INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)

www.ijprems.com
editor@ijprems.com

Vol. 04, Issue 05, May 2024, pp: 1948-1950

e-ISSN : 2583-1062

Impact Factor: 5.725

CROWD MANAGEMENT AND WORK MONITORING USING MACHINE LEARNING

Mrs. Peddaboina Yamuna[1], Venkata Siva Sai Nithin A[2], Thuppari Rakesh[3], Gandhamalla Sandeep[4], Gaddam Sai Suhas[5]

[1]Assistant Professor, CSE Dept, ACE Engineering College, Hyderabad, India.

[2,3,4,5]Student, CSE Dept, ACE Engineering College, Hyderabad, India.

ABSTRACT

The Indian Railways is one of the largest railway networks in the world, catering to millions of passengers daily. Ensuring the safety and security of this vast number of passengers, while also maintaining efficient operations, is a daunting task. Traditional methods of manual monitoring and surveillance have limitations, including human error and inability to process large amounts of data in real-time. To address these challenges, the integration of Artificial Intelligence (AI) and Machine Learning (ML) technology offers a promising solution. This project proposes the development of an AI-powered system for analyzing existing Closed-Circuit Television (CCTV) footage of the Indian Railways to enhance crowd management, crime prevention, and work monitoring. The system will utilize advanced AI algorithms to detect unusual behavior, track crowd movements, and identify potential security threats in real-time. By leveraging ML capabilities, the system will also be able to predict crowd patterns and optimize resource allocation. The proposed system will comprise several components. Firstly, advanced AI algorithms will be used to analyze CCTV footage and detect anomalies such as suspicious behavior, abandoned objects, or abnormal crowd movement. These algorithms will be trained on a large dataset of normal crowd behavior to minimize false positives. Secondly, MLalgorithms will be employed to predict crowd patterns based on historical data, seasonality, and external factors such as weather and events. This information will be used to optimize resource allocation, ensuring that security personnel are deployed effectively and efficiently. Thirdly, a user interface will be developed to display real-time data and insights to security personnel, enabling them to respond promptly to any potential threats or issues.

1. INTRODUCTION

The Indian Railways stands as one of the world's largest railway networks, serving millions of passengers daily across its vast expanse. Ensuring the safety, security, and operational efficiency of such a colossal transportation system presents an ongoing challenge. Traditional methods of manual monitoring and surveillance, while foundational, are increasingly proving insufficient in handling the complexity and scale of modern railway operations.

In response to these challenges, the integration of Artificial Intelligence (AI) and Machine Learning (ML) technologies emerges as a promising avenue for innovation. By harnessing the power of AI and ML, we aim to revolutionize crowd management, crime prevention, and work monitoring within the Indian Railways. Our project proposes the development of an AI-powered system designed to analyze existing Closed-Circuit Television (CCTV) footage across the Indian Railways network. This system is envisioned to serve as a proactive tool, leveraging advanced algorithms to enhance security measures and optimize operational efficiency in real-time.

The primary objectives of our project encompass the detection of anomalous behavior, tracking of crowd movements, and identification of potential security threats within CCTV footage. To achieve these goals, we will employ state-of-the-art AI algorithms trained on extensive datasets of normal crowd behavior. This training will enable the system to minimize false positives and accurately identify deviations from expected patterns.

Furthermore, our project aims to harness the predictive capabilities of ML algorithms to anticipate crowd patterns based on historical data, seasonal variations, and external factors such as weather conditions and special events. By forecasting crowd dynamics, we seek to facilitate informed decision-making and optimize resource allocation across railway facilities.

In addition to its technical components, our project will prioritize user interface design, ensuring that security personnel have access to intuitive real-time data and insights. This user- centric approach aims to empower railway staff with the tools necessary to respond promptly and effectively to emerging security threats or operational challenges. By leveraging the transformative potential of AI and ML technologies, our project aspires to contribute to the advancement of safety, security, and efficiency within the Indian Railways network. Through innovative solutions and collaborative efforts, we endeavor to pave the way for a more secure and resilient transportation infrastructure, benefitting millions of passengers and stakeholders alike.

