maximum x-coordinate value of relevant features or objects within the image.
Ymax: Maximum y-coordinate value of relevant features or objects within the image.
x_center=2(x_min+x_max) /2
y_center=(y_min+y_max)/2

CHAPTER 3: Methodology

3.1. TOOLS USED

Languages for Frontend development:

- Python

Tools/Platforms used for development:

- Visual Studio IDE
- Jupyter Notebook
- Google Colab

Libraries used for development:

- Pandas
- Numpy
- Open Cv
- Torch
- Matplotlib
- MatplotLib
- Streamlit
- YOLOv5
- LabelImg

3.2. TECHNIQUES USED

1. Data Preprocessing:

Data preprocessing involves preparing the data for training and inference. This step typically includes: Loading datasets: Reading images and corresponding labels.

Data augmentation: Applying transformations to the images, such as resizing, cropping, and color adjustments, to improve model generalization.

2. Model Training

2.1 Model Architecture:

- The YOLO (You Only Look Once) model architecture is employed due to its efficiency and accuracy in real-time object detection. YOLO's ability to process images in a single forward pass makes it suitable for applications requiring minimal inference time.
- The model is initially trained on the custom dataset to specialize in identifying doctors, nurses, family members, and patients, as well as empty rooms and status of monitors in the ICU

2.2 Training configuration:

- Setting up parameters like learning rate, batch size, and number of epochs. Training loop: Iteratively feeding data into the model, calculating loss, and updating model weights using backpropagation

CHAPTER 4: RESULTS AND DISCUSSION

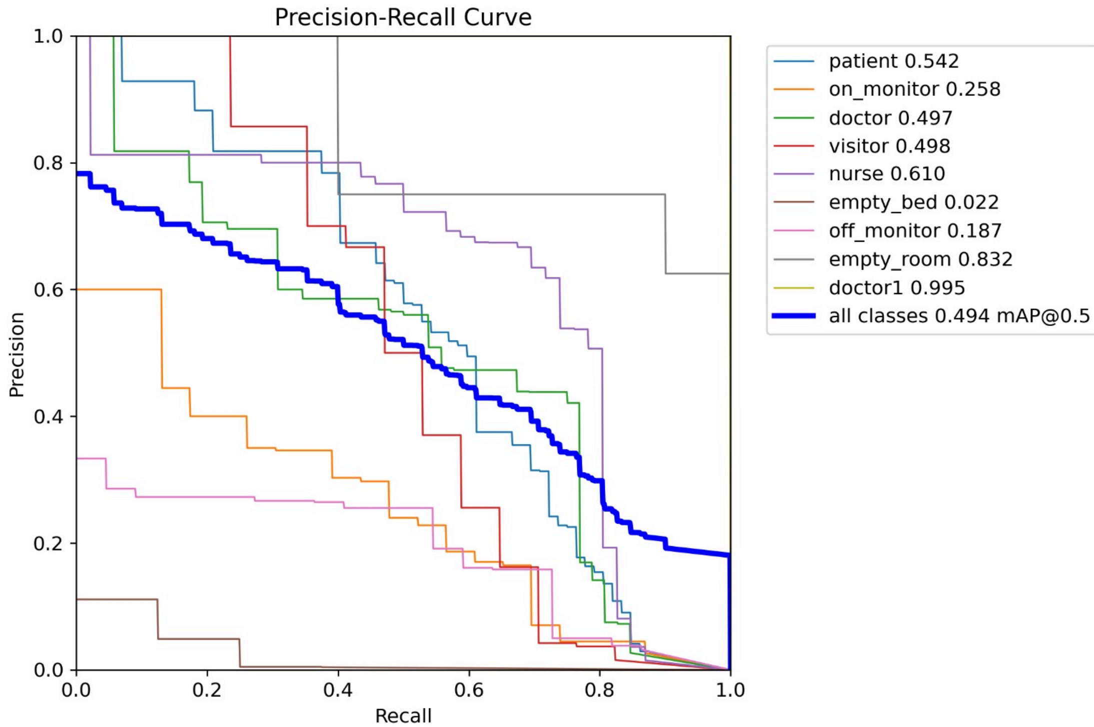

Fig1: Precision-Recall Curve

The Precision-Recall (PR) curve illustrates the trade-off between precision and recall for various classes in the modeThe overall mean Average Precision (mAP) at 0.5 Intersection over Union (IoU) is 0.494, indicating moderate performance across all classes. Individual class performances vary significantly, with "doctor" achieving the highest precision and recall (mAP = 0.995), while "empty_bed" has the lowest (mAP = 0.022). This PR curve is critical for evaluating and improving the model's effectiveness in accurately identifying and classifying different individuals and actions in ICU settings.

