OBJECT DETECTION AND CLASSIFICATION BASED ON VARIOUS DEEP LEARNING TECHNIQUES

Author : Mr. Vaibhav Narkhede

PREFACE

In today's world, video surveillance systems generate a massive influx of data, often overwhelming human operators. This deluge of information makes it challenging to effectively monitor and analyze events in real-time, hindering proactive intervention and efficient post-event investigation. Artificial intelligence (AI) offers a powerful paradigm shift in addressing these limitations. By endowing surveillance systems with the ability to automatically perceive, reason, and learn from visual data, AI promises to unlock unprecedented levels of efficiency, accuracy, and actionable insights. This exploration delves into the crucial need for an efficient mechanism to harness the transformative potential of AI within video surveillance. Efficiency, in this context, encompasses several key aspects: optimized resource utilization (computation, storage, bandwidth), rapid and accurate processing of video streams, scalability to handle large deployments, and seamless integration with existing infrastructure. Without a robust and efficient underlying mechanism, the promise of AI-powered surveillance risks being bottlenecked by practical limitations. The subsequent discussion will highlight the challenges and opportunities in building such an efficient mechanism, exploring various AI techniques like deep learning, computer vision algorithms, and edge computing strategies. It will also touch upon the importance of data management, model optimization, and system architecture in achieving truly efficient and impactful AI-driven video surveillance applications. The goal is to pave the way for surveillance systems that are not merely passive recording devices but intelligent sentinels capable of proactively safeguarding our communities and assets.

It is my sincere hope that this book will inspire you, provide clarity on complex topics, and serve as a trusted resource in your personal and professional development. While writing this book I personally thank to my parents who always support me, my wife and child who has given me confidence to complete this book. I also thank to my respected guide Dr. Jawandiya Sir without which this book is not possible. And last but not the least my friends who always encourage me to gain new knowledge in my field ------Vaibhav

Research Methodology Used for Object Detection & Identification of Unusual Behavior

This chapter of this research work provides the detailed information about the methodology used for developing this framework where we have defined different phases for the object detection, object classification and recognition along with the identification of the unusual behavior.

1. Research Methodology

Several crucial steps must be taken in order to develop a framework to develop the system for better security. The framework's high-level description is given below:

A. Phase 1: Object Detection and Classification with Live Camera Feed Our project's first phase is to develop a system that can identify and categorize items in real time using a live video stream. The core objective is to accurately recognize and classify objects into two basic categories: cars and humans.

B.

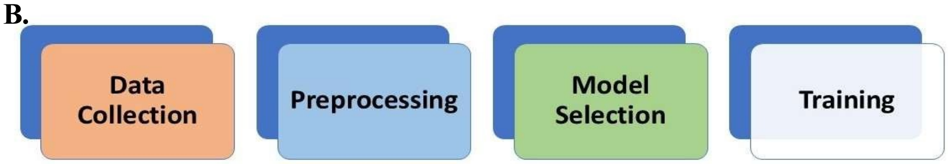

Fig 1: Components of Phase 1

- **Data Collection**: Compile a varied collection comprising pictures of people and other kinds of cars taken in various settings, lighting situations, and perspectives.
- **Preprocessing**: Preprocess the collected data to ensure uniformity in size, color, and orientation. Augment the dataset if necessary to improve model generalization.
- **Model Selection**: Choose a suitable deep learning model architecture for item recognition and classification, such as SSD (Single Shot MultiBox Detector), YOLO (You Only Look Once), or Faster R-CNN (Region-based Convolutional Neural Network).
- **Training**: Train the selected model on the prepared dataset using a powerful

GPU-accelerated infrastructure to expedite the training process.

C. Phase 2: Anomaly Detection for Humans and Vehicles

In Phase 2, our focus shifts to identifying unusual behaviors exhibited by humans and vehicles detected in Phase 1. The goal is to create an anomaly detection system capable of flagging suspicious activities or behaviors in real-time.

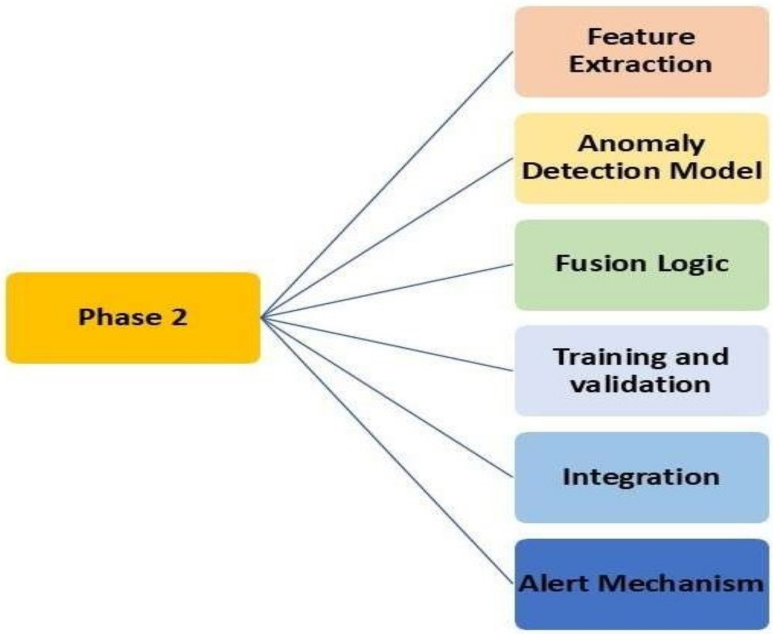

Fig 2: Components of Phase 2

- **Feature Extraction**: Gather pertinent information from the objects that have been spotted, such as their direction, speed, movement patterns, and proximity to specific locations.

- **Anomaly Detection Model**: Develop an anomaly detection model using techniques like supervised learning (e.g., classification algorithms trained on labelled anomalous behavior).or unsupervised learning (e.g., clustering algorithms such as k- means) or

- **Fusion Logic**: YOLO and SSD was used for the fusion by implementing the intersection over union (IoU) and Average confidence score for final detection of the unusual behavior.

- **Training and Validation**: Labelled data with both normal and abnormal behaviors can be used to train the anomaly detection model. Make sure the model can effectively identify abnormalities while reducing false positives by validating its performance using the right metrics.

- **Integration**: Integrate the anomaly detection module with the object detection and classification system developed in Phase 1, enabling seamless analysis of live camera feeds.
- **Alert Mechanism**: Implement an alert mechanism to notify relevant stakeholders (e.g., security personnel, system administrators) in real-time when anomalous behavior is detected. This could involve triggering alarms, sending notifications to mobile devices, or integrating with existing security systems.

By completing both phases of the project, we will have a comprehensive AI powered surveillance system capable of not only detecting and classifying humans and vehicles in real-time but also identifying suspicious or unusual behaviors, thus enhancing security and situational awareness in the monitored environment. The Methodology for this system will make use of different algorithms in the area of Artificial Intelligence which will improve performance of Video Surveillance System in regards with the accuracy and efficiency.

2 Fusion Technique

In order to increase the precision and dependability of identifying anomalous behaviors in crowds, fusion systems for abnormality detection combine data from several sources or employ several approaches. These methods combine various sensors, algorithms, or modalities (including audio, video, thermal data, and even social media analysis) to identify anomalous activity in a crowd more thoroughly and accurately. Fusion procedures are effective because they reduce the drawbacks of individual approaches by utilizing the advantages of several data sources and methodologies.

Here are some key fusion techniques used for abnormality detection in crowds:

3 Data-Level Fusion (Sensor Fusion)

Fusion at the Data level combines data from multiple sensors to create a single, cohesive dataset before any analysis. The goal is to use complementary data to provide richer information about the environment or crowd. Video cameras might capture visual data while thermal sensors detect heat signatures, and both data streams are combined for better detection accuracy, especially in low-light or challenging conditions. In public safety, combining visual surveillance with thermal imaging helps detect abnormalities such as overcrowding in low visibility situations (e.g., at night or in smoke-filled environments).

- **Weighted Averaging**: Combines multiple sensor readings into a single dataset based on the relevance or confidence level of each source.
- **Kalman Filtering**: Predicts and combines sensor data over time for more reliable results.

4 Feature-Level Fusion

Features are taken independently from several data sources and then integrated to create a single feature vector, which is subsequently examined for anomalies in feature-level fusion. This is a middle-ground fusion technique that merges relevant information from each modality or sensor. Combining crowd density features from video data with acoustic features like noise levels to detect crowd panic or abnormal behaviour. Used in large event surveillance, where video footage and sound recordings are analysed together to detect crowd unrest (e.g., fights or stampedes).

- **Concatenation of Features**: creates a single, comprehensive feature vector for analysis by combining feature vectors from several modalities.
- **PCA (Principal Component Analysis)**: decreases the fused feature set's dimensionality, which facilitates anomaly detection.

5 Decision-Level Fusion

In this type of fusion, individual decisions from separate anomaly detection systems or algorithms are combined to make a final decision. Each system operates independently, and its output (normal or abnormal) is then fused to produce a more robust decision. Outputs from crowd detection systems using video analysis, audio monitoring, and social media data are independently analysed and their decisions are fused using techniques like voting or confidence weighting. In smart city monitoring, decision-level fusion could be applied by using multiple surveillance systems to detect anomalies in large gatherings, like protests or unexpected crowd formations.

- **Majority Voting**: The outcome that most systems agree on is used to make the ultimate decision.
- **Dempster-Shafer Theory**: A probabilistic framework for combining

decisions with varying degrees of certainty from different sources.