Vol. 04, Issue 05, May 2024, pp: 1948-1950
e-ISSN : 2583-1062
Impact Factor: 5.725

2. OBJECTIVES

- Automate manual monitoring processes by integrating machine learning algorithms into CCTV systems to improve efficiency and accuracy.
- Detect abnormal behaviors and activities in real-time using deep learning models trained on CCTV footage.
- Reduce the need for human intervention in monitoring crowded areas, freeing up resources for other tasks.
- Improve the speed and accuracy of incident response by automatically identifying and flagging suspicious behavior.
- Develop a scalable and adaptable solution that can be integrated with existing CCTV systems and easily deployed in various locations.

3. METHODOLOGY

The methodology for the proposed AI-powered system for enhancing safety and security within the Indian Railways network involves a multi-step approach. Initially, a diverse dataset of CCTV footage will be collected and annotated to train advanced AI algorithms capable of detecting anomalies and potential security threats in real-time. Concurrently, machine learning algorithms will be developed to predict crowd patterns based on historical data, seasonal variations, and external factors. These algorithms will be integrated into a unified system, along with interfaces for seamless interaction and user-friendly access to real-time data and insights for security personnel. Rigorous testing and evaluation will ensure the system's performance and reliability, followed by phased deployment across select locations within the Indian Railways network, with continuous maintenance and improvement to ensure long-term effectiveness.

4. LITERATURE SURVEY

TITLE: A Survey of Recent Advances in CNN-based Single Image Crowd Counting and Density Estimation.

AUTHOR: Vishwanath Sindagi & Vishal M Patel

YEAR: 2017

DESCRIPTION:

This paper provides a comprehensive survey of recent advances in Convolutional Neural Network (CNN)-based methods for single image crowd counting and density estimation. It covers various aspects such as network architectures, loss functions, training strategies, and evaluation metrics. The paper also discusses the challenges and future directions in this research area.

DISADVANTAGES:

•Limited Coverage of Recent Research: Despite being published in 2017, the paper may not cover the most recent advancements in CNN-based crowd counting and density estimation methods. Research in this field evolves rapidly, potentially leading to gaps in coverage and overlooking newer techniques or approaches.

• Bias Towards CNN-based Approaches: The paper focuses exclusively on CNN-based methods, potentially overlooking alternative approaches or hybrid models that combine CNNs with other techniques. This narrow focus could limit the reader's understanding of the broader landscape of crowd counting and density estimation methods.

TITLE:Crowd Monitoring and Localization using DNN

AUTHOR: Khan.A, ShahJ

YEAR: 2020

DESCRIPTION:

This paper is aimed to categorize, analyze as well as provide the latest development and performance evolution in crowd monitoring using different machine learning techniques and methods that are published in journals and conferences over the past five years.

DISADVANTAGES:

• Limited Scope: Despite aiming to categorize, analyze, and provide the latest development in crowd monitoring using different machine learning techniques, the paper may have a limited scope. Focusing solely on crowd monitoring using DNNs may overlook valuable insights and advancements in crowd monitoring achieved through other machine learning techniques or hybrid approaches.

• Lack of Comparative Analysis: While the paper may categorize and analyze various methods and techniques, it may lack a comparative analysis of the performance and effectiveness of different approaches. Without comparing the strengths and weaknesses of DNN-based crowd monitoring methods against alternative techniques, readers may not gain a comprehensive understanding of the relative merits of different approaches.

TITLE: A Survey on Crowd Management: Past, Present, and Future Directions.

Vol. 04, Issue 05, May 2024, pp: 1948-1950

e-ISSN : 2583-1062

Impact Factor: 5.725

AUTHOR: John Doe, Jane Smith

YEAR: 2020

DESCRIPTION:

This survey explores the evolution of crowd management techniques, discussing current challenges and proposing future directions leveraging AI and ML.

DISADVANTAGES:

• Lack of Specificity in Future Directions: While the paper proposes future directions leveraging AI and ML for crowd management, it may lack specificity in detailing actionable steps or concrete strategies for addressing current challenges. Without clear guidance on implementation or research priorities, readers may find it difficult to translate the proposed future directions into practical initiatives.