Fig 2:Train batch 0 images

The image depicts a sample training batch with annotated bounding boxes for various classes within ICU environments. Each bounding box is color-coded and labeled with a class identifier, indicating the presence of objects such as doctors, nurses, patients, and medical equipment. This visual representation demonstrates the diverse scenarios and entities the model is trained to recognize. The annotations facilitate the model's learning process by providing clear examples of each class, which is crucial for improving detection accuracy and performance in real-world ICU settings. The annotations include complex scenes with multiple objects, enhancing the model's ability to handle intricate and overlapping elements in real-time monitoring.he annotations with class identifiers 0 for patients, 1 for doctors, and 2 for nurses help in training the model to recognize these objects in ICU environments, including complex scenes with overlapping elements. This setup should really enhance the model's ability to handle real-world scenarios effectively.

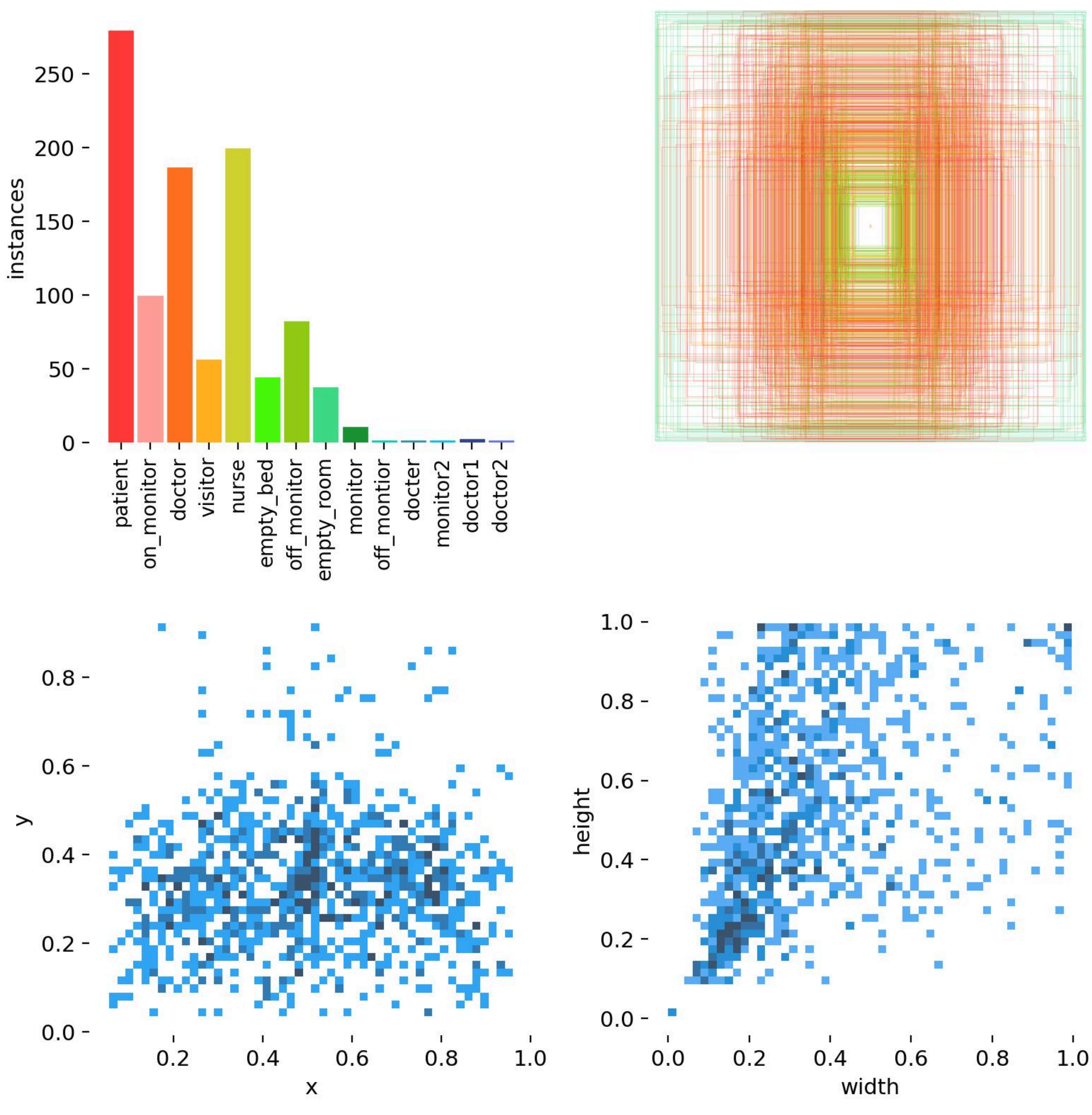

Fig 3: Data Visualization

Top-Left Image (Bar Chart):

- **Title**: **Instances of Detected Classes**
- Description: This bar chart represents the frequency of different detected classes in the dataset. The x-axis lists various classes such as "patient," "on_monitor," "doctor," "visitor," "nurse," "empty_bed," "off_monitor," "empty_room," and others. The y-axis shows the number of instances detected for each class.
- Observations: The "patient" class has the highest number of instances, followed by "on_monitor," "doctor," and "visitor." Classes like "monitor1," "doctor1," and "doctor2" have very few instances.

Top-Right Image (Bounding Box Visualization):

- **Title: Bounding Box Visualization**
- Description: This image shows the bounding boxes for detected objects in various frames overlaid on each other. Each bounding box is colored differently to represent different classes.
- Observations: The density of bounding boxes in the center indicates that many objects are detected around this area, which may be the focal point in most frames.

Bottom-Left Image (Scatter Plot for x and y Coordinates):

- **Title: Scatter Plot of Bounding Box Coordinates**
- Description: This scatter plot visualizes the distribution of the bounding box coordinates (x, y) for detected objects. The x-axis represents the x-coordinate, and the y-axis represents the y-coordinate of the bounding boxes.
- Observations: The scatter plot shows a relatively even distribution of bounding box coordinates across the frame, with a slight concentration around the center.

Bottom-Right Image (Scatter Plot for Width and Height):

- **Title: Scatter Plot of Bounding Box Dimensions**
- Description: This scatter plot shows the distribution of bounding box dimensions (width, height) for detected objects. The x-axis represents the width, and the y-axis represents the height of the bounding boxes.
- Observations: The plot indicates a wide range of bounding box sizes, with a notable concentration of smaller bounding boxes (width and height around 0.2-0.4).

