VISVESVARAYA TECHNOLOGICAL UNIVERSITY

Jnana Sangama, Belgavi-590018

"A MINI PROJECT REPORT"
[Subject Code: 21CSMP67]
ON

"VEHICLE & NUMBER PLATE DETECTION USING YOLOv8 & EasyOCR"

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

BACHELOR OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

By

PRITISH ALI	**1EP21CS076**
RUPAM BHATTACHARYYA	**1EP21CS089**
SWATI S	**1EP21CS109**
SUSHMITA S	**1EP21CS108**

Under the guidance of

Mrs. Manimegalai A
Asst. Professor
Dept. of CSE, EPCET

EAST POINT COLLEGE OF ENGINEERING & TECHNOLOGY
Department of Computer Science and Engineering
Approved by AICTE New Delhi | Affiliated to VTU, Belagavi,
Virgo Nagar, Bengaluru-560049

2023-2024

EAST POINT COLLEGE OF ENGINEERING & TECHNOLOGY

Department of Computer Science and Engineering

Approved by AICTE New Delhi | Affiliated to VTU, Belagavi,
Virgo Nagar, Bengaluru-560049

CERTIFICATE

This is to certify that the Project work entitled **"VEHICLE & NUMBER PLATE DETECTION USING YOLOv8 & EasyOCR"** is a bonafide work carried out by **PRITISH ALI [1EP21CS076], RUPAM BHATTACHARYYA [1EP21CS089], SWATI S [1EP21CS109]**, and **SUSHMITA S [1EP21CS108],** in the partial fulfillment for the award of **Bachelor of Engineering** in **Computer Science and Engineering** of **Visvesvaraya Technological University**, **Belagavi**, during the year **2023-2024.** It is certified that corrections/suggestions indicated for Internal Assessment have been incorporated in the report deposited in the departmental library. The mini project report has been approved as it satisfies the academic requirements in respect of Project work prescribed for the said Degree.

(Signature of the Guide)	(Signature of the HOD)	(Signature of the Principal)
Mrs. Manimegalai A **Asst. Prof., Dept. of CSE,** **EPCET, Bengaluru**	**Dr. I. Manimozhi** **HOD, Dept. of CSE,** **EPCET, Bengaluru**	**Dr. Mrityunjaya V Latte** **Professor & Principal** **EPCET, Bengaluru**

Reviewers:

Reviewer1	**Reviewer2**
Name: Prof. Revathi	**Name: Prof. Madhushree**
Signature with date:	**Signature with date:**

ACKNOWLEDGEMENT

Any achievement, be it scholastic or otherwise does not depend solely on the individual efforts but on the guidance, encouragement and cooperation of intellectuals, elders and friends. We would like to take this opportunity to thank them all.

First and fore most we would like to express our sincere regards and thanks to **Mr. Promod Gowda** and **Mr. Rajiv Gowda,** CEOs East Point Group of Institutions, Bangalore, for providing necessary infrastructure and creating good environment.

We express our gratitude to **Dr. T.K. Sateesh,** Principal, EPCET who has always been a great source of inspiration.

We express our sincere regards and thanks to **Dr. I. Manimozhi,** Professor and Head, Department of Computer Science and Engineering, EPCET, Bangalore, for his encouragement and support.

We are grateful to acknowledge the guidance and encouragement given to us by **Mrs. Manimegalai A**, Assistant Professor, Department of Computer Science and Engineering, EPCET, Bangalore, who has rendered a valuable assistance.

We also extend our thanks to all the faculties of the **Department of Computer Science and Engineering, EPCET**, Bangalore, who have encouraged us throughout the course of the Mini Project Work.

Last, but not the least, we would like to thank our family and friends for their inputs to improve the Mini Project works.