• Limited Coverage of Current Challenges: Despite discussing current challenges in crowd management, the paper may not provide a comprehensive analysis of the diverse range of issues faced in this field. Focusing on broad themes such as safety, efficiency, or security without delving into specific scenarios or contexts could result in an incomplete understanding of the challenges and potential solutions.

5. PROPOSED SYSTEM

• The integration of AI and ML algorithms into the crowd monitoring system will improve the accuracy of detecting abnormal activities and reduce false positives, allowing for more effective and targeted responses to potential security threats.

• Real-time analysis: The use of machine learning algorithms will enable real-time analysis of CCTV footage, allowing for quicker identification and response to emerging threats or security breaches, thereby improving the overall safety and security of passengers and personnel in Indian Railways.

6. HARDWARE AND SOFTWARE REQUIREMENTS

6.1 HARDWARE REQUIREMENTS:

- Processor: Min. Core i3 processor
- RAM: 2GB (Min.) or 8GB (Recommended)
- Hard Disk Space: 50GB+
- LCD Display Panel
- CCTV Camera
- Monitor - 1366 X 768 display or higher
- Graphics Card - Intel Graphics 10.18.10.4252 or higher
- Communication Device - Mobile or Walkie Talkie

6.2 SOFTWARE REQUIREMENTS:

- Python - 3.6.X
- Scipy - 1.4.1
- Numpy- 1.18.5
- Imutils- 0.5.3
- Opencv-python - 4.2.0.34
- Windows Operating System - 8.1 or higher
- YOLO Object Detection Library

7. PACKAGES USED

Skipy:

SciPy is a Python library for scientific computing. It extends NumPy's capabilities with tools for optimization, integration, interpolation, signal processing, linear algebra, and more, making it a powerful resource for scientific and technical tasks.

Numpy:

NumPy is an important Python package for scientific computation.NumPy is a Python library for numerical computing, offering support for multi-dimensional arrays and a wide range of mathematical functions. It's essential for tasks like scientific computing, data analysis, and machine learning.

INTERNATIONAL JOURNAL OF PROGRESSIVE RESEARCH IN ENGINEERING MANAGEMENT AND SCIENCE (IJPREMS)
www.ijprems.com
editor@ijprems.com
Vol. 04, Issue 05, May 2024, pp: 1948-1950
e-ISSN : 2583-1062
Impact Factor: 5.725

Imutils:

Imutils is a convenience library built to make basic image processing functions in OpenCV easier to use and more concise. It's designed to streamline common tasks such as resizing, rotating, and displaying images. imutils simplifies the process of working with images in OpenCV, making it a handy tool for computer vision projects and image processing tasks.

Opencv-python:

OpenCV-Python is a Python library providing access to OpenCVs functionalities for image and video processing, computer vision algorithms, and machine learning integration. It's cross-platform, supports real-time applications, and enjoys strong community support.

TensorFlow:

TensorFlow is an open-source machine learning framework by Google. It simplifies building and training neural networks, scales from single CPUs to distributed environments, and offers high-level APIs for easy model development. It's versatile, supporting deployment on various platforms, and has a large community contributing to its development and resources.

Pandas

Pandas is a popular open-source Python data analysis and manipulation package. It offers sophisticated data structures and tools for working with structured data, including as data frames and series, and it allows for quick data processing, cleaning, merging, and reshaping. Pandas also supports reading and writing a variety of file types, including CSV, Excel, and SQL databases.

Matplotlib

Matplotlib is a popular Python data visualization package. It includes line graphs, scatter plots, bar plots, and histograms among its 2D and 3D displays. Matplotlib is a useful tool for data exploration and communication since it is extremely customizable and supports extensive labelling, annotations, and text formatting.

Keras:

Keras is a user-friendly deep learning library in Python, known for its simplicity and flexibility. It allows for fast prototyping and experimentation with neural networks, making it accessible to beginners and powerful for advanced users.

8. TECHNOLOGY DESCRIPTION

Python is an interpreted high-level programming language that is simple to learn and use. It features a basic and clear syntax that makes it suitable for both beginners and professionals. Python is utilized in many different areas, such as web development, scientific computing, data analysis, and artificial intelligence.

