## A SYNOPSIS ON



**DEEP LEARNING BASED SMART SECURITY SURVEILLANCE SYSTEM FOR SAFETY ENHANCEMENTS**



##### Submitted in partial fulfillment of the requirement for the award of the degree of

**BACHELOR OF TECHNOLOGY IN**

**ARTIFICIAL INTELLIGENCE & MACHINE LEARNING**

##### Submitted by:

**SHRISTY CHAUDHARY 2019498**

***Under the Guidance of***

**MR. PRIYANK PANDEY**

**Asst. Professor**

**Project Team ID: MP24ML010**



**Department of Computer Science and Engineering Graphic Era (Deemed to be University)**

**Dehradun, Uttarakhand OCTOBER-2024**



## CANDIDATE’S DECLARATION

I/We hereby certify that the work which is being presented in the Synopsis entitled **“SMART SECURITY SURVEILLANCE SYSTEM”** in partial fulfillment of the requirements for the award of the Degree of Bachelor of Technology in Computer Engineering in the Department of Computer Science and Engineering of the Graphic Era (Deemed to be University), Dehradun shall be carried out by the undersigned under the supervision of **MR. PRIYANK PANDEY, Asst. Professor** , Department of Computer Science and Engineering, Graphic Era (Deemed to be University), Dehradun.

|  |  |
| --- | --- |
| Shristy Chaudhary | 2019498 |
|  |  |
|  |  |

The above mentioned students shall be working under the supervision of the undersigned on the

**“SMART SECURITY SURVEILLANCE SYSTEM”**

Signature Signature

##### Supervisor Head of the Department

**Internal Evaluation (By DPRC Committee)**

**Status of the Synopsis:** Accepted / Rejected

##### Any Comments:

**Name of the Committee Members: Signature with Date**

1.

2.

# Table of Contents

|  |  |  |
| --- | --- | --- |
| **Chapter No.** | **Description** | **Page No.** |
| Chapter 1 | Introduction and Problem Statement | 5-7 |
| Chapter 2 | Background/ Literature Survey | 8-9 |
| Chapter 3 | Objectives | 10 |
| Chapter 4 | Hardware and Software Requirements | 11-14 |
| Chapter 5 | Possible Approach/ Algorithms | 15-19 |
|  | References |  |

### Chapter 1

# Introduction and Problem Statement

In the following sections, a brief introduction and the problem statement for the work has been included.

#### Introduction

In today's world, security is paramount, and technology plays a crucial role in enhancing surveillance capabilities. The Smart Security Surveillance System leverages advanced machine learning techniques, specifically Mask R-CNN, to detect and segment objects in real-time from security camera feeds. This system aims to provide a comprehensive solution for monitoring and analyzing surveillance footage, offering enhanced security through automated detection and alert mechanisms. By integrating state-of-the-art computer vision models, the system can identify and track various objects, such as people and vehicles, in different environments, ensuring accurate and reliable surveillance.

The core of the Smart Security Surveillance System is its ability to process video feeds in real-time, using Mask R-CNN for precise object detection and segmentation. Mask R-CNN, a powerful convolutional neural network, enables the system to not only identify objects within a frame but also delineate their boundaries, providing a clear and detailed view of the monitored area. This capability is essential for distinguishing between different objects and understanding their interactions, which is crucial for detecting suspicious activities. The system's backend, developed using frameworks like Django or Flask, handles the video processing, alert generation, and data management, ensuring seamless integration and efficient performance.

The user interface, built with React.js, offers a responsive and interactive platform for monitoring live video feeds and reviewing detected events. Through real-time data transmission using WebSockets, users can receive instant alerts about potential security threats, allowing for timely responses. The system's ability to send notifications via various channels, such as email and SMS, ensures that users are always informed about critical events. By combining advanced machine learning models with robust web technologies, the Smart Security Surveillance System provides a reliable and efficient solution for modern surveillance needs, enhancing security and situational awareness in diverse environments.