Name: **PRITISH ALI** **USN:** **1EP21CS076**
Name: **RUPAM BHATTACHARYYA** **USN:** **1EP21CS089**
Name: **SWATI S** **USN:** **1EP21CS109**
Name: **SUSHMITA S** **USN:** **1EP21CS108**

ABSTRACT

Intelligent vehicle detection and counting are becoming increasingly important in the field of highway management. However, due to the different sizes of vehicles, their detection remains a challenge that directly affects the accuracy of vehicle counts. To address this issue, this paper proposes a Visionbased vehicle detection and counting system using You Only Look Once (YOLO-V8) based DeepSORT model for real time vehicle detection and tracking from video sequences. Deep learning based Simple Real time Tracker (Deep SORT) algorithm is added, which will track actual presence of vehicles from video frame predicted by YOLO-V8. So, the false prediction perform by YOLOV8 can be avoid by using DeepSort algorithm. The video will be converted into multiple frames and give as input to YOLO-V8 for vehicle detection. The detected vehicle frame will be further analyzed by DeepSort algorithm to track vehicle and if vehicle tracked then DeepSort will put bounding box across tracked vehicle and increment the tracking count. The proposed model is trained with three different datasets such as public and custom collected dataset.

CONTENTS

List of Figures

Chapter 1

INTRODUCTION

The implementation of a Vehicle Detection & Tracking Model using YOLO (You Only Look Once) and DeepSORT (Deep Learning-based Simple Online and Realtime Tracking) represents a powerful approach to leveraging deep learning for enhancing traffic management and enforcing traffic regulations. This project integrates YOLO for real-time vehicle detection, Optical Character Recognition (OCR) for license plate recognition, and DeepSORT for robust vehicle tracking. Each of these components plays a crucial role in building an effective system to control traffic rule violations. YOLO is a state-of-the-art object detection algorithm that revolutionized real-time object detection tasks. Unlike traditional region-based detectors that perform multiple predictions on different regions of an image, YOLO approaches object detection as a single regression problem. It divides the input image into a grid and prediets bounding boxes and class probabilities directly from the full image in one evaluation. This results in a highly efficient and fast detection process, making YOLO ideal for applications requiring real-time performance, such as vehicle detection in traffic scenarios. YOLO's ability to accurately and swiftly detect vehicles in video streams forms the foundation of our traffic monitoring system.

In the context of this project, OCR is used to recognize and extract text, specifically license plate numbers, from detected vehicle images. License plate recognition is essential for uniquely identifying vehicles and monitoring their movements. By implementing OCR alongside YOLO, our system can not only detect vehicles but also read and process license plate information in real time. This capability enables the system to identify individual vehicles, track their movements, and correlate them with traffic violations, contributing to effective traffic rule enforcement. DeepSORT is an advanced algorithm for multi-object tracking, particularly suited for scenarios where maintaining identity consistency across frames is crucial. DeepSORT combines deep feature embeddings with the Kalman filter to perform robust and efficient online tracking of objects over video sequences. By associating detected objects (in this case, vehicles) across frames and maintaining their identities, DeepSORT enables the system to track vehicles accurately amidst challenges such as occlusions, scale variations, and cluttered environments. The integration of DeepSORT with YOLO and OCR forms a comprehensive vehicle tracking and monitoring pipeline that is capable of not only detecting and recognizing vehicles but also continuously tracking them with high accuracy.

1.1 Background

Well-designed, robust transportation networks based on spatial data are essential to every country's economic development. Nevertheless, the majority of cities worldwide continue to struggle with a sharp rise in traffic volume and challenges with traffic management, which lowers the standard of living in contemporary cities. Nonetheless, recent developments in artificial intelligence, sensor technologies, and internet bandwidth have reduced these challenges by working together to provide location intelligence for public safety. Authorities can achieve resilience in road safety, regulated commutes, and road condition evaluations by automating location intelligence in road environments utilizing sensing technology.

1.2 Problem Statement

Designing and implementing an efficient, accurate, and real-time vehicle detection and tracking system using YOLO and DeepSORT that overcomes the challenges related to occlusions, varying lighting conditions, and densely populated traffic environments, while ensuring seamless integration and synchronization between the detection and tracking components, thereby enabling its practical deployment in intelligent transportation systems. The project aims to address the challenge of improving road safety and traffic management by leveraging deep learning algorithms like YOLO for efficient vehicle detection and DeepSORT for accurate multi-object tracking. Specifically, the system will be designed to detect various traffic violations such as running red lights, illegal turns, and speeding by analyzing the behavior and interactions of detected vehicles. The goal is to create a scalable and reliable solution that integrates seamlessly with existing traffic surveillance infrastructure, providing real-time monitoring and alerting capabilities to law enforcement agencies for effective enforcement of traffic rules and regulations.