6 Hybrid Fusion (Combination of Levels)

Hybrid fusion involves combining data, features, and decisions at multiple levels to achieve more accurate abnormality detection. This approach offers flexibility in selecting the most effective fusion strategy at each stage of analysis. Thermal data and video are first combined at the data level to improve image clarity, then feature- level fusion is performed to extract crowd movement patterns, followed by decision-level fusion to finalize the anomaly detection. Used in highly sensitive environments such as airport security, where multiple types of data (e.g., visual, thermal, audio) are analysed together to detect unusual behaviour in crowds.

- **Multi-Sensor Data Fusion Architecture**: Combines multiple fusion strategies into a layered structure that integrates data, feature, and decision levels.

7 Multimodal Fusion

The combination of various sensory data, including text, audio, and video, to identify anomalies in a crowd is known as multimodal fusion. This method makes use of the advantages of several kinds of data , providing richer information and better detection of complex anomalies. Using video surveillance for real-time crowd monitoring, coupled with audio analysis for detecting screams or panic, and incorporating social media feeds to predict crowd unrest. In large public gatherings,

multimodal fusion helps security teams monitor for abnormal crowd behaviours, such as a sudden increase in noise levels, coupled with unusual crowd movement.

- **Multimodal Deep Learning**: Neural networks designed to handle multiple data modalities (e.g., video, audio, sensor) simultaneously, learning from the interactions between them.
- **Joint Representation Learning**: A method where multiple modalities are mapped into a shared feature space, allowing the system to correlate different data types effectively.

8 Context-Aware Fusion

Context-aware fusion takes into account additional contextual information when combining data from multiple sources. This technique improves anomaly detection by considering the environment and external factors that could influence the crowd's behaviour. In an outdoor event, the fusion system may combine video footage of the crowd with weather data (e.g., temperature, wind speed) and contextual information (e.g., event type) to adjust the anomaly detection threshold. This approach is valuable for dynamic environments, such as disaster management or emergency response, where environmental conditions can significantly impact crowd behaviour.

- **Bayesian Networks**: Use probabilistic models to incorporate contextual information into the fusion process, making anomaly detection more sensitive to the situation at hand.
- **Markov Models**: Track the evolution of crowd behaviour over time, integrating context to detect anomalies based on transitions between normal and abnormal states.

9 Time-Series and Temporal Fusion

Temporal fusion techniques are designed to capture and fuse data over time, detecting anomalies based on changes in crowd behaviour across different time points. This is useful for understanding how the crowd evolves and whether any gradual shifts indicate potential threats. A system that monitors a crowd's movement speed and direction over time, identifying sudden changes that may indicate a panic or an unexpected rush. Used in public transportation hubs or stadiums to detect real-time abnormalities, such as sudden surges or bottlenecks in pedestrian flow.

- **Recurrent Neural Networks (RNNs)**: Analyse sequential data, detecting temporal anomalies in patterns such as crowd movement or density over time.
- **Kalman Filters**: Used to smooth and predict future crowd behaviour by fusing past and present data.

A Mechanism for Application in Video Surveillance

1 System Design

There are three primary parts to the deep convolutional architecture for detecting anomalous activity in a smart surveillance system.s:

- Human identification & differentiation.
- A posture classifying module.
- A module for detecting and identifying anomalous behaviors.
- Fusion processing of YOLO and SSD with DBSCAN.

The proposed methodology for crowd behavior detection, illustrated in Figure 4.1, distinguishes between suspicious and normal behavior by fusing YOLO and DBSCAN models on input videos. Video preprocessing involves frame extraction and dataset cleaning. Subsets of the dataset are separated for testing, validation, and training. Based on a number of criteria, crowd behavior is then classified as either suspicious or normal using a classification technique.

2 System Flow

A normal flow for the system is as follows:

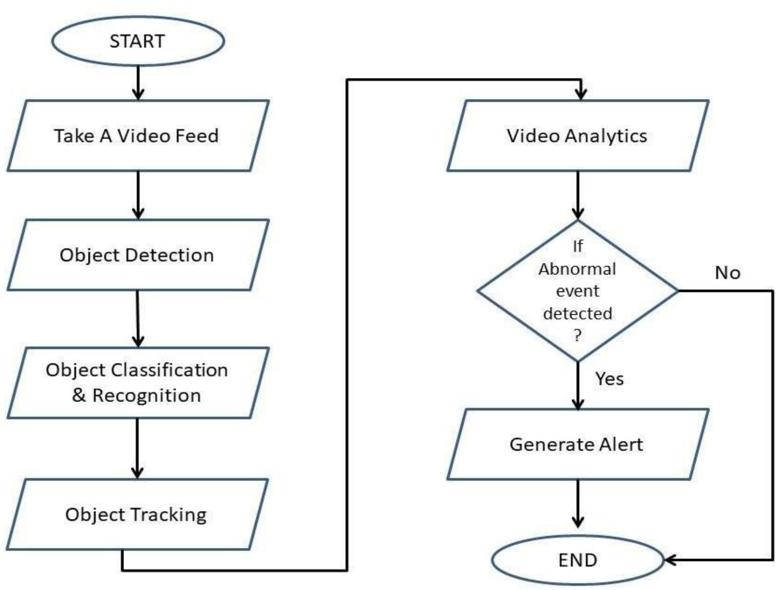

Fig 3 Normal flow diagram for this system

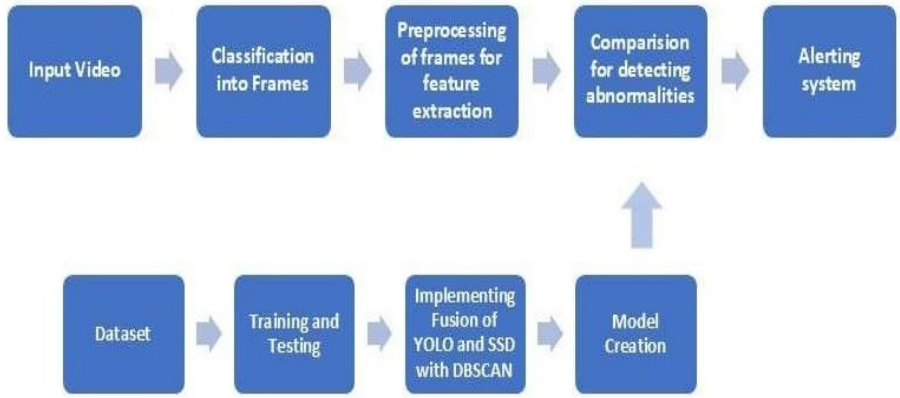

Fig.4 Diagrammatic representation of the proposed system

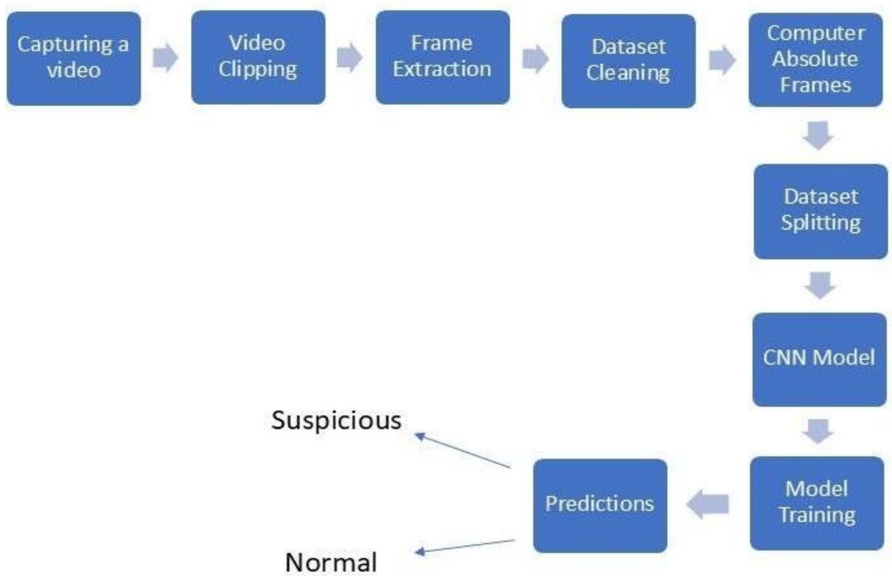

Fig. 5 Framewise flow of the system

A growing number of individuals are leading increasingly inactive lives. Studies suggest that many people spend over half of their day seated, with those experiencing limited mobility potentially sitting for up to 85% of their waking hours.[26] This trend is often exacerbated by muscle and bone deterioration, which can result from aging or neurodegenerative conditions.[27] Consequently, individuals with reduced mobility frequently rely on wheelchairs to maintain their independence. While wheelchairs significantly improve mobility, their use inherently increases sedentary behavior, potentially leading to both physical and psychological challenges associated with prolonged sitting.[28]

3. Flow of the system for training videos Step 1:

Train the videos from the dataset.

Step 2:

Capturing the video Cap=cv2.videocapture()

Step 3:

Converting the trained videos into frames in jpg format

Step 4:

Captured frames will be stored automatically in the folder.

Step 5:

When searching for the image, if it is available, it will perform the process of resizing and finding the inter-area.

Image=cv2.resize(image,(227,227),interpolation=cv2.INTER_AREA) Step 6:

After resizing, it will convert the original image into a grayscale image

Gray=0.2989*image[:,:,0]+0.5870*[:,:,1]+0.1140*image[:,:,2]

Step 7:

Resized Stored images will be calculated by using Mean and Std. Dev

Step 8:

And the result will be saved as model training using YOLO ,SSD and DBSCAN

4.1.1 Flow of the system for testing videos Step 1:

Load the saved model.

Step 2:

Capturing the test video Cap=cv2.videocapture()

Step 3:

Captured frames will be opened, the empty array will be generated.

Step 4:

Read the captured frame. Step 4: When searching for the image, if it is available, it will perform the process for resizing (600 x 700) and find the inter-area.