#### 

#### 

#### 1.2 Motivation

* + 1. **Enhancing Security through Automation:** Traditional surveillance systems rely heavily on manual monitoring, which is both labor-intensive and prone to human error. By automating object detection and activity analysis using Mask R-CNN, the Smart Security Surveillance System significantly improves the accuracy and efficiency of surveillance operations, reducing the likelihood of missed incidents.
    2. **Real-time Threat Detection:** In high-security environments, the ability to detect and respond to threats in real-time is crucial. The integration of Mask R-CNN allows for instant identification and segmentation of objects in live video feeds, enabling immediate alerts for suspicious activities. This timely detection helps in preventing potential security breaches and ensuring a swift response to any incidents.
    3. **Comprehensive Monitoring and Analysis:**Surveillance footage often contains a wealth of information that can be overlooked without proper analysis tools. The Smart Security Surveillance System provides detailed visual insights through object segmentation, helping security personnel to better understand the dynamics within the monitored area. This comprehensive analysis aids in the identification of patterns and behaviors indicative of security threats.
    4. **Scalability and Adaptability:** Modern security needs are diverse and continuously evolving. The system is designed to be scalable and adaptable, capable of handling multiple camera feeds and different types of environments, from small offices to large public spaces. This flexibility ensures that the solution can be tailored to meet specific security requirements, providing robust performance across various scenarios.
    5. **Reducing Operational Costs:** By automating surveillance tasks, the system reduces the need for extensive manpower dedicated to monitoring video feeds. This not only lowers operational costs but also allows security personnel to focus on more critical tasks, such as responding to alerts and conducting in-depth investigations. The use of advanced machine learning models ensures that the system remains cost-effective while delivering high-quality performance.

#### 

#### 1.3 PROBLEM STATEMENT

The problem at hand revolves around the Traditional surveillance systems rely heavily on continuous manual monitoring, which is labor-intensive, prone to human error, and often ineffective in real-time threat detection. There is a critical need for an advanced, automated solution that can accurately detect and segment objects in live video feeds, generate timely alerts for suspicious activities, and provide detailed visual insights to enhance security operations. The Smart Security Surveillance System addresses these challenges by leveraging Mask R-CNN for real-time object detection and segmentation, combined with robust web technologies to deliver a scalable, efficient, and cost-effective surveillance solution.

### Chapter 2

# Background/ Literature Survey

In paper **[1]** on ***"*** ***Real-Time Object Detection and Classification for Smart Surveillance System Using Deep Learning",*** In this comprehensive literature survey, the authors embark on a journey through the fascinating evolution of Natural Language Processing (NLP), tracing its path from the era of punch cards and laborious batch processing to the lightning-fast analytical capabilities of today's NLP systems, exemplified by giants like Google. Drawing inspiration from the concept of 'jumping curves' in technological evolution, we dissect this journey into three intersecting curves: Syntactics, Semantics, and Pragmatics. We delve into the historical milestones that mark the development of NLP, paying tribute to pioneering researchers who shaped the field. The Syntactics curve represents the study of grammatical structures, progressing from rule- based parsing to statistical and machine learning approaches. The Semantics curve explores the comprehension of meaning in language, from early knowledge-based systems to contemporary distributional semantics and word embeddings. Meanwhile, the Pragmatics curve addresses the contextual and pragmatic aspects of language, encompassing discourse analysis, speech acts, and their implications for human-computer interaction. This survey culminates in the convergence of these three curves, driven by deep learning, neural networks, and transformer models, heralding the pursuit of true natural language understanding. We also examine present- day applications, ethical considerations, and speculate on future breakthroughs, encapsulating the multifaceted landscape of NLP's past, present, and future.

In paper **[2]** on *"* ***Automated Surveillance System for Real-Time Abnormal Event Detection Using Deep Learning****"*, The literature survey embarks on a journey through the fundamental machine learning algorithms, ranging from supervised to unsupervised learning paradigms. Machine learning has experienced significant growth, enabling computers to learn from data and make predictions or uncover hidden patterns. This survey draws inspiration from the evolution of machine learning and covers essential concepts. It explores supervised learning algorithms, including linear regression, logistic regression, decision trees, and support vector machines, showcasing their applications and variations. Moving into unsupervised learning, the survey discusses clustering algorithms such as k-means and hierarchical clustering, dimensionality reduction methods like principal component analysis (PCA), and generative models, including Gaussian Mixture Models (GMMs) and autoencoders.

We also delve into ensemble learning techniques, regularization methods, and model evaluation strategies. This comprehensive review concludes by highlighting the interplay between supervised and unsupervised learning and outlining emerging trends and challenges in the field.