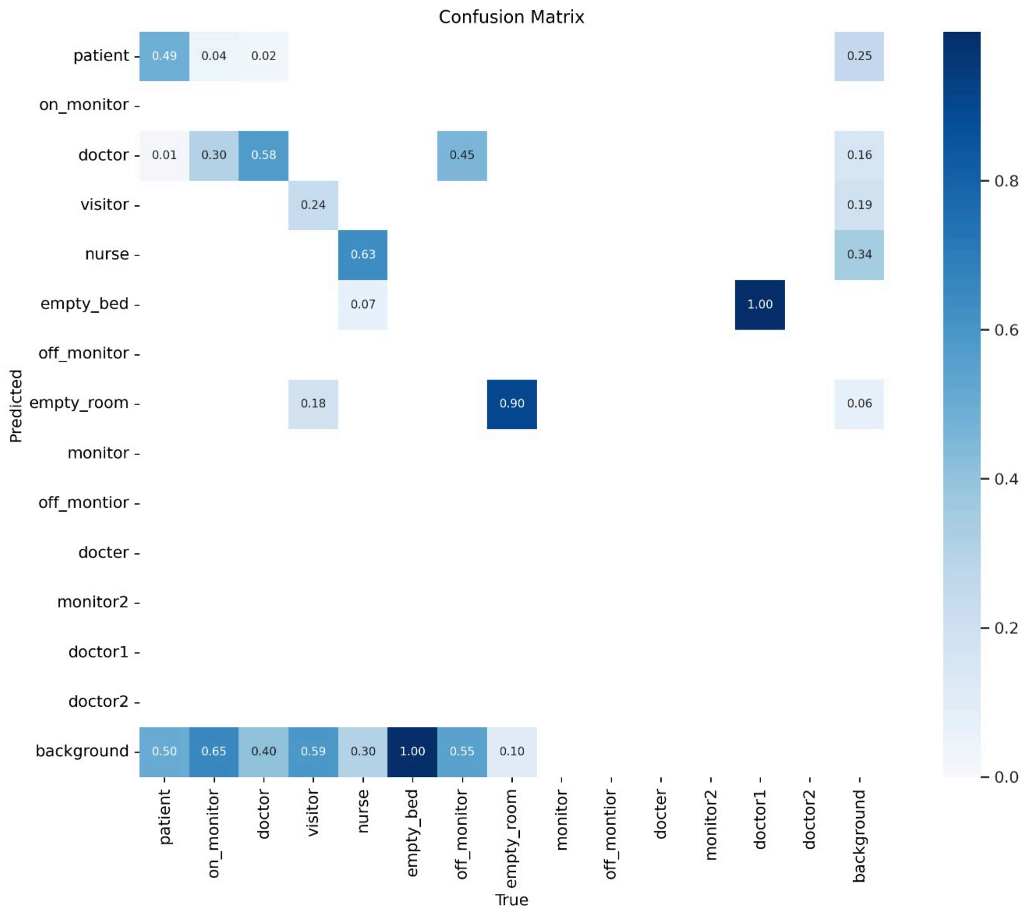

Fig 4: Confusion matrix

This image is a confusion matrix visualizing the performance of the object detection model used in the TeleICU monitoring system. The matrix compares the predicted classes against the true classes, providing insight into the model's accuracy and areas of misclassification**.**

Description:

- True Labels (Y-axis): Actual classes: "patient," "on_monitor," "doctor," "visitor," "nurse," "empty_bed," "off_monitor," "empty_room," "monitor," "off_monitor," "doctor," "monitor2," "doctor1," "doctor2," and "background."
- Predicted Labels (X-axis): Classes predicted by the model.
- Values: Proportion of predictions for each true class falling into each predicted class.