Image=cv2.resize(image,(227,227),interpolation=cv2.INTER_AREA) Step 5:

After resizing, it will convert the original image into a grayscale image

Gray=0.2989*image[:,:,0]+0.5870*[:,:,1]+0.1140*image[:,:,2]

Step 6:

Resized Stored images will be calculated by using gray Mean and Std. Dev

Step 7.:

The gray result will be stored in an array.

Step 8:

The dumped array image is resized and expanded.

Step 9:

Output has been predicted with the help of a dump array image.

Step 10:

Output is measured with the resized dump image array with the loss value of 0.0068 to find the events.

4 YOLO5

YOLOv5 is a state-of-the-art object detection model designed for speed and accuracy. Its architecture can be broken down into three main parts: the Backbone, the Neck, and the Head. Here's a straightforward explanation of each component and how they work together to detect objects in images and video.

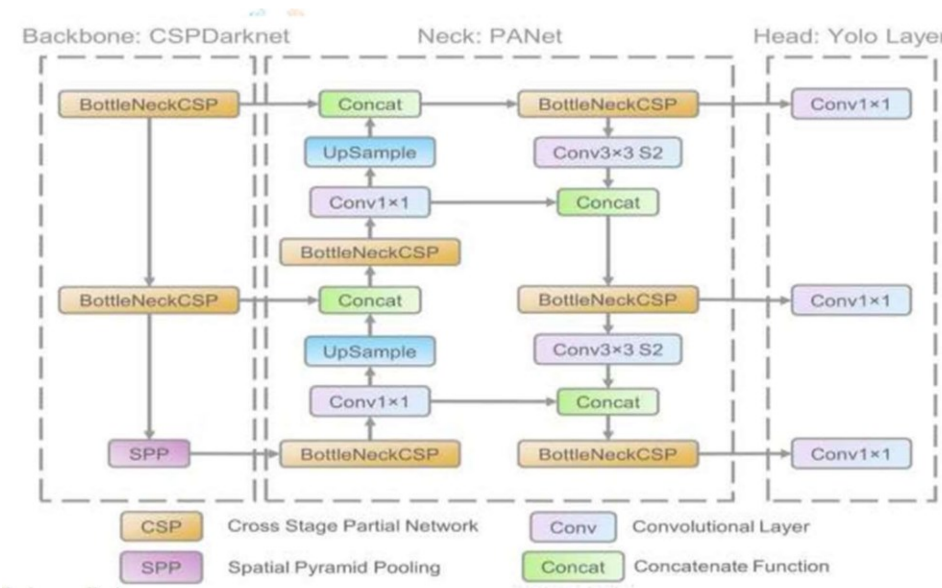

Fig. 6 YOLO5 Architecture

5 Backbone Function:

Features are extracted from the input image by the backbone. These characteristics include patterns and specifics that aid the model in identifying the objects and their potential locations. Convolutional Layers generate feature maps by applying filters to the input image. Different elements of the image, including edges, textures, and colors, are captured by each layer. More intricate and abstract details are revealed as the image moves through the levels. This procedure is similar to how the human brain processes visual information at several levels to identify shapes and things.

6 Neck Function:

The Neck further processes the features extracted by the Backbone to make them more useful for detecting objects of different sizes. It combines and refines these features to improve detection accuracy. Feature Pyramid Network (FPN): This component helps the model recognize objects at multiple scalesIt uses the feature maps from the various Backbone phases and processes them to ensure that both small and large objects can be detected effectively. Path Aggregation Network (PAN): This helps in enhancing feature representation by combining features from different layers. This procedure enhances the model's comprehension of context and sharpens object boundaries. The Neck acts like a smart organizer, taking the detailed information from the Backbone and sorting it in a way that makes it easier for the model to spot objects, no matter their size.

7 Architecture of SSD

SSD is composed of two main components:Base Network (Backbone):
A pre-trained convolutional neural network (such as VGG16 or ResNet) forms the initial component of SSD. It functions as a feature extractor, producing high- level feature maps from the input image.

- **Additional Feature Layers:**

To produce feature maps of various scales, a number of convolutional layers are layered on top of the underlying network. Objects with varying sizes and aspect ratios can be detected using these layers.

Workflow of SSD

- **Input Image:**

The input image (usually 300x300 or 512x512) is passed through the backbone network, generating feature maps.

- **Convolutional Feature Maps:**

SSD utilizes multiple-layer feature maps (e.g., conv4_3, conv7, and further additional layers). These layers provide **multi-scale detection** (e.g., smaller layers detect small objects, deeper layers detect larger objects).

- **Anchor Boxes (Default Boxes):**

SSD specifies a collection of anchor boxes with various aspect ratios and scales at every point in the feature maps. These boxes are used to predict the presence of objects. For example: 4:3, 1:1, 2:1 aspect ratios, etc. SSD places these default boxes at **every spatial position** of the feature maps.

- **Class Scores and Bounding Box Regression:**
- SSD forecasts each anchor box's:
  - **Class scores**: Probability of each object class (e.g., cat, car, person, etc.).
  - **Bounding box offsets**: Adjustments to improve the accuracy of the anchor box’s fit to the actual object.
- **Non-Maximum Suppression (NMS):**

Multiple predictions for the same object can appear. SSD selects the box with the highest confidence in order to remove redundant detections using Non-Maximum Suppression (NMS).

- **Output:**

The final output is a set of detected objects with:

- **Class labels** (e.g., cat, person)
- **Bounding box coordinates** (x, y, width, height)
- **Confidence scores** (how likely the detection is correct)

Advantages of SSD Speed:

SSD can process images in real time, achieving high frame rates suitable for applications like video processing.

Multi-scale Feature Detection:

Using different feature maps enables the detection of both small and large objects.

Single-shot Detection:

Unlike Faster R-CNN, SSD performs classification and object localization in a single forward pass.

Challenges:

- **Performance on Small Objects:**

  Compared to two-stage detectors like Faster R-CNN, SSD typically has trouble handling small objects.

- **Trade-off between Speed and Accuracy:**

Faster variants of SSD (like SSD300) may not be as accurate as slower models like Faster R-CNN.

Variants of SSD:

- **SSD300:** Processes 300x300 images for faster predictions.
- **SSD512:** Processes 512x512 images with higher accuracy, though slower.

- **MobileNet-SSD:** An SSD that is lighter and better suited for embedded and mobile systems.

8 Proposed Methodology to achieve the Objectives

Data Gathering and Preparation

A dataset of surveillance footage from numerous environments such as streets, buildings, parking lots, and public areas is collected. Ensure the dataset includes examples of both normal and anomalous activities. Data Labels are annotated in collected footage with bounding boxes for the objects and activities of interest. This helps model learn to identify and classify different types of objects and behaviors. Specific annotations for anomalies are included, marking anomalous events. Video frames are resized to a consistent size suitable for YOLOv5 input (e.g., 640x640 pixels). Pixel values are normalized to ensure consistency and improve model performance. Data is augmented with techniques like flipping, rotation, and color adjustments to increase robustness.

Model Training

Fine-tune the YOLOv5 and DBSCAN model on the annotated surveillance dataset. This includes adjustment of model weights to better recognize objects and activities specific to the surveillance context. A combination of supervised learning (with labelled normal and anomalous events) is used to enhance detection accuracy. Model is optimized using a loss function that combines classification loss (to correctly identify object classes), localization loss (to accurately predict bounding boxes), and confidence loss (to ensure the detected objects are likely to be correct). Monitor training progress using metrics like precision, recall, and mean average precision.

9 Anomaly Detection

The anomaly detection step of our proposed methodology is essential since it ensures that observed items are classified. During this stage, the model is fed the patterns discovered in visual images or real-time frames during the analysis phase. The next step is to classify the discovered objects into the highlighted categories. Security camera live video feeds were processed using DBSCAN and an improved YOLOv5 model. Each frame's contents are detected and categorized by the model, which also assigns a label and a bounding box to each object.

4.1.2 Requirement specification for the system to execute

This study used a computer hardware specification that included an NVIDIA Quadro RTX 5000, a Jupyter notebook or Visual Studio Code, a Core i5 CPU, and 16GB of RAM. Libraries such as TensorFlow, NumPy, Pillow, Seaborn, and OpenCV v3.3.0 were utilized to set up the development environment. The Darknet-53 framework and CUDA 10.2 were also built with GPU processing to expedite model training. Pretrained weights were imported from the YOLOv5 model and DBSCAN for initialization.

Implementation of System

This chapter deals with the complete procedure for implementation of designed system using python , stating how to install the python ,installing the desired libraries , loading the pretrained models for object detection, recognition and classification. IT consist of snapshots of the source code for our project also.