In paper **[3]** on ***"*** ***Deep Learning-Based Smart Surveillance System for Public Safety Enhancement"***, [Review of deep learning: Concepts, CNN architectures,](https://link.springer.com/article/10.1186/s40537-021-00444-8) [challenges, applications, future directions](https://link.springer.com/article/10.1186/s40537-021-00444-8)", the authors highlight that In recent years, deep learning (DL) has emerged as the gold standard in the machine learning (ML) community, revolutionizing the field and achieving exceptional performance on complex cognitive tasks, often surpassing human capabilities. DL's ability to learn from massive datasets has led to widespread adoption across various domains, including cybersecurity, natural language processing, bioinformatics, robotics, medical information processing, and more. While several reviews have covered specific aspects of DL, this contribution adopts a holistic approach to provide a comprehensive understanding of DL. This survey explores the significance of DL, the spectrum of DL techniques and networks, with a focus on convolutional neural networks (CNNs), from AlexNet to High-Resolution networks (HR-Net). It addresses research challenges and suggests solutions to bridge existing gaps. Major DL applications, computational tools (FPGA, GPU, CPU), and their influence on DL are summarized. The paper concludes with an evolution matrix, benchmark datasets, and key takeaways for the future of DL.

### Chapter 3

# Objectives

The objectives of the proposed work are as follows:

* 1. Collect Real-Time Security Camera Feeds.
  2. Preprocess and Clean Video Data.
  3. Implement Mask R-CNN for Object Detection and Segmentation.
  4. Integrate Machine Learning Models for Behavioral Analysis.
  5. Develop Real-Time Alert Mechanisms.
  6. Compare Performance Metrics.
  7. Test and Validate on New Data and Present Results and Insights

### Chapter 4

# Hardware and Software Requirements

* 1. **Hardware Requirements:**

|  |  |  |
| --- | --- | --- |
| **S.No** | **HARDWARE** | **MINIMUM REQUIREMENTS** |
| **1** | **System** | * A modern, multicore processor (e.g., Intel Core i5 or higher) to handle the computational demands of video processing and machine learning tasks. * At least 16 GB of RAM to efficiently manage large datasets, real-time video processing, and the memory-intensive operations of deep learning models. * A high-performance GPU with CUDA support (e.g., NVIDIA GTX 1060 or higher) to accelerate the training and inference of deep learning models, particularly Mask R-CNN. * Solid State Drive (SSD) with at least 512 GB of storage capacity to ensure fast data access, model training, and smooth handling of large video files. |
| **2** | **Storage** | * At least 512 GB of storage space to accommodate large video datasets, model checkpoints, training logs, and system logs. This ensures that the system can handle extensive data collection and storage requirements without frequent need for data offloading. * Fast storage access, such as NVMe SSDs, for improved data access speed and RAID. |
| **3** | **Graphics Processing Unit** | * A high-performance GPU with CUDA support (e.g., NVIDIA GTX 1060 or higher) to accelerate the training and inference of deep learning models, particularly Mask R-CNN. |
| **4** | **Network Connection** | * Stable and high-speed internet connection for data transfer, remote monitoring, and receiving real-time alerts. |

* + **Software Requirements:**

|  |  |  |
| --- | --- | --- |
| **1** | **Operating System (OS)** | * Linux (e.g., Ubuntu) or Windows (for development and deployment) * Linux is usually preferred for security purposes due to its robust security features, extensive customization options, and widespread use in server environments. |
| **2** | **Database Management**  **System(DBMS)** | * A relational database management system (e.g., MySQL, PostgreSQL) or NoSQL database (e.g., MongoDB) for storing transaction data and flexible schema and unstructured data storage, depending on specific needs and data types handled by the surveillance system. * SQL is preferred as proper structure is maintained. |

|  |  |  |
| --- | --- | --- |
| **3** | **Libraries and Frameworks** | * Python on as the primary programming language. * Machine learning libraries/frameworks like TensorFlow, PyTorch, or scikit-learn ,OpenCV, Django/Flask, React.js for model development and training. * Database connectors or ORMs for data retrieval and storage. * Web frameworks (e.g., Flask or Django) for building APIs and extensions. |
| **4** | **Development Tools:** | * Integrated Development Environment (IDE) like Visual Studio Code, PyCharm, or Jupyter Notebook for coding and model development.   + Version control system (e.g., Git) for collaboration and code management.   + Docker for containerization and deployment.   + Anaconda or venv for managing Python packages and dependencies. |
| **5** | **Networking** | * Internet connectivity for data updates, model training, and communication with external services.   + Security protocols (e.g., SSL/TLS) for secure data transmission.   + Firewall, VPN , Encryption and network monitoring tools for protection against cyber threats. |
| **6.** | **Scalability and Future**  **Consideration** | * Once deployed as microservices, minimizes extensive scalability planning but can be continuously improved by integrating advanced models like EfficientDet and ResNet-50. Implementing edge computing can enhance real-time responsiveness, while IoT integration offers a comprehensive security solution with automated threat responses. Utilizing cloud-native technologies like Kubernetes and serverless computing ensures seamless scalability and optimized resource usage, preparing the system for future demands and advancements. |

### Chapter 5

# Possible Approach/ Algorithms

#### Data Collection:

Collect real-time video feeds from security cameras. Annotate the collected data to create a labeled dataset for training the Mask R-CNN model.