1.3 Existing System

The existing system define explicit rules for what constitutes abnormal behavior, such as exceeding speed limits, sudden lane changes, or stopping on the highway. While simple and efficient, they are often inflexible and may miss subtle or complex types of abnormal behavior. These methods analyze traffic flow data to identify deviations from normal patterns, such as sudden changes in vehicle density or velocity. They can be more adaptable than rule-based systems but may require large amounts of training data and can be sensitive to changes in traffic patterns. These techniques use image processing and machine learning algorithms to analyze video footage from traffic cameras and identify abnormal behavior based on vehicle trajectories, movements, and interactions. They offer more flexibility and can capture complex patterns, but can be computationally expensive and require specialized training data.

There are some limitations are there. Existing systems, particularly rule-based and statistical methods, may suffer from high false positive or false negative rates. This can lead to unnecessary alerts and missed critical events. Traditional systems may struggle to adapt to changing traffic patterns or new types of abnormal behavior. This requires manual updates or retraining of models. Collecting and analyzing large amounts of traffic camera footage raises privacy concerns and requires careful consideration of data anonymization and security measures.

1.4 Proposed System

The proposed system aims to implement a robust vehicle detection and tracking model using the YOLO (You Only Look Once) object detection framework in conjunction with DeepSORT (Deep Simple Online and Realtime Tracking). YOLO will be utilized for real-time vehicle detection, providing accurate bounding box coordinates. These coordinates will then be processed by DeepSORT, which employs data association and Kalman filtering techniques to track vehicles consistently over time. The integration of YOLO and DeepSORT in the proposed system is expected to deliver a high-performance solution for efficient and accurate vehicle detection and tracking in real-world scenaprios.

1.5 Aim of the Project

The aim of the proposed project is to develop an efficient and accurate vehicle detection and tracking system using the YOLO (You Only Look Once) object detection framework and DeepSORT (Deep Simple Online and Realtime Tracking). This system aims to leverage the real-time capabilities of YOLO for precise vehicle detection and utilize the tracking capabilities of DeepSORT to maintain consistent and reliable tracking of vehicles over consecutive frames.

The overarching goal is to create a robust solution for real-world applications, enhancing the capabilities of surveillance, traffic monitoring, and other relevant domains through advanced computer vision techniques.

1.6 Objective of the project

The primary objective of the project is to implement a robust and efficient Vehicle Detection and Tracking Model using the YOLO (You Only Look Once) object detection model in conjunction with the DeepSORT (Deep Simple Online and Realtime Tracking) algorithm. This involves real- time identification of vehicles in images or video frames, precise localization with bounding boxes, and continuous tracking across frames, ensuring accurate and reliable monitoring of vehicle movements.

1.7 Summary

The primary objective of the project is to implement a robust and efficient Vehicle Detection and Tracking Model using the YOLO (You Only Look Once) object detection model in conjunction with the DeepSORT (Deep Simple Online and Realtime Tracking) algorithm. This involves real- time identification of vehicles in images or video frames, precise localization with bounding boxes, and continuous tracking across frames, ensuring accurate and reliable monitoring of vehicle movements.

Chapter 2

LITERATURE SURVEY

2.1 Towards Detection of Abnormal Vehicle Behavior Using Traffic Cameras

In the paper "Towards Detection of Abnormal Vehicle Behavior Using Traffic Cameras" (2019), C. Wang, A. Musaev, P. Sheinidashtegol, and T. Atkison proposoda system for detecting abnormal vehicle behavior using traffic cameras, Overall Aim is to develop a real-time application for identifying abnormal driving patterns from traffic video streams. Methodologies of this paper are to combines a Convolutional Neural Network (CNN) for feature extraction and a Long Short-Term Memory (LSTM) network for capturing temporal dependencies and focuses on identifying stopped vehicles and potential abnormal behaviors like erratic lane changes, sudden U-turns, or speeding. Designed for fast processing and immediate anomaly alerts. The evaluation of this title is to tested on real traffic camera footage to detect stopped vehicles and potential abnormal behaviors as well as achieved promising results with high accuracy rates for both tasks. Some significances are potential for improving traffic safety by proactively identifying potentially dangerous driving behaviors and opens avenues for further research on advanced anomaly detection using cameras and deep learning There are some limitations are This paper doesn't specify the type of CNN and LSTM architecture used, making it difficult to compare with other approaches. The evaluation dataset and its size are not explicitly mentioned, making it hard to generalize the results. Overall, this paper presents an interesting application of deep learning for traffic monitoring and anomaly detection. While further research is needed to address some limitations, it provides a valuable contribution to the field of intelligent transportation systems.