Implementation process continues with following basic steps:

- **Training and Testing Process**
- **Modelling Process**
- **Detection process**
- **Visualization process**

1 PROCESS FOR SSD AND YOLO

Libraries used

```
F: > Phd_Projects > Vaibhav_Phd > Vaibhav_Sir_New > process_ssd.py > ...
import tkinter as tk
from tkinter import filedialog, messagebox
import cv2
import torch
import os
import numpy as np
from sklearn.cluster import DBSCAN
from sklearn.preprocessing import StandardScaler
from sklearn.metrics import confusion_matrix, precision_score, recall_score, f1_score, precision_recall_curve
import warnings
import matplotlib.pyplot as plt
import seaborn as sns
from matplotlib.backends.backend_tkagg import FigureCanvasTkAgg
```

Fig 7 Libraries used in the system

SSD Model loading

```python
# Load the SSD MobileNet model
from torchvision.models.detection import ssdlite320_mobilenet_v3_large

model = ssdlite320_mobilenet_v3_large(weights="DEFAULT")  # Updated to use weights
model.eval()  # Set the model to evaluation mode


def process_ssd_data(video_input_path, video_output_dir):
    video_output_path = os.path.join(video_output_dir, 'output_video_2.avi')

    # Define categories to detect (0: person, 2: car, 7: truck)
    target_classes = [0, 2, 7]  # Adjust based on your needs

    # Initialize video capture and writer
    cap = cv2.VideoCapture(video_input_path)
    if not cap.isOpened():
        messagebox.showerror("Error", f"Could not open video file {video_input_path}")
        return

    frame_width = int(cap.get(cv2.CAP_PROP_FRAME_WIDTH))
    frame_height = int(cap.get(cv2.CAP_PROP_FRAME_HEIGHT))
    fourcc = cv2.VideoWriter_fourcc(*'MJPG')  # You can try other codecs like 'MP4V'
    out = cv2.VideoWriter(video_output_path, fourcc, 20.0, (frame_width, frame_height))
```

Fig. 8 SSD model loading phase for processing

Object detection phase in SSD

```python
# Function to detect anomalies using DBSCAN
def detect_anomaly_with_dbscan(feature_vectors, eps=3.0, min_samples=5):
    scaler = StandardScaler()
    scaled_features = scaler.fit_transform(feature_vectors)
    dbscan = DBSCAN(eps=eps, min_samples=min_samples)
    labels = dbscan.fit_predict(scaled_features)

    # Identify points classified as noise (-1)
    anomalies = np.where(labels == -1)[0]
    return anomalies, labels

# Feature vectors to be extracted for each detected object
feature_vectors = []
true_labels = []  # To store true labels (0 for normal, 1 for anomaly)
predicted_labels = []  # To store predicted labels (0 for normal, 1 for anomaly)

# Process the video
while cap.isOpened():
    ret, frame = cap.read()
    if not ret:
        break

    # Convert the frame to a tensor
    frame_tensor = torch.from_numpy(frame).permute(2, 0, 1).float() / 255.0  # Convert to tensor and normalize
    frame_tensor = frame_tensor.unsqueeze(0)  # Add batch dimension

    # Object detection
    with torch.no_grad():
        detections = model(frame_tensor)  # Run the model
```

```python
# Object detection
with torch.no_grad():
    detections = model(frame_tensor)  # Run the model

# Process detections
for i, (box, label, score) in enumerate(zip(detections[0]['boxes'], detections[0]['labels'], detections[0]['scores'])):
    if int(label) in target_classes and score > 0.5:
        # Get bounding box coordinates
        x1, y1, x2, y2 = box.int().tolist()
        # Crop the detected region for feature extraction
        crop = frame[y1:y2, x1:x2]
        crop_resized = cv2.resize(crop, (64, 64))  # Resize to fit the input size
        crop_flattened = crop_resized.flatten()  # Flatten the image to a 1D vector

        # Add the feature vector to the list
        feature_vectors.append(crop_flattened)

        # Assume the first class (0) is normal and others are anomalies (for demonstration)
        true_labels.append(1 if int(label) != 0 else 0)
```

Fig. 9 Object detection phase in SSD

Anomaly detection in SSD using DBSCAN

```python
# Perform DBSCAN anomaly detection
if len(feature_vectors) > 0:
    anomalies, labels = detect_anomaly_with_dbscan(np.array(feature_vectors), eps=2.5, min_samples=4)

    for i, (box, label, score) in enumerate(zip(detections[0]['boxes'], detections[0]['labels'], detections[0]['scores'])):
        if int(label) in target_classes and score > 0.5:
            x1, y1, x2, y2 = box.int().tolist()
            color = (0, 255, 0)  # Default color is green

            # If the current detection is identified as an anomaly
            if i in anomalies:
                color = (0, 0, 255)  # Color changes to red
                predicted_labels.append(1)  # Mark as anomaly
            else:
                predicted_labels.append(0)  # Mark as normal

            # Draw bounding box and label
            label_text = f"{int(label)}: {'Anomaly' if i in anomalies else 'Normal'}"
            cv2.rectangle(frame, (x1, y1), (x2, y2), color, 2)
            cv2.putText(frame, label_text, (x1, y1 - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.9, color, 2)
```

Fig. 10 Anomaly detection in SSD

YOLO model loading phase

```python
def process_data(video_input_path, video_output_dir):
    # Load the YOLOv5 model
    model_path = os.path.join('models', 'yolov5s.pt')
    model = torch.hub.load('ultralytics/yolov5', 'custom', path=model_path)

    # Define categories to detect (0: person, 2: car, 7: truck)
    target_classes = [0, 2, 7]

    # Initialize video capture and writer
    cap = cv2.VideoCapture(video_input_path)
    if not cap.isOpened():
        messagebox.showerror("Error", f"Could not open video file {video_input_path}")
        return

    frame_width = int(cap.get(cv2.CAP_PROP_FRAME_WIDTH))
    frame_height = int(cap.get(cv2.CAP_PROP_FRAME_HEIGHT))
    fourcc = cv2.VideoWriter_fourcc(*'MJPG')  # You can try other codecs like 'MP4V'
    video_output_path = os.path.join(video_output_dir, 'output_video_1.avi')
    out = cv2.VideoWriter(video_output_path, fourcc, 20.0, (frame_width, frame_height))
```

Fig. 11 YOLO model loading phase for processing

Object detection phase in YOLO

```python
# Function to detect anomalies using DBSCAN
def detect_anomaly_with_dbscan(feature_vectors, eps=3.0, min_samples=5):
    scaler = StandardScaler()
    scaled_features = scaler.fit_transform(feature_vectors)
    dbscan = DBSCAN(eps=eps, min_samples=min_samples)
    labels = dbscan.fit_predict(scaled_features)

    # Identify points classified as noise (-1)
    anomalies = np.where(labels == -1)[0]
    return anomalies, labels

# Feature vectors to be extracted for each detected object
feature_vectors = []
true_labels = []  # To store true labels (e.g., 0 for normal, 1 for anomaly)
predicted_labels = []  # To store predicted labels (e.g., 0 for normal, 1 for anomaly)
```

Fig. 12 Object detection phase in YOLO

Anomaly detection in YOLO using DBSCAN

```python
# Perform DBSCAN anomaly detection
if len(feature_vectors) > 0:
    anomalies, labels = detect_anomaly_with_dbscan(np.array(feature_vectors), eps=2.5, min_samples=4)

    for i, detection in enumerate(detections):
        x1, y1, x2, y2, conf, cls = detection
        if int(cls) in target_classes and conf > 0.5:
            color = (0, 255, 0)  # Default color is green

            # If the current detection is identified as an anomaly
            if i in anomalies:
                color = (0, 0, 255)  # Color changes to red
                predicted_labels.append(1)  # Mark as anomaly
            else:
                predicted_labels.append(0)  # Mark as normal

            # Draw bounding box and label
            label = f"{model.names[int(cls)]}: {'Anomaly' if i in anomalies else 'Normal'}"
            cv2.rectangle(frame, (int(x1), int(y1)), (int(x2), int(y2)), color, 2)
            cv2.putText(frame, label, (int(x1), int(y1) - 10), cv2.FONT_HERSHEY_SIMPLEX, 0.9, color, 2)
```

Fig. 13 Anomaly detection in YOLO

Performance analysis

```
# Evaluate the performance of the anomaly detection
if true_labels and predicted_labels:
    precision = precision_score(true_labels, predicted_labels)
    recall = recall_score(true_labels, predicted_labels)
    f1 = f1_score(true_labels, predicted_labels)
    cm = confusion_matrix(true_labels, predicted_labels)

    # # Display precision, recall, and F1 score in a message box
    # messagebox.showinfo("Anomaly Detection Results",
    #                     f"Precision: {precision:.2f}\nRecall: {recall:.2f}\nF1 Score: {f1:.2f}")

    # Optionally, return the metrics if you need to further process or log them
    return precision, recall, f1, cm
```

Fig. 14 F1 score, Precision and Recall evaluation

Findings and Conversation

In PHASE 1 we find out the processed output of the YOLO model and in Phase 2 we processed this output as an input to SSD and accuracy with the fusion of model is more in comparison to individual model. The output of individual model and combined models was given as follows:

YOLO

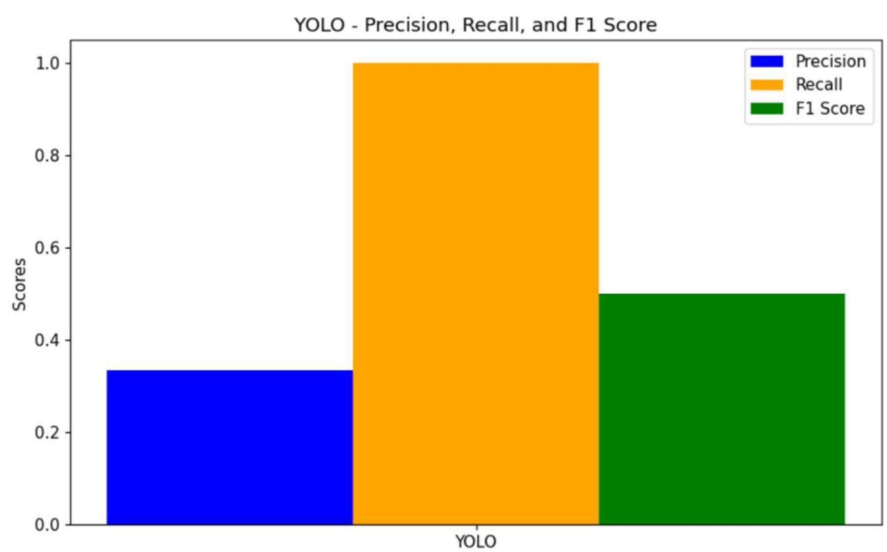

Fig. 15 F1 Score for YOLO

a. F1 Score: 0.5011

F1 score is harmonic mean of precision and recall . A score of 0.5011 is moderate, reflecting the trade-off between the high recall and low precision.