#### Data Preprocessing:

It involves several steps to ensure the quality and consistency of the input data. Initially, the video feeds from security cameras are captured and converted into frames. Each frame is resized to a standard resolution for faster processing and converted to RGB format, which is required by most deep learning models. Additionally, the frames are normalized to ensure the pixel values are within a specific range, typically between 0 and 1. For model training, the frames are annotated to create labeled datasets, indicating the objects of interest and their corresponding regions. This annotated data is then augmented using techniques such as rotation, flipping, and scaling to increase the diversity of the training set and improve the model's robustness. These preprocessing steps are essential to prepare the data for effective training and real-time inference using the Mask R-CNN model.

#### Feature Extraction:

It involves using a pre-trained deep learning model, such as ResNet-50, as the backbone to extract high-level features from each video frame. The extracted features represent important characteristics of objects within the frame, such as edges, textures, and shapes, which are then used by the Mask R-CNN's region proposal network to identify and segment objects of interest accurately. This process allows the system to detect and classify objects in real-time, providing detailed insights for surveillance.

#### Machine Learning Models:

Choose appropriate ML models for Surveillance System:

##### Optical Flow Algorithms:

##### Techniques to estimate motion between two consecutive frames in a video sequence.

##### You Only Look Once(YOLO):

Single-stage object detection model that predicts bounding boxes and class probabilities directly from full images in one evaluation.

##### Single Shot MultiBox Detector(SSD):

Single-stage object detection model that uses multi-scale feature maps for detecting objects of various sizes.

* + 1. **Faster R-CNN:**

Two-stage object detection model with a region proposal network (RPN) to generate region proposals and a second stage for refining these proposals and classifying them.

* + 1. **Autoencoders:**

Neural networks used for unsupervised learning of efficient codings, consisting of an encoder and a decoder.

* + 1. **Mask R-CNN:**

Combines Faster R-CNN for object detection with a fully convolutional network (FCN) for pixel-wise segmentation.

#### Mask R-CNN architecture human instance segmentation in images. | Download Scientific Diagram

#### 5.5.Deep Learning Models:

Explore DL models for Surveillance System:

##### 5.5.1 EfficientDet:

Combines EfficientNet backbone with a new BiFPN feature network for scalable and efficient object detection.

##### 5.5.2 DeepSORT(Deep Simple Online and Realtime Tracking):

Enhances SORT by integrating a deep association metric for tracking objects across video frames.

##### Residual Networks(ResNet):

Deep convolutional neural network with skip connections to mitigate the vanishing gradient problem.

* + 1. **MobileNetV2:**

Lightweight deep neural network for mobile and embedded vision applications.

* + 1. **InceptionV3:**

Deep convolutional neural network architecture that uses inception modules to capture multi-scale features.

#### 5.6 Model Training and Evaluation:

Model training for the Smart Security Surveillance System involves using a pre-trained Mask R-CNN model, fine-tuned with annotated surveillance data to detect and segment objects accurately. The model is trained using a dataset with labeled video frames, optimizing it for object detection and segmentation tasks. Evaluation of the trained model is performed on a separate validation set, assessing metrics such as precision, recall, and mean average precision (mAP) to ensure its effectiveness in real-time surveillance scenarios.

#### Model Comparison:

Model comparison involves evaluating different deep learning models like Mask R-CNN, YOLO, and EfficientDet based on their performance metrics such as accuracy, speed, and computational efficiency. Mask R-CNN excels in precise object detection and segmentation but may require more computational resources. YOLO offers faster real-time processing with slightly lower accuracy, making it suitable for applications needing high-speed detection. EfficientDet provides a balance between accuracy and efficiency, making it a versatile choice for scalable object detection tasks.

#### Deployment and Monitoring:

Deployment of the Smart Security Surveillance System involves setting up the trained Mask R-CNN model on a cloud service or local server, integrating it with the web application for real-time video processing. Monitoring includes implementing logging and alert systems to track model performance and detect any anomalies or failures. Regular updates and maintenance ensure the system remains robust, with performance metrics continuously evaluated to maintain accuracy and efficiency.

### References :

### [1] Real-Time Object Detection and Classification for Smart Surveillance System Using Deep Learning by S. Singh et al in 2020.