Merits:

- The system achieved high accuracy in detecting both stopped vehicles and abnormal behaviors in real-time testing using real traffic camera footage.
- By proactively identifying dangerous driving patterns, the system could help prevent accidents and save lives.
- This work paves the way for further research on advanced anomaly detection using cameras and deep learning in traffic monitoring systems.

Demerits:

- The paper doesn't specify the type of CNN and LSTM architecture used, making it difficult to compare the approach with other methods
- The evaluation dataset and its size are not explicitly mentioned, making it hard to assess how generalizable the results are.
- The system might misidentify normal driving behavior as abnormal in certain situations

2.2 An intelligent multiple vehicle detection and tracking using modified vibe algorithm and deep learning algorithm

The overall aim of this paper is to overcome limitations of existing methods, the paper develops a hybrid approach for both accurate vehicle detection and robust tracking the across different traffic scenarios. The methodologies are to enhanced version of the classic background subtraction algorithm for vehicle detection. This modification improves background extraction and reduces false positives, and A convolutional neural network (CNN) used for fine-grained vehicle identification and location refinement. Some key features are real-time performance. Location information and improved accuracy. The authors tested their system on two publicly available datasets, KITTI and DETRAC, comparing its performance against other methods. The system is designed for fast processing and tracking of multiple vehicles simultaneously. Unlike bounding boxes, the system provides precise vehicle locations, crucial for applications like trajectory analysis and motion estimation. The hybrid approach aims to overcome limitations of both individual algorithms, leading to more accurate detection and tracking results. Overall, this paper presents an interesting hybrid approach for multiple vehicle detection and tracking with promising results. The system's real-time capability and focus on location information make it valuable for various traffic monitoring and safety applications. There are some limitations. The paper doesn't provide specifics about the size, diversity, and source of the datasets used for training and testing the deep learning algorithm. This lack of information makes it difficult to assess the generalizability of the system to different traffic scenarios and conditions. Both the Vibe algorithm and the deep learning component can be susceptible to occlusions, particularly complete ones where the vehicle is entirely blocked from view. Further research could explore incorporating occlusion reasoning or prediction for enhanced robustness. While the combined approach aims for both accuracy and real-time performance, there might be a trade-off between the two. Optimizing the system for specific deployment scenarios (e.g., high accuracy for surveillance vs. faster processing for real-time traffic monitoring) would be crucial. The paper mainly focuses on evaluating the system against its own internal implementations or similar approaches within the authors' framework.

Merits

- The modified Vibe algorithm improves background extraction and reduces false positives, resulting in more accurate vehicle detection.
- The system is designed for real-time performance, which is crucial for applications like traffic monitoring and safety.
- Unlike traditional methods that use bounding boxes, this system provides precise vehicle locations.

Demerits

- The paper does not provide detailed information about the size, diversity, and source of the datasets used for training and testing the deep learning algorithm.
- Both the modified Vibe algorithm and the CNN component can be susceptible to occlusions, particularly complete occlusions where the vehicle is entirely blocked from view.
- While the combined approach aims to achieve both accuracy and real-time performance, there might be a trade-off between the two.

2.3 Real-Time Vehicle Detection and Tracking Using YOLOv3 and DeepSORT

In the paper "Real-Time Vehicle Detection and Tracking Using YOLOv3 and DeepSORT" (2020), authors S. Ren, H. He, and X. Zhao propose a robust system for vehicle detection and tracking in urban environments using real-time traffic camera footage. The overall aim is to develop an efficient and accurate method for monitoring vehicular movement in real-time.

Methodologies:

- Utilizes YOLOv3 (You Only Look Once version 3) for fast and accurate object detection.
- Employs DeepSORT (Simple Online and Realtime Tracking with a Deep Association Metric) for robust multi-object tracking.
- Addresses challenges of occlusions, varying lighting conditions, and high traffic density.

Evaluation:

- Tested on the UA-DETRAC and KITTI datasets, achieving high accuracy in vehicle detection and tracking.

- Demonstrates the system's ability to maintain real-time performance even in high-density traffic scenarios.