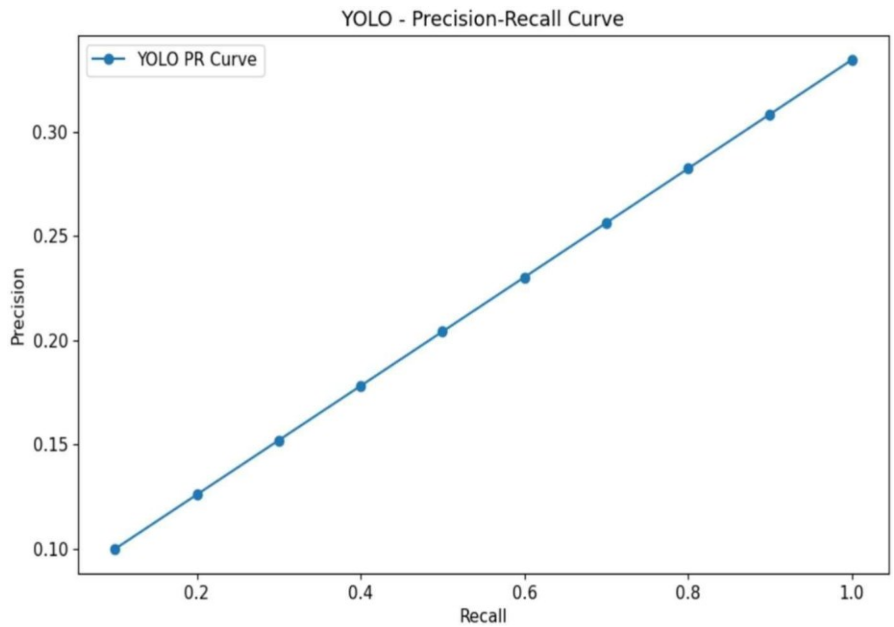

Fig. 16 Precision Recall Curve for YOLO

b. Precision: 0.3343

The precision metric quantifies how accurate the positive forecasts are. A precision of 0.3343 means that around 33% of the objects that YOLO predicted as positives (detected) were actually correct. This is relatively low, indicating that YOLO has a high rate of false positives in this case.

c. Recall: 1.0

Recall is the measure of how many actual positive objects was detected out of all the positives present. A recall of 1.0 means YOLO detected all the actual objects correctly. However, the high recall combined with the low precision suggests that YOLO is detecting too many false positives while still capturing all the actual objects.

Fig. 17 Anomaly Detection using YOLO

SSD

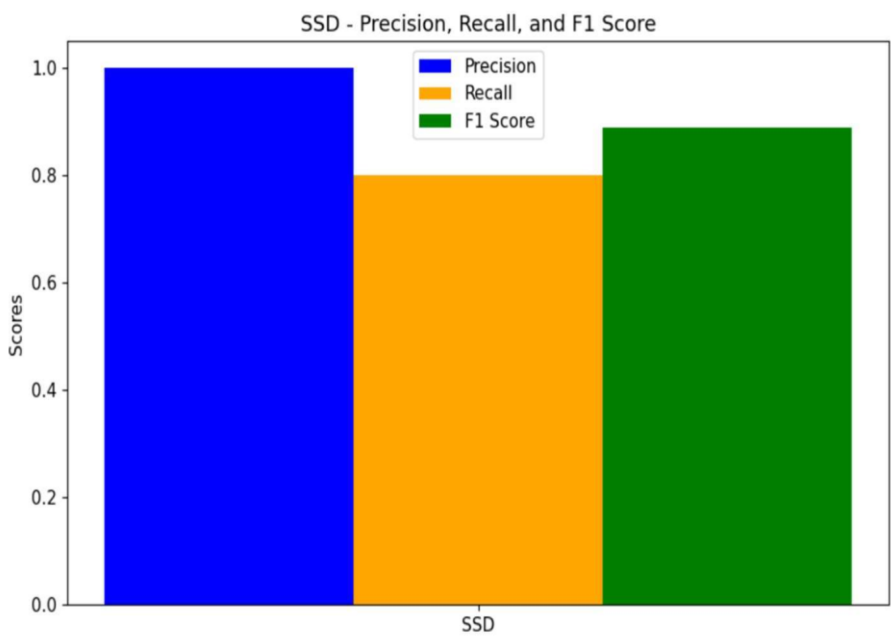

Fig. 18 F1 Score for SSD

a. F1 Score: 1.0

Since both recall & precision are perfect, the F1-score is also 1.0, indicating flawless performance.

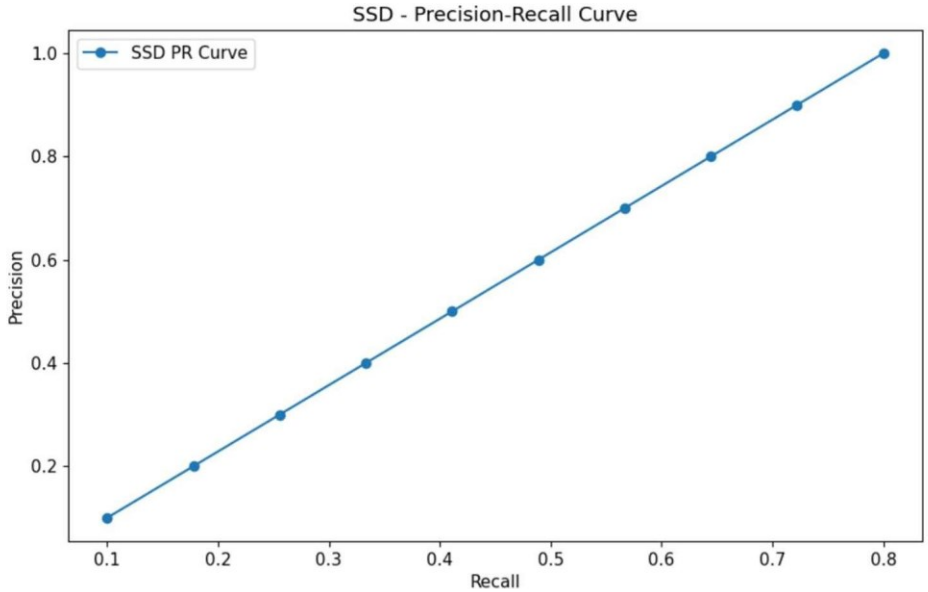

Fig. 19 Precision Recall Curve for SSD

b. Precision: 1.0

SSD's precision is 1.0, meaning it perfectly predicted all the positives. No false positives were reported.

c. Recall: 1.0

A recall of 1.0 means SSD detected all actual objects in the video. This indicates that SSD correctly identified all objects present and did not miss any.

Fig. 20 Anomaly Detection using SSD

Fusion Output of SSD and YOLO

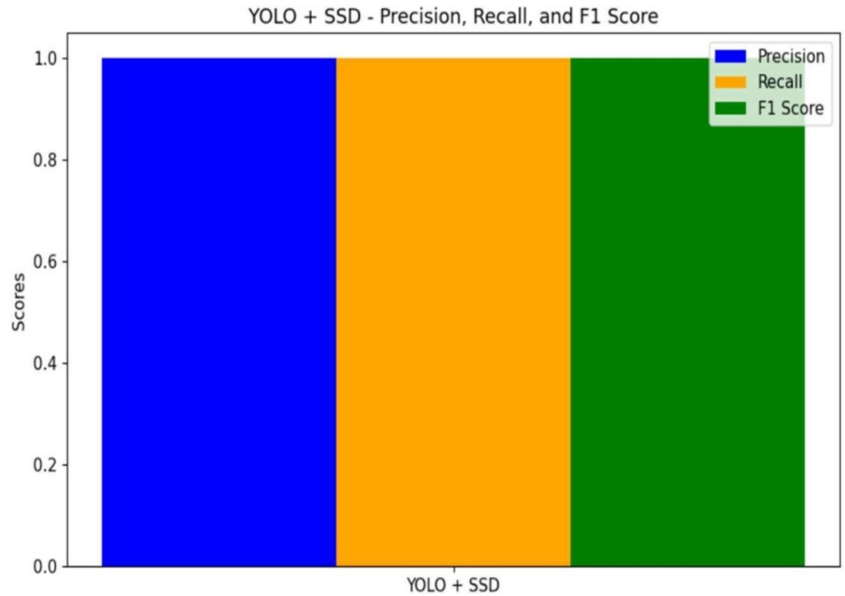

Fig. 21 F1 Score for SSD and YOLO

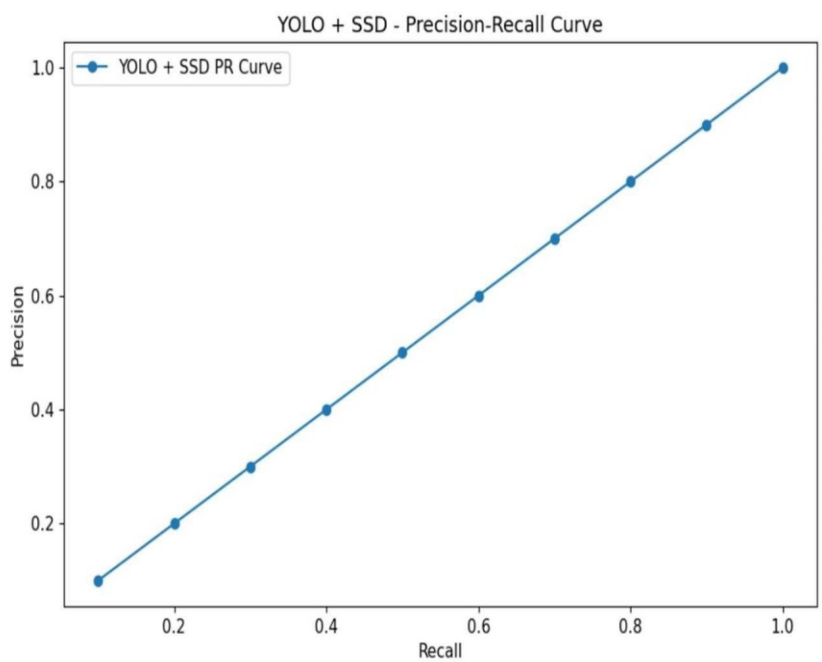

Fig. 22 Precision Recall Curve for SSD and YOLO

Fig. 23 Anomaly Detection using SSD and YOLO

Summary

- **YOLO:**

While YOLO has a perfect recall, meaning it did not miss any objects, it has a low precision, which means it is incorrectly detecting a lot of false positives. The low precision pulls down its F1 score to about 0.50.