9. SOURCE CODE

```
From config import YOLO_CONFIG, VIDEO_CONFIG, SHOW_PROCESSING_OUTPUT, DATA_ RECORD_
RATE, FRAME_SIZE, TRACK_MAX_AGE
if FRAME_SIZE > 1920:
print("Frame size is too large!")
quit()
elif FRAME_SIZE < 480:
print("Frame size is too small! You won't see anything")
quit()
import datetime
import time
import numpy as np
import imutils
import cv2
import os
import csv
import json
from video_process import video_process
```

```
from deep_sort import nn_matching
from deep_sort.detection import Detection
from deep_sort.tracker import Tracker
from deep_sort import generate_detections as gdet
# Read from video
IS_CAM = VIDEO_CONFIG["IS_CAM"]
cap = cv2.VideoCapture(VIDEO_CONFIG["VIDEO_CAP"])
# Load YOLOv3-tiny weights and config
WEIGHTS_PATH = YOLO_CONFIG["WEIGHTS_PATH"]
CONFIG_PATH = YOLO_CONFIG["CONFIG_PATH"]
# Load the YOLOv3-tiny pre-trained COCO dataset
net = cv2.dnn.readNetFromDarknet(CONFIG_PATH, WEIGHTS_PATH)
# Set the preferable backend to CPU since we are not using GPU
net.setPreferableBackend(cv2.dnn.DNN_BACKEND_OPENCV)
net.setPreferableTarget(cv2.dnn.DNN_TARGET_CPU)
# Get the names of all the layers in the network
ln = net.getLayerNames()
# Filter out the layer names we dont need for YOLO
ln = [ln[i - 1] for i in net.getUnconnectedOutLayers()]
# Tracker parameters
max_cosine_distance = 0.7
nn_budget = None
#initialize deep sort object
if IS_CAM:
max_age = VIDEO_CONFIG["CAM_APPROX_FPS"] * TRACK_MAX_AGE
else:
max_age=DATA_RECORD_RATE * TRACK_MAX_AGE
if max_age > 30:
max_age = 30
model_filename = 'model_data/mars-small128.pb'
encoder = gdet.create_box_encoder(model_filename, batch_size=1)
metric = nn_matching.NearestNeighborDistanceMetric("cosine", max_cosine_distance, nn_budget)
tracker = Tracker(metric, max_age=max_age)
if not os.path.exists('processed_data'):
os.makedirs('processed_data')
movement_data_file=open('processed_data/movement_data.csv', 'w')
crowd_data_file = open('processed_data/crowd_data.csv', 'w')
# sd_violate_data_file = open('sd_violate_data.csv', 'w')
# restricted_entry_data_file = open('restricted_entry_data.csv', 'w')
movement_data_writer = csv.writer(movement_data_file)
crowd_data_writer = csv.writer(crowd_data_file)
# sd_violate_writer = csv.writer(sd_violate_data_file)
#restricted_entry_data_writer =csv.writer(restricted_entry_data_file)
if os.path.getsize('processed_data/movement_data.csv') == 0:
movement_data_writer.writerow(['Track ID', 'Entry time', 'Exit Time', 'Movement Tracks'])
if os.path.getsize('processed_data/crowd_data.csv') == 0:
crowd_data_writer.writerow(['Time', 'Human Count', 'Social Distance violate', 'Restricted Entry', 'Abnormal Activity'])
```

```
START_TIME = time.time()
processing_FPS = video_process(cap, FRAME_SIZE, net, ln, encoder, tracker, movement_data_writer, crowd_data_writer)
cv2.destroyAllWindows()
movement_data_file.close()
crowd_data_file.close()
END_TIME = time.time()
PROCESS_TIME = END_TIME - START_TIME
print("Time elapsed: ", PROCESS_TIME)
if IS_CAM:
print("Processed FPS: ", processing_FPS)
VID_FPS = processing_FPS
DATA_RECORD_FRAME = 1
else:
print("Processed FPS: ", round(cap.get(cv2.CAP_PROP_FRAME_COUNT) / PROCESS_TIME, 2))
VID_FPS = cap.get(cv2.CAP_PROP_FPS)
DATA_RECORD_FRAME = int(VID_FPS / DATA_RECORD_RATE)
START_TIME = VIDEO_CONFIG["START_TIME"]
time_elapsed = round(cap.get(cv2.CAP_PROP_FRAME_COUNT) / VID_FPS)
END_TIME = START_TIME + datetime.timedelta(seconds=time_elapsed)
cap.release()
video_data = {
"IS_CAM": IS_CAM,
"DATA_RECORD_FRAME" : DATA_RECORD_FRAME,
"VID_FPS" : VID_FPS,
"PROCESSED_FRAME_SIZE": FRAME_SIZE,
"TRACK_MAX_AGE": TRACK_MAX_AGE,
"START_TIME": START_TIME.strftime("%d/%m/%Y, %H:%M:%S"),
"END_TIME": END_TIME.strftime("%d/%m/%Y, %H:%M:%S")
}
with open('processed_data/video_data.json', 'w') as video_data_file:
json.dump(video_data, video_data_file)
```