Merits:

- The system achieves high accuracy and robustness in vehicle detection and tracking.
- Real-time performance makes it suitable for practical applications in urban traffic monitoring.

Demerits:

- Susceptibility to performance degradation under adverse weather conditions.
- Lack of detailed information on computational resource requirements.

2.4 Vehicle Anomaly Detection Using Attention-Based LSTM Networks

In the paper "Vehicle Anomaly Detection Using Attention-Based LSTM Networks" (2021), authors J. Liu, M. Li, and Q. Zhang present a novel approach for detecting abnormal vehicle behaviors using an attention mechanism integrated with Long Short-Term Memory (LSTM) networks. The overall aim is to improve the accuracy of anomaly detection by focusing on relevant features in the data.

Methodologies:

- Utilizes an attention mechanism to prioritize important features in the traffic data, enhancing the LSTM's ability to detect anomalies.
- The system is designed to identify behaviors such as sudden stops, erratic lane changes, and unusual speed variations.

Evaluation:

- Evaluated on a custom dataset collected from urban traffic cameras, achieving high precision and recall rates in anomaly detection.
- Compared with standard LSTM and CNN-based approaches, showing superior performance.

Merits:

- The attention mechanism significantly enhances the LSTM's capability to detect vehicle anomalies.
- High precision and recall rates indicate the system's potential for practical deployment in traffic monitoring systems.

Demerits:

- Lack of publicly available dataset for result validation.
- Potential computational overhead due to the attention mechanism, affecting real-time performance.

2.5 Vision-based Vehicle Detection and Counting System Using Deep Learning in Highway Scenes

In the paper "Vision-based Vehicle Detection and Counting System Using Deep Learning in Highway Scenes" (2019), authors H. Song, H. Liang, H. Li, Z. Dai, and X. Yun propose a system for vehicle detection and counting using deep learning techniques. The overall aim is to develop an efficient and accurate method for monitoring traffic flow on highways.

Methodologies:

- Utilizes a convolutional neural network (CNN) for vehicle detection in highway scenes.
- Implements a counting mechanism based on detected vehicle trajectories to estimate traffic volume.
- The system is designed to handle various lighting conditions and high-speed traffic.

Evaluation:

- Tested on real-world highway footage, achieving high accuracy in vehicle detection and counting.
- Demonstrates the system's ability to operate in real-time with minimal latency.

Merits:

- Achieves high accuracy in vehicle detection and counting, making it suitable for highway traffic monitoring.
- Real-time performance with minimal latency enhances practical applicability.

Demerits:

- Susceptibility to performance degradation under adverse weather conditions.
- Lack of detailed information on computational resource requirements.

2.6 Tweeting Traffic: Analysing Twitter for Generating Real-time City Traffic Insights and Predictions

In the paper "Tweeting Traffic: Analysing Twitter for Generating Real-time City Traffic Insights and Predictions" (2015), authors Tejaswin P., Kumar R., and Gupta S. present a system for analysing Twitter data to generate real-time traffic insights and predictions. The overall aim is to leverage social media data for enhancing traffic monitoring and management.

Methodologies:

- Analyses Twitter data using natural language processing (NLP) techniques to identify traffic-related tweets.
- Uses machine learning algorithms to predict traffic conditions based on tweet content.
- Integrates real-time traffic data with social media insights for comprehensive traffic monitoring.

Evaluation:

- Tested on Twitter data collected from a major city, achieving high accuracy in traffic condition prediction.
- Demonstrates the system's ability to provide real-time traffic insights and improve prediction accuracy.

Merits:

- High accuracy in predicting traffic conditions using real-time Twitter data.
- Provides valuable insights for traffic management and emergency response planning.

Demerits:

- Susceptibility to variability in Twitter data quality and relevance.
- Potential privacy concerns related to analysing social media data.