- **SSD:**

On the other hand, SSD performed perfectly with a recall, precision and f1 score of 1.0, indicating it identified all objects correctly without any false.

- **SSD and YOLO:**

After making fusion of SSD and YPLO the output changes and the accuracy increases. In this process we make use of processed output of SSD as an input to YOLO to make an efficiency with higher performance.

References

[1] Elharrouss, Omar, Noor Almaadeed, and Somaya Al-Maadeed. 2021. "A Review of Video Surveillance Systems." Journal of Visual Communication and Image Representation 77 (February): 103116. https://doi.org/10.1016/j.jvcir.2021.103116.

[2] Aung, Htet, Alexander V. Bobkov, and Nyan Lin Tun. 2021. "Face Detection in Real Time Live Video Using Yolo Algorithm Based on VGG16 Convolutional Neural Network." Proceedings - 2021 International Conference on Industrial Engineering, Applications and Manufacturing, ICIEAM 2021, IEEE Access 697–702. https://doi.org/10.1109/ICIEAM51226.2021.9446291.

[3] Zh Satybaldina, D, N S Glazyrina, K A Kalymova, and V S Stepanov. 2021. "Development of an Algorithm for Abnormal Human Behavior Detection in Intelligent Video Surveillance System." IOP Conference Series: Materials Science and Engineering 1069 (1): 012046. https://doi.org/10.1088/1757-899x/1069/1/012046.

[4] Ezzat, Mostafa Ahmed, Mohamed A. Abd El Ghany, Sultan Almotairi, and Mohammed A.M. Salem. 2021. "Horizontal Review on Video Surveillance for Smart Cities: Edge Devices, Applications, Datasets, and Future Trends." Journal of Sensors 21 (9): 1–28. https://doi.org/10.3390/s21093222.

[5] Nehru, Arun. 2021. "Materials Today : Proceedings Deep Learning-Based Real-World Object Detection and Improved Anomaly Detection for Surveillance Videos." Materials Today: Proceedings, no. xxxx. https://doi.org/10.1016/j.matpr.2021.07.064.

[6] Sarda, Abhishek, Shubhra Dixit, and Anupama Bhan. 2021. "Object Detection for Autonomous Driving Using YOLO Algorithm." Proceedings of 2021 2nd International Conference on Intelligent Engineering and Management, ICIEM 2021, IEEE Access no. Icicv: 447–51. https://doi.org/10.1109/ICIEM51511.2021.9445365.

[7] Vinith, M. A., G. Pradeep, and B. Priya. 2021. "An Approach for Detecting and Identifying Suspected Weapons Using YOLO Algorithm." 2021 3rd International Conference on Signal Processing and Communication, ICPSC

2021, IEEE Access , no. May:478–80. https://doi.org/10.1109/ICSPC51351.2021.9451686.

[8] Xu, Jie. 2021. "A Deep Learning Approach to Building an Intelligent Video Surveillance System." Multimedia Tools and Applications 80 (4): 5495–5515. https://doi.org/10.1007/s11042-020-09964-6.

[9] Singh, Virender, Swati Singh, and Pooja Gupta. 2020. "ScienceDirect Real- Time Anomaly Recognition Through CCTV Using Neural Networks." Procedia Computer Science 173 (2019): 254–63. https://doi.org/10.1016/j.procs.2020.06.030.

[10] William Tarimo, Moustafa Sabra, Shonan Hendre. 2020. "Real Time Deep Learning based Object Detection Framework." IEEE Symposium series on Computational Intelligence (SSCI) 21 (2): 1829–36.

[11] Bamne, Bulbul, Neha Shrivastava, Lokesh Parashar, and Upendra Singh. 2020. "Transfer Learning-Based Object Detection by Using Convolutional Neural Networks." Proceedings of the International Conference on Electronics and Sustainable Communication Systems, ICESC 2020, IEEE Access, no. Icesc: 328–32. https://doi.org/10.1109/ICESC48915.2020.9156060.

[12] Alhabshee, Sharifah Maryam, and Abu Ubaidah Bin Shamsudin. 2020. "Deep Learning Traffic Sign Recognition in Autonomous Vehicle." 2020 IEEE Student Conference on Research and Development, SCOReD 2020, IEEE Access, no. September: 438–42. https://doi.org/10.1109/SCOReD50371.2020.9251034.

[13] Nguyen, Minh T., Linh H. Truong, Trang T. Tran, and Chen Fu Chien. 2020. "Artificial Intelligence Based Data Processing Algorithm for Video Surveillance to Empower Industry 3.5." Computers and Industrial Engineering 148 (July): 106671. https://doi.org/10.1016/j.cie.2020.106671.

[14] Ishtiaq, Muhammad, Rashid Amin, Sultan H. Almotiri, Mohammed A. Al Ghamdi, and Hamza Aldabbas. 2020. "Deep Learning Based Intelligent Surveillance System." International Journal of Advanced Computer Science and Applications 11 (4): 603–13. https://doi.org/10.14569/IJACSA.2020.0110479.

[15] Socha, Robert, and Bogusław Kogut. 2020. "Urban Video Surveillance as a Tool to Improve Security in Public Spaces." Sustainability (Switzerland) 12 (15). https://doi.org/10.3390/su12156210.

[16] Tanberk, Senem, Dilek Bilgin Tukel, and Mitat Uysal. 2020. "A Simple AI- Powered Video Analytics Framework for Human Motion Imitation." Proceedings - 2020 Innovations in Intelligent Systems and Applications Conference, IEEE Access, ASYU 2020, 0–1. https://doi.org/10.1109/ASYU50717.2020.9259895.

[17] Hamami, Faqih, Iqbal Ahmad Dahlan, Setya Widyawan Prakosa, and Khamal Fauzan Somantri. 2020. "Big Data Analytics for Processing Real-Time Unstructured Data from CCTV in Traffic Management." 2020 International Conference on Data Science and Its Applications, ICoDSA 2020, IEEE Access, 6–10. https://doi.org/10.1109/ICoDSA50139.2020.9212858.

[18] Kulkarni, Vaibhavi. 2020. "Video Analytics for Face Detection and Tracking," International Conference on Advances in Computing, Communication Control and Networking (ICACCCN),IEEE ,962–65.

[19] Khudhair, Abdullah B., and Rana F. Ghani. 2020. "IoT Based Smart Video Surveillance System Using Convolutional Neural Network." Proceedings of the 6th International Engineering Conference "'Sustainable Technology and Development'", IEC 2020, IEEE Access, 163–68. https://doi.org/10.1109/IEC49899.2020.9122901.

[20] Beghdadi, Azeddine, Ismail Bezzine, and Muhammad Ali Qureshi. 2020. "A Perceptual Quality-Driven Video Surveillance System." Proceedings - 2020 23rd IEEE International Multi-Topic Conference, INMIC 2020. https://doi.org/10.1109/INMIC50486.2020.9318122.

[21] Shidik, Guruh Fajar, Edi Noersasongko, Adhitya Nugraha, Pulung Nurtantio Andono, Jumanto Jumanto, and Edi Jaya Kusuma. 2019. "A Systematic Review of Intelligence Video Surveillance: Trends, Techniques, Frameworks, and Datasets." IEEE Access 7: 170457–73. https://doi.org/10.1109/ACCESS.2019.2955387.

[22] Liu, Yutong, Linghe Kong, Guihai Chen, Fangqin Xu, and Zhanquan Wang. 2019. "Light-Weight AI and IoT Collaboration for Surveillance Video Pre- Processing." Journal of Systems Architecture, IEEE Access, no. November: 101934. https://doi.org/10.1016/j.sysarc.2020.101934.

[23] Sreenu, G., and M. A. Saleem Durai. 2019. "Intelligent Video Surveillance: A Review through Deep Learning Techniques for Crowd Analysis." Journal of Big Data 6 (1): 1–28. https://doi.org/10.1186/s40537-019-0212-5.

[24] Mohana, and H. V. Ravish Aradhya. 2019. "Object Detection and Tracking Using Deep Learning and Artificial Intelligence for Video Surveillance Applications." International Journal of Advanced Computer Science and Applications, IEEE Access 10 (12): 517–30.

[25] Zhao, Zhong Qiu, Peng Zheng, Shou Tao Xu, and Xindong Wu. 2019. "Object Detection with Deep Learning: A Review." IEEE Transactions on Neural Networks and Learning Systems 30 (11): 3212–32. https://doi.org/10.1109/TNNLS.2018.2876865.