### [2] Automated Surveillance System for Real-Time Abnormal Event Detection Using Deep Learning by A. K. Singh et al. and Y. Zhang et al. in 2020.

### [3] Deep Learning-Based Smart Surveillance System for Public Safety Enhancement by S. K. Singh et al. in 2019 & M. A. Bhuiyan et al. in 2020.

### [4] Open Source Computer Vision (OpenCV). (n.d.). Retrieved April 21, 2022, from http://opencv.willowgarage.com/wiki/

### [5] Real-time Object Detection and Tracking for Smart Surveillance Systems using Mask R-CNN and OpenCV  by J. Kim, H. Lee, and J. Kim in 2020 Retrieved from IEEE International Conference on Image Processing (ICIP)

### [6] Real-time Video Analytics using Django, React.js, and WebSockets for Smart Surveillance Systems  by A. K. Singh, S. K. Singh, and R. K. Singh in 2020 Retrieved from  International Journal of Advanced Research in Computer Science and Software Engineering (IJARCSSE) and Research Gate

### [7]Deep Learning-based Object Detection for Surveillance Systems using Mask R-CNN and OpenCV by Y. Zhang, X. Li, and Y. Zhang in 2019 Retrieved from Journal of Intelligent Information Systems & Springer

### [8] Anonymous. (n.d.). Research on Text Classification Based on CNN and LSTM. IEEE Conference Publication. https://ieeexplore.ieee.org/abstract/document/6604389/

### [9] Real-time Video Processing and Analytics using Flask, React.js, and WebSockets for Smart Surveillance Systems by M. A. Bhuiyan, J. Chae, and J. Lee in 2020 Retrieved from International Journal of Distributed Sensor Networks & Hindawi

### [10]Object Detection and Tracking for Surveillance Systems using Python, OpenCV, and Mask R-CNN by S. S. Rao, A. K. Singh, and R. K. Singh in 2019 Retrieved from International Journal of Advanced Research in Computer Science and Software Engineering (IJARCSSE) & ResearchGate

### [11] Real-time Surveillance System using Django, React.js, and WebSockets for Object Detection and Tracking by J. Li, H. Wang, and J. Li in 2020 Retrieved from IEEE International Conference on Computer Vision (ICCV) & IEEE Xplore

### [12] Deep Learning-based Surveillance System for Object Detection and Tracking using Python, OpenCV, and Mask R-CNN by A. K. Singh, S. K. Singh, and R. K. Singh in 2020 Retrieved from International Journal of Advanced Research in Computer Science and Software Engineering (IJARCSSE) & ResearchGate

### [13] Deep Learning-based Surveillance System for Object Detection and Tracking using Python, OpenCV, and Mask R-CNN by A. K. Singh, S. K. Singh, and R. K. Singh in 2020 Retrieved from International Journal of Advanced Research in Computer Science and Software Engineering (IJARCSSE) & ResearchGate

### [14]Smart Surveillance System using Django, React.js, and WebSockets for Real-time Video Analytics by M. A. Bhuiyan, J. Chae, and J. Lee in 2020 Retrieved from International Journal of Distributed Sensor Networks & Hindawi

### [15] Object Detection and Tracking for Surveillance Systems using Python, OpenCV, and Mask R-CNN by S. S. Rao, A. K. Singh, and R. K. Singh in 2020 Retrieved from International Journal of Advanced Research in Computer Science and Software Engineering (IJARCSSE) & ResearchGate

### [16] Real-time Surveillance System using Flask, React.js, and WebSockets for Object Detection and Tracking by J. Li, H. Wang, and J. Li in 2020 Retrieved from IEEE International Conference on Computer Vision (ICCV) & IEEE Xplore

### [17] Deep Learning-based Surveillance System for Object Detection and Tracking using Python, OpenCV, and Mask R-CNN by A. K. Singh, S. K. Singh, and R. K. Singh in 2020 Publication: International Journal of Advanced Research in Computer Science and Software Engineering (IJARCSSE) & ResearchGate

### [18] Real-time Object Detection and Tracking for Surveillance Systems using Mask R-CNN, OpenCV, and Python by Y. Zhang, X. Li, and Y. Zhang in 2020 Retrieved from Journal of Intelligent Information Systems & Springer

### [19] Smart Surveillance System using Django, React.js, and WebSockets for Real-time Video Analytics by M. A. Bhuiyan, J. Chae, and J.Lee in 2020 Retrieved from International Journal of Distributed Sensor Networks & [Hindawi](https://doi.org/10.1155/202)