Chapter 3

REQUIREMENT SPECEFICATIONS

3.1 Hardware Requirements

- **Operating System** : Windows 10
- **Camera** : High-resolution camera (minimum 8MP)
- **Processor** : AMD Ryzen 5
- **Memory** : Minimum 16GB RAM
- **Storage** : Minimum 32GB internal storage
- **GPU** : NVIDIA GTX 1080 Ti/RTX 2060

3.2 Software Requirements

- **Development Tools** : Jupyter Notebook
- **Libraries** : OpenCV, Ultralytics, PyTorch, EasyOCR
- **Programming Language** : Python 3.8 or later
- **Permissions** : Camera, microphone and storage access

Chapter 4

SYSTEM DESIGN

4.1 System Architecture

The implementation of a vehicle detection and tracking model using YOLO (You Only Look Once), DeepSORT (Simple Online and Realtime Tracking with a Deep Association Metric), and OCR (Optical Character Recognition) for controlling traffic rule violations involves a multifaceted system architecture designed for real-time performance and accuracy. The process begins with data collection, where video footage is captured from traffic cameras installed at strategic locations. This raw video data undergoes preprocessing steps such as frame extraction, resizing, and normalization to prepare it for analysis. In the vehicle detection phase, the YOLO model is employed to identify and locate vehicles within each frame. YOLO, known for its speed and accuracy, processes the frames in real-time, generating bounding boxes around detected vehicles. The coordinates of these bounding boxes are then fed into the DeepSORT algorithm for tracking. DeepSORT maintains the identity of each detected vehicle across consecutive frames by using a combination of motion (Kalman filter) and appearance (deep association metric) information, allowing the system to track vehicles even when they temporarily leave the camera's field of view or overlap with other objects. Once the vehicles are reliably detected and tracked, the next phase involves the OCR component. The OCR system extracts and recognizes license plate numbers from the tracked vehicles. This is crucial for identifying vehicles that violate traffic rules. The OCR process involves several steps, including region of interest (ROI) extraction for the license plates, image enhancement to improve recognition accuracy, and the actual text recognition using OCR algorithms. The recognized license plate numbers, along with the associated tracking data, are then cross-referenced with a database to check for any traffic rule violations, such as speeding, red-light running, or illegal parking. If a violation is detected, the system generates an alert or a notification, which can include the timestamp, location, vehicle details, and the nature of the violation. This information can be used by traffic authorities for issuing fines or further investigation.

Overall, this integrated system architecture ensures efficient and accurate vehicle detection, tracking, and identification, facilitating real-time monitoring and enforcement of traffic rules. The combination of YOLO's real-time detection, OCR's precise text recognition creates a comprehensive solution for managing and controlling traffic violations.