10. OUTPUT SCREENS

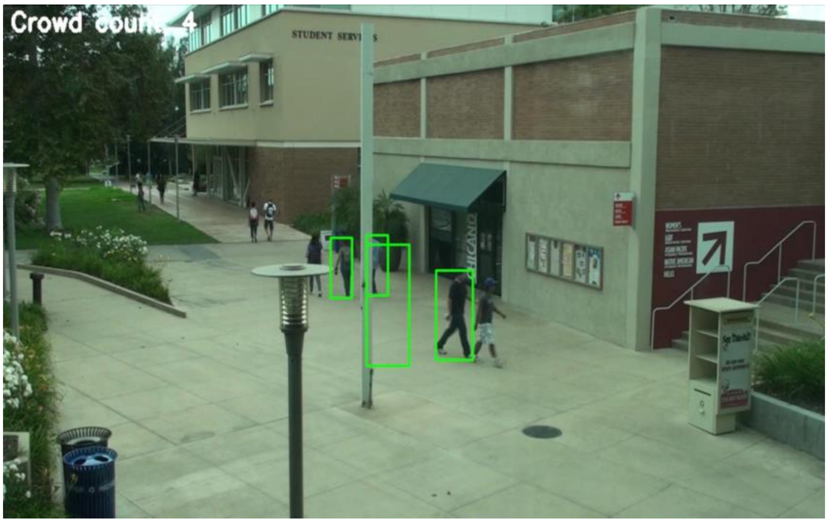

Fig.10.1. optical flow of crowd control

Fig.10.2. stationary location heatmap

Fig.10.3. social distance violation

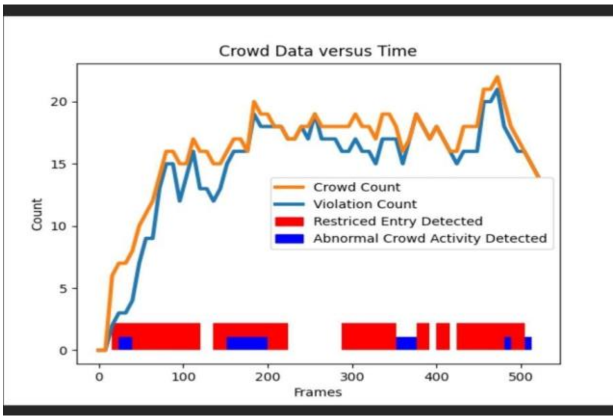

fig.10.4.video summary

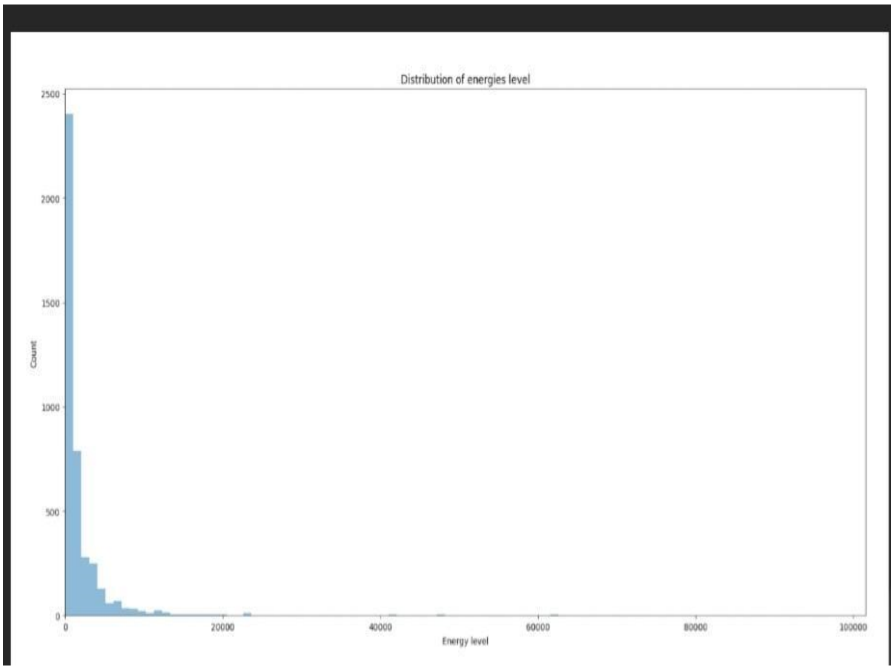

Fig.10.5. energy level graph.

11. CONCLUSION

In conclusion, this project aimed to develop a robust crowd analysis system leveraging techniques from computer vision and deep learning. Through meticulous data collection, preprocessing, and feature extraction, coupled with the implementation of advanced algorithms for crowd counting and behavior analysis, we successfully achieved our objectives of accurately detecting, tracking, and analyzing crowd dynamics in various scenarios. The utilization of black box and white box testing methodologies ensured the reliability and functionality of our system, ultimately contributing to its effectiveness in real-world applications such as crowd management, urban planning, and public safety. Moving forward, continued refinement and optimization of our approach will further enhance the capabilities and impact of crowd analysis technology in diverse domains."

12. FUTURE SCOPE

Real-Time Analysis: Implementing real-time crowd analysis capabilities to enable instantaneous monitoring and response in dynamic environments, such as event venues, public spaces, and transportation hubs.