[26] Muthurasu N, Rajasekar V, "Unveiling Anomalies in Crowds through Advanced Deep Learning for Unusual Activity Detection", Journal of Theoretical and Applied Information Technology 15th March 2024. Vol.102. No 5

[27] Patrick Vermander, Aitziber Mancisidor, Itziar Cabanes and Nerea Perez, "Intelligent systems for sitting posture monitoring and anomaly detection: an overview", journal of NeuroEngineering and Rehabilitation (2024) 21:28

[28] Esraa A. Mahareek, Eman K. ElSayed, Nahed M. ElDesouky, Kamal A. ElDahshan, "Anomaly Detection in Surveillance Videos: Survey", International Journal of Theoretical and Applied Research, 2024, Vol. 3, No. 1, 328-343

[29] Pierre PATHE, Benjamin Pannetier, Bartheye Olivier, Daniel, "Abnormal Behavior State-of-the-Art for UAV Detection in Complex Environments", HAL Id:hal-04513213 https://hal.science/hal-04513213 Submitted on 20 Mar 2024

[30] Huu-Thanh Duong, Viet-Tuan Le and Vinh Truong Hoang, "Deep Learning- Based Anomaly Detection in Video Surveillance: A Survey", Sensors 2023, 23, 5024. https://doi.org/10.3390/s23115024

[31] Neamah, K. A, "Mathematical Model for Handling Unstable Time Series by Using a Linear Approximation Technique", 2022, January 20, Webology, 19(1), 2835–2852.

[32] Mustafa, M. , "Real-Time Video Anomaly Detection for Smart Surveillance", SSRN Electronic Journal, 2022.

[33] Chakole, P. D., Satpute, V. R., & Cheggoju, N., " Crowd behavior anomaly detection using correlation of optical flow magnitude", Journal of Physics: Conference Series, 2273(1), 2022, May 1.

[34] Luque Sánchez, F., Hupont, I., Tabik, S., & Herrera, F. , "Revisiting crowd behaviour analysis through deep learning: Taxonomy, anomaly detection, crowd emotions, datasets, opportunities and prospects", 2020, December. Information Fusion, 64, 318–335.

[35] Bhuiyan, M. R., Abdullah, J., Hashim, N., & Al Farid, F. , "Video analytics using deep learning for crowd analysis: a review. Multimedia Tools and Applications", 2022, March 29, 81(19), 27895–27922.

[36] Lalit, R., Purwar, R. K., Verma, S., & Jain, A. , "Crowd abnormality detection in video sequences using supervised convolutional neural network. Multimedia Tools and Applications" , 2021, December 14, 81(4), 5259–5277.

[37] Sonkar, R., Rathod, S., Jadhav, R., & Patil, D. , "CROWD ABNORMAL BEHAVIOUR DETECTION USING DEEP LEARNING" ITM Web of Conferences, 2020, 32, 03040.

[38] Bamaqa, A., Sedky, M., Bosakowski, T., Bakhtiari Bastaki, B., & Alshammari, N. O. , "SIMCD: SIMulated crowd data for anomaly detection and prediction. Expert Systems With Applications" , 2022, October., 203, 117475.

[39] Sharif, M. H., Jiao, L., & Omlin, C. W. , "Deep Crowd Anomaly Detection by Fusing Reconstruction and Prediction Networks. Electronics", 2023, March 23, 12(7), 1517.

[40] Jia, D., Zhang, C., & Zhang, B. , "Crowd density classification method based on pixels and texture features", Machine Vision and Applications, 2021, January 27, 32(2).

[41] Mehmood, A. , "Abnormal Behavior Detection in Uncrowded Videos with Two-Stream 3D Convolutional Neural Networks", Applied Sciences, 2021, January 27, 11(8), 3523.

[42] Wang, B., & Yang, C. , "Video Anomaly Detection Based on Convolutional Recurrent AutoEncoder", Sensors, 22(12), 2022, June 20, 4647.

[43] Surucu, O., Gadsden, S. A., & Yawney, J. , "Condition Monitoring using Machine Learning: A Review of Theory, Applications, and Recent

Advances", Expert Systems With Applications, 2023, 221, 119738.

[44] Chen, S., & Guo, W. , "Auto- Encoders in Deep Learning—A Review with New Perspectives", Mathematics, 2023, April 7, 11(8), 1777.

[45] "Smart Surveillance: A Review & Survey Through Deep Learning Techniques for Detection & Analysis", Journal of Sensor Networks and Data Communications, 2023, October 5,3(1).

[46] Sreenu, G., & Saleem Durai, M. A., " Intelligent video surveillance: a review through deep learning techniques for crowd analysis" Journal of Big Data, 2019, June 6, 6(1).

[47] Nafea, O., Abdul, W., Muhammad, G., & Alsulaiman, M., "Sensor-Based Human Activity Recognition with Spatio- Temporal Deep Learning", Sensors, 2021, March 18 ,21(6), 2141.

[48] Nawaz, A., Zhiqiu, H., Senzhang, W., Hussain, Y., Khan, I., & Khan, Z. , "Convolutional LSTM based transportation mode learning from raw GPS trajectories", IET Intelligent Transport Systems, 2020, April 8, 14(6), 570–577.

[49] Zou, Z., Cheng, Y., Qu, X., Ji, S., Guo, X., & Zhou, P. , "Attend to count: Crowd counting with adaptive capacity multiscale CNNs", Neurocomputing, 2019, November ,367, 75–83.

[50] Shrestha, S., & Vanneschi, L. , "Improved Fully Convolutional Network with Conditional Random Fields for Building Extraction. Remote Sensing, 2018, July 18 , 10(7), 1135.

[51] Wan, B., Jiang, W., Fang, Y., Luo, Z., & Ding, G. , "Anomaly detection in video sequences: A benchmark and computational model", IET Image Processing, 2021, May 22,15(14), 3454– 3465.

[52] Xia, L., Mi, S., Zhang, J., Luo, J., Shen, Z., & Cheng, Y. , "Dual-Stream Feature Extraction Network Based on CNN and Transformer for Building Extraction", Remote Sensing, 2023, May 22 , 15(10), 2689.

[53] Ahmed, A. A., & Echi, M. (2021). , "Hawk-Eye: An AI-Powered Threat Detector for Intelligent Surveillance Cameras", IEEE Access, 9, 63283– 63293. https://doi.org/10.1109/ACCESS.2021.3074319

[54] Alhabshee, S. M., & Shamsudin, A. U. bin. (2020) , " Deep Learning Traffic Sign Recognition in Autonomous Vehicle ", 2020 IEEE Student Conference on Research and Development, SCOReD 2020, 438–442.

https://doi.org/10.1109/SCOReD50371.2020.9251034

[55] Alkalouti, H. N., & al Masre, M. A. (2021, April 21), "Encoder-decoder model for automatic video captioning using yolo algorithm.", 2021 IEEE International IOT, Electronics and Mechatronics Conference, IEMTRONICS 2021 - Proceedings. https://doi.org/10.1109/IEMTRONICS52119.2021.9422600

[56] Aung, H., Bobkov, A. v., & Tun, N. L. (2021) , " Face detection in real time live video using yolo algorithm based on VGG16 convolutional neural network", Proceedings - 2021 International Conference on Industrial Engineering, Applications and Manufacturing, ICIEAM 2021, 697–702. https://doi.org/10.1109/ICIEAM51226.2021.9446291

[57] Chen, J., Li, K., Deng, Q., Li, K., & Yu, P. S. (2019), "Distributed Deep Learning Model for Intelligent Video Surveillance Systems with Edge Computing", https://doi.org/10.1109/TII.2019.2909473

[58] Fu, L. Y., Tunku, U., & Rahman, A. (2020), "Video Surveillance Using Deep Learning With Few Data Samples".

[59] Hamami, F., Prakosa, S. W., Dahlan, I. A., & Somantri, K. F. (2020)., " Big Data Analytics for Processing Real-time Unstructured Data from CCTV in Traffic Management", in 2020 International Conference on Data Science and Its Applications (ICoDSA).

[60] Ishtiaq, M., Amin, R., Almotiri, S. H., al Ghamdi, M. A., & Aldabbas, H. (2020). , "Deep learning based intelligent surveillance system", International Journal of Advanced Computer Science and Applications, 11(4), 603–613. https://doi.org/10.14569/IJACSA.2020.0110479

[61] Kalbo, N., Mirsky, Y., Shabtai, A., & Elovici, Y. (2020), "The security of ip-based video surveillance systems", In Sensors (Switzerland) (Vol. 20, Issue 17, pp. 1–27). MDPI AG. https://doi.org/10.3390/s20174806

[62] Liu, Y., Kong, L., Chen, G., Xu, F., & Wang, Z. (2021), "Light-weight AI and IoT collaboration for surveillance video pre-processing", Journal of Systems Architecture, 114. https://doi.org/10.1016/j.sysarc.2020.101934

[63] Mansour, R. F., Escorcia-Gutierrez, J., Gamarra, M., Villanueva, J. A., & Leal, N. (2021). , "Intelligent video anomaly detection and classification using faster RCNN with deep reinforcement learning model. Image and Vision Computing", 112. https://doi.org/10.1016/j.imavis.2021.104229

[64] Neto, A. R., Silva, T. P., Batista, T., Delicato, F. C., Pires, P. F., & Lopes, F. (2021). , "Leveraging edge intelligence for video analytics in smart city applications", Information (Switzerland), 12(1), 1–26. https://doi.org/10.3390/info12010014

[65] Ozturk, G., Koker, R., Eldogan, O., & Karayel, D. (2020, October 22)., "Recognition of Vehicles, Pedestrians and Traffic Signs Using Convolutional Neural Networks", 4th International Symposium on Multidisciplinary Studies and Innovative Technologies, ISMSIT 2020 - Proceedings. https://doi.org/10.1109/ISMSIT50672.2020.9255148

[66] Proceedings of the 6th International Engineering Conference (IEC2020) : February 26-27, 2020, Erbil, Kurdistan Region-Iraq. (2020). IEEE.