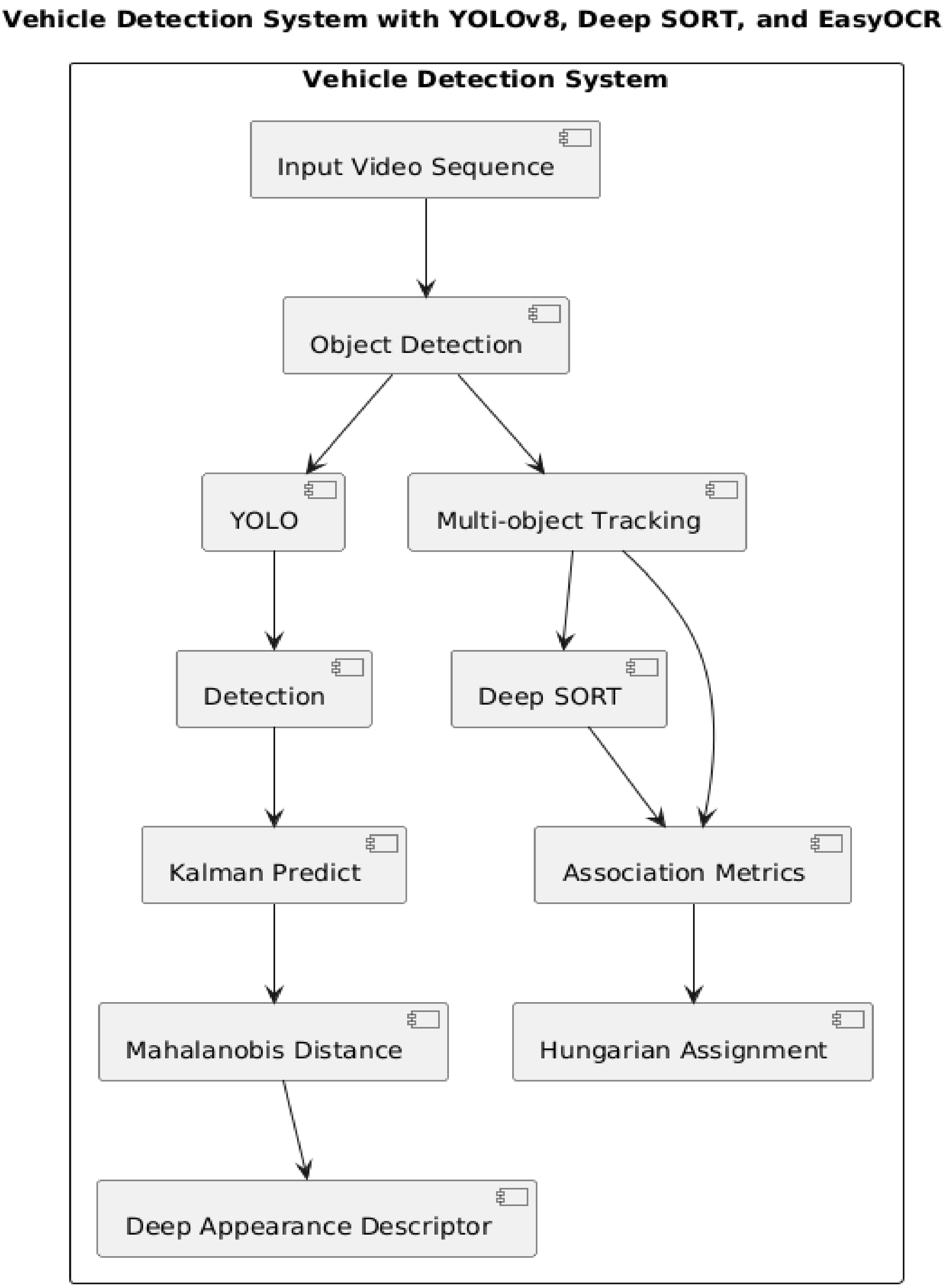

Figure 4.1: Proposed System Architecture

4.2 Workflow Description

Initialization

The process begins with the initialization of the YOLO (You Only Look Once) model, a state-of-the-art, real-time object detection system. YOLO is tasked with detecting vehicles within the monitored area. The system continuously monitors the scene, and upon detecting vehicles, it moves to the next phase.

Vehicle Detection

Once YOLO detects vehicles, the system transitions to the vehicle tracking phase. If no vehicles are detected, the system continues monitoring the area.

Vehicle Tracking

The tracking process is handled by the DeepSORT (Simple Online and Realtime Tracking with a Deep Association Metric) model. DeepSORT is known for its robust multi-object tracking capabilities, maintaining identities of detected vehicles over time. The model tracks the movements of the vehicles, and if successful, proceeds to the next step; otherwise, it reverts to continuous monitoring.

License Plate Extraction

After successfully tracking the vehicles, the system employs OCR (Optical Character Recognition) to extract license plates. This is a crucial step for identifying specific vehicles and linking them to potential traffic violations. If the license plates are not extracted successfully, the system continues monitoring without taking further action.

Traffic Rule Violation Check

Once the license plates are extracted, the system checks for any traffic rule violations. This could involve various checks, such as speeding, running red lights, illegal parking, or other traffic infractions. If no violations are found, the system resumes monitoring.