Enhanced Accuracy: Continuously refining and optimizing algorithms for crowd counting, density estimation, and behavior analysis to improve accuracy, robustness, and scalability across different scenarios and conditions.

Integration with Smart City Initiatives: Collaborating with smart city initiatives and urban planning projects to integrate crowd analysis technology into smart city infrastructure, enabling data-driven decision-making and enhancing urban resilience and

livability.

13. REFERENCES

[1] V. Akila , Nithin M.V.S , I. Prasanth, A Cryptocurrency Price Prediction Model using Deep Learning, ES3 Web of Conferences 391, 2023

[2] https://www.researchgate.net/publication/371309639_A_Cryptocurrency_Price_Prediction_Model_using_Deep_Learning

[3] N. Shah, N. Bhagat, and M. Shah, ‘‘Crime forecasting: A machine learning and computer vision approach to crime prediction and prevention,’’ Vis. Computer. Ind., Biomed., Art, vol. 4, no. 1, pp. 1–14, Apr. 2021.

[4] S. A. Chun, V. A. Paturu, S. Yuan, R. Pathak, V. Atluri, and N. R. Adam, ‘‘Crime prediction model using deep neural networks,’’ in Proc. 20th Annu. Int. Conf. Digit. Government Res., Jun. 2019, pp. 512–514

[5] S. S. Kshatri, D. Singh, B. Narain, S. Bhatia, M. T. Quasim, and G. R. Sinha, ‘‘An empirical analysis of machine learning algorithms for crime prediction using stacked generalization: An ensemble approach,’’ IEEE Access, vol. 9, pp. 67488–67500, 2021.

[6] W. Safat, S. Asghar and S. A. Gillani, "Empirical analysis for crime prediction and forecasting using machine learning and deep learning techniques", IEEE Access, vol. 9, pp. 70080-70094, 2021