[67] Proceedings of the International Conference on Electronics and Sustainable Communication Systems (ICESC 2020) : 02-04, July 2020. (2020). [IEEE].

[68] Ragedhaksha, Darshini, Shahil, & Arunnehru, J. (2023) , "Deep learning- based real-world object detection and improved anomaly detection for surveillance videos", Materials Today: Proceedings, 80, 2911–2916. https://doi.org/10.1016/j.matpr.2021.07.064

[69] Shaposhnikov, S. . (2020). Proceedings of the 2020 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (ElConRus) : January 27-30, 2020, St. Petersburg and Moscow, Russia. St. Petersburg Electrotechnical University "LETI."

[70] Singh, V., Singh, S., & Gupta, P. (2020). , "Real-Time Anomaly Recognition Through CCTV Using Neural Networks", Procedia Computer Science, 173, 254–263. https://doi.org/10.1016/j.procs.2020.06.030

[71] Smith, C. L., & Brooks, D. J. (2013) , "Integrated Identification Technology", In Security Science (pp. 153–175). Elsevier. https://doi.org/10.1016/b978-0-12-394436-8.00007-2

[72] Sung, C. S., & Park, J. Y. (2021), "Design of an intelligent video surveillance system for crime prevention: applying deep learning technology", Multimedia Tools and Applications, 80(26–27), 34297–34309. https://doi.org/10.1007/s11042-021-10809-z

[73] Tseng, C. H., Hsieh, C. C., Jwo, D. J., Wu, J. H., Sheu, R. K., & Chen, L. C. (2021). , "Person Retrieval in Video Surveillance Using Deep Learning- Based Instance Segmentation". Journal of Sensors, 2021.

https://doi.org/10.1155/2021/9566628

[74] Wang, P., Wang, P., & Fan, E. (2021). , "Violence detection and face recognition based on deep learning", Pattern Recognition Letters, 142, 20– https://doi.org/10.1016/j.patrec.2020.11.018

[75] Xiaoyan, W., Pei, C., Rui, L., Bo, Y., & Wenzhao, N. (2020). , "Research on intelligent analysis technology of power monitoring video data based on convolutional neural network", Proceedings - 2020 5th International Conference on Mechanical, Control and Computer Engineering, ICMCCE 2020, 2461–2464. https://doi.org/10.1109/ICMCCE51767.2020.00531

[76] Yi, J., Min, C., & Kawsar, F. (2021). , "Vision Paper: Towards Software- Defined Video Analytics with Cross-Camera Collaboration", SenSys 2021 - Proceedings of the 2021 19th ACM Conference on Embedded Networked Sensor Systems, 474–477. https://doi.org/10.1145/3485730.3493453

[77] Zong Chen, Dr. J. I. (2020). "Smart Security System for Suspicious Activity Detection in Volatile Areas. Journal of Information Technology and Digital World", 02(01), 64–72. https://doi.org/10.36548/jitdw.2020.1.006

[78] Zou, X. (2019). , "A Review of object detection techniques", Proceedings - 2019 International Conference on Smart Grid and Electrical Automation, ICSGEA 2019, 251–254. https://doi.org/10.1109/ICSGEA.2019.00065

[79] Ben Mabrouk, A., & Zagrouba, E. (2018, January). , "Abnormal behavior recognition for intelligent video surveillance systems: A review", Expert Systems With Applications, 91, 480–491.

[80] Ammar, H., & Cherif, A. (2021, March 10). , " DeepROD: a deep learning approach for realtime and online detection of a panic behavior in human crowds", Machine Vision and Applications, 32(3). https://doi.org/10.1007/s00138-021-01182-w

[81] GCR, K. (2020, July 25). , "Automatic Detection and Initiation for Multiple Human Tracking Using Particle Filter", Journal of Advanced Research in Dynamical and Control Systems, 12(SP7), 2607–2612.

[82] Wan, B., Jiang, W., Fang, Y., Luo, Z., & Ding, G. (2021, May 22). , "Anomaly detection in video sequences: A benchmark and computational model", IET Image Processing, 15(14), 3454–3465.

[83] Ammar, H., & Cherif, A., "DeepROD: a deep learning approach for realtime and online detection of a panic behavior in human crowds", Machine Vision

and Applications, 32(3).

[84] Socha, R., & Kogut, B. (2020) , "Urban video surveillance as a tool to improve security in public spaces", Sustainability (Switzerland), 12(15). https://doi.org/10.3390/su12156210

[85] Y. C. Hou, M. Z. Baharuddin, S. Yussof, and S. Dzulkifly, "Social distancing detection with deep learning model," in 2020 8th International Conference on Information Technology and Multimedia (ICIMU). IEEE, 2020, pp. 334–338

[86] Shri, S. J., & Jothilakshmi, S. (2019). , "Anomaly Detection in Video Events using Deep Learning. International Journal of Innovative Technology and Exploring Engineering", 8(9), 1313–1316. https://doi.org/10.35940/ijitee.I7914.078919

[87] Peng Wu, Jing Liu, and Fang Shen. 2019. , "A Deep One Class Neural Network for Anomalous Event Detection in Complex Scenes", IEEE Transactions on Neural Networks and Learning Systems (2019)

[88] M. Izadi, Z. Azimifar, and G. H. Jowkar, "Abnormal event detection in indoor video using feature coding," in 19th CSI International Symposium on Artificial Intelligence and Signal Processing, AISP 2017, 2018, vol. 2018- Janua, pp. 151–155.

[89] Turchini, F., Seidenari, L., Uricchio, T., & del Bimbo, A. (2018) , "Deep learning based surveillance system for open critical areas", Inventions, 3(4). https://doi.org/10.3390/inventions3040069

[90] W. Guangli, LIU Liping, Zhang Chen, TAN Dengtai. "Video Abnormal Event Detection Based on ELM", 2019 IEEE 4th International Conference on Signal and Image Processing (ICSIP), Wuxi, China, pp 367 – 371

[91] G. S. Chadha, A. Rabbani, and A. Schwung, "Comparison of semi-supervised deep neural networks for anomaly detection in industrial processes," IEEE Int. Conf. Ind. Informatics, vol. 2019-July, pp. 214–219, 2019.

[92] Shriram S, Sivasankar E. , "Anomaly detection on shuttle data using unsupervised learning techniques", In 2019 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE). 2019;pages 221–225. IEEE

[93] Raghunandan, A., Mohana, Raghav, P., & Aradhya, H. V. R. (2018). , "Object Detection Algorithms for Video Surveillance Applications" , Proceedings of the 2018 IEEE International Conference on

Communication and Signal Processing, ICCSP 2018, 563–568. https://doi.org/10.1109/ICCSP.2018.852446

[94] Vartouni A, Kashi SS, Teshnehlab M. , "An anomaly detection method to detect web attacks using stacked auto-encoder" , In 2018 6th Iranian Joint Congress on Fuzzy and Intelligent Systems (CFIS). 2018;pages 131–134. IEEE.

[95] N. M. Palomino, N., G. C. Chávez. "Abnormal Event Detection in Video Motion and Appearance Information," in Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications. CIARP 2017. Lecture Notes in Computer Science, 2018, vol. 10657, February 2018, pp. 382–390. Springer, Cham.

[96] Nikouei, S.Y., Chen, Y., Faughnan, T.R. ,"Smart surveillance as an edge service for real-time human detection and tracking", in: 2018 IEEE/ACM Symposium on Edge Computing (SEC), pp. 336–337. Seattle, WA, USA (2018).

[97] E. d'Afflisio et al., "Detecting anomalous deviations from standard maritime routes using the Ornstein– Uhlenbeck process," IEEE Transactions on Signal Processing, 2018

[98] Varghese EB, Thampi SM (2018) , "A deep learning approach to predict crowd behavior based on emotion", In International Conference on Smart Multimedia Aug 24 (pp. 296-307). Springer, Cham

[99] W. Liu, W. Luo, D. Lian and S. Gao, "Future frame prediction for anomaly detection a new baseline," in Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, pp. 6536–6545, 2018

[100] Vignesh, K., Yadav, G., Sethi, A. , "Abnormal event detection on BMTT- PETS 2017 surveillance challenge", In: 2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), pp. 2161–2168. Honolulu, HI, USA (2017).

[101] M. S. Parwez, D. B. Rawat, and M. Garuba, "Big Data Analytics for User- Activity Analysis and User-Anomaly Detection in Mobile Wireless Network," IEEE Transactions on Industrial Informatics, vol. 13, no. 4, pp. 2058–2065, Aug. 2017, doi: 10.1109/TII.2017.2650206.

[102] A. Sarkale, K. Shah, A. Chaudhary, and T. P. N, "A Literature Survey: Neural Networks for object detection," International Journal for Research and

Innovation, vol. 1, no. 1, 2018, Accessed: Oct. 23, 2024. [Online]. Available: www.viva-technology.org/New/IJRI

[103] J. Redmon and A. Farhadi, "YOLOv3: An Incremental Improvement", Accessed: Oct. 23, 2024. [Online]. Available: https://pjreddie.com/yolo/.

[104] "Sci-Hub | Object Detection Using Convolutional Neural Networks. TENCON 2018 - 2018 IEEE Region 10 Conference | 10.1109/TENCON.2018.8650517." Accessed: Oct. 23, 2024. [Online]. Available: https://sci-hub.st/10.1109/TENCON.2018.8650517

[105] C. Anusha, "Object Detection using Deep Learning," International Journal of Computer Applications, Accessed: Oct. 23, 2024. [Online]. Available: https://www.academia.edu/48046097/Object_Detection_using_Deep_Learni ng.

[106] "Real-Time Object Detection Using Deep Learning And Open cv", doi: 10.35940/ijitee.L1103.10812S19.