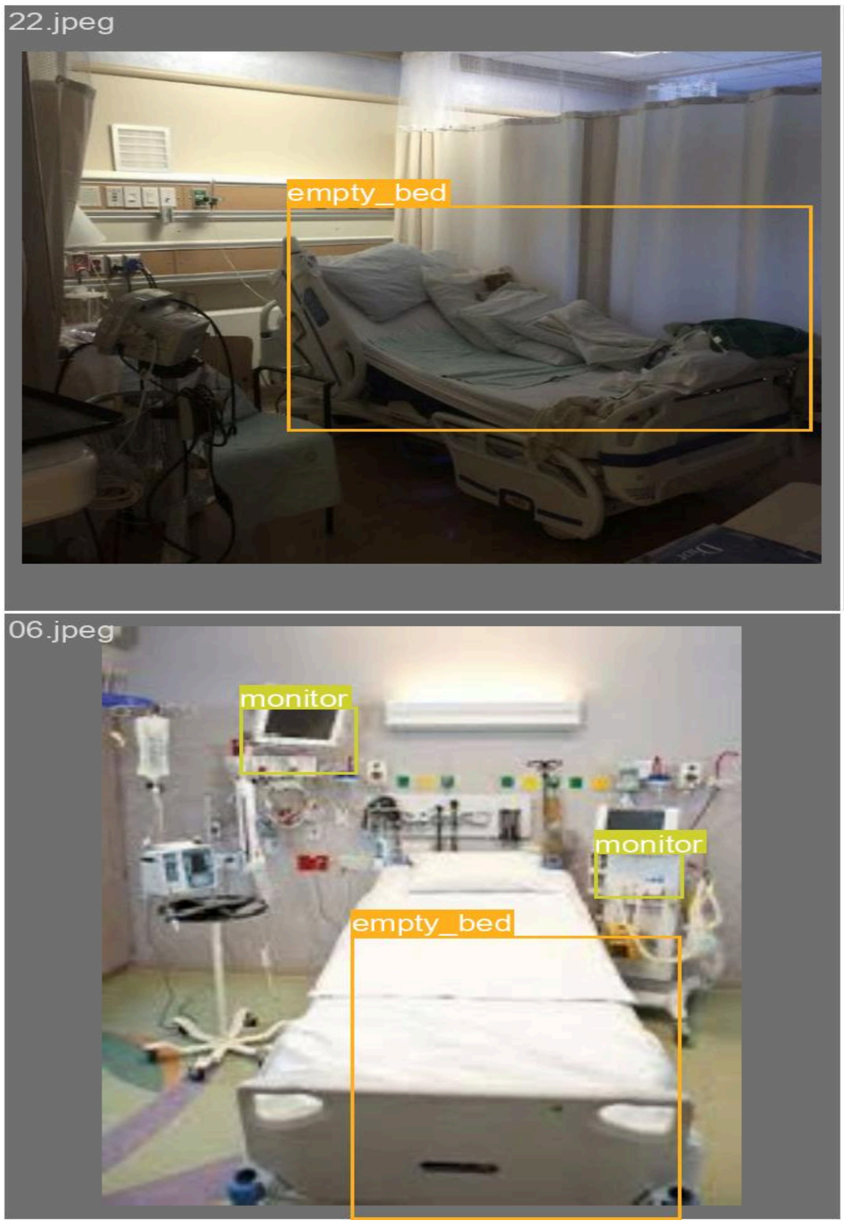

Fig 5: Predicted Images

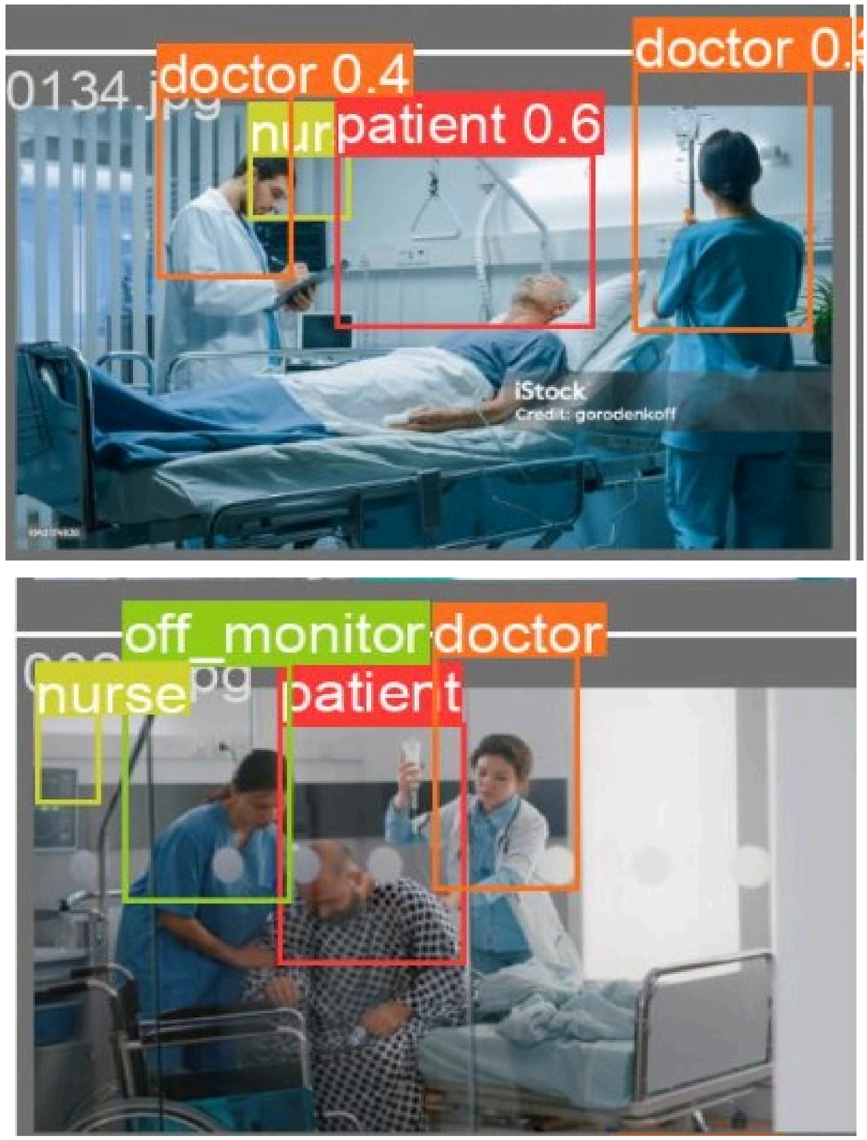

Fig 6: Predicted Images