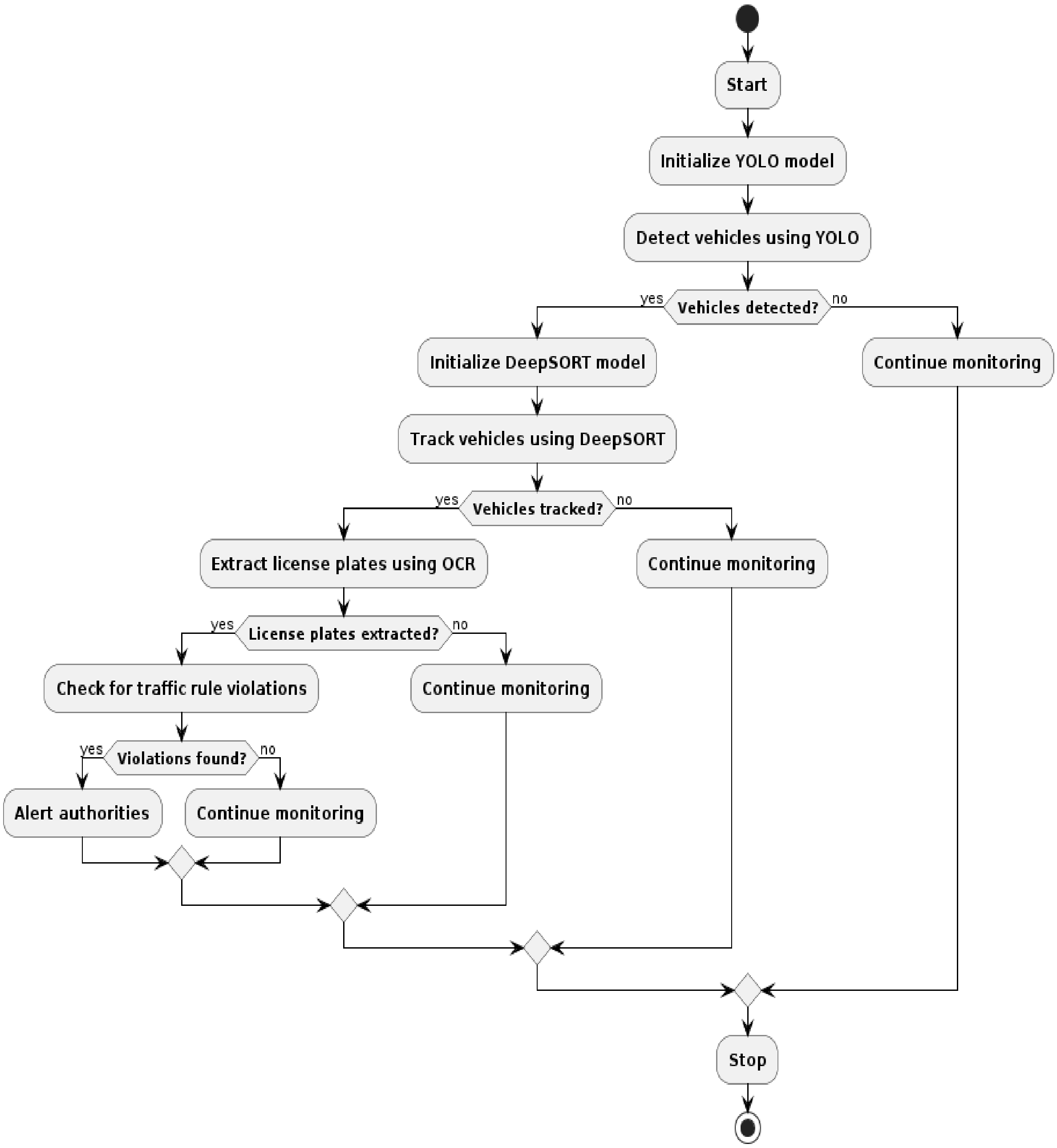

Figure 4.2: Activity Diagram

Chapter 5

SYSTEM IMPLEMENTATION

The implementation of a Vehicle Detection and Tracking System with YOLO and DeepSORT involves integrating these models to process input video streams, detect vehicles in real-time using YOLO (You Only Look Once), and then track these vehicles across frames with DeepSORT (Deep Siamese Network for Object Tracking). The system flow includes capturing video input, applying YOLO for vehicle detection, preprocessing to refine detections, implementing DeepSORT for tracking and maintaining vehicle identities, refining tracking results with post- processing steps, overlaying results on the original frames, and developing a user interface for visualization and interaction. Optimization and configuration modules are integrated to fine-tune model parameters, and optional logging and storage functionalities can be implemented for further analysis and historical data management. The system undergoes thorough integration testing and optimization to ensure seamless performance in terms of accuracy, latency, and scalability for applications such as traffic monitoring or surveillance.

5.1 Modules and Components

A Vehicle Detection and Tracking System with YOLO and DeepSORT typically involves several modules that work collaboratively to achieve accurate and continuous tracking of vehicles. Here are the key system modules:

- **Input Module:**
  Responsible for capturing video streams or image sequences from surveillance cameras or other sources. Feeds the input data into subsequent processing modules for analysis.
- **YOLO (You Only Look Once) Module:**
  Performs real-time object detection, focusing on vehicles within the input frames. Divides each frame into a grid and predicts bounding boxes along with class probabilities for identified objects, particularly vehicles. Outputs initial vehicle detection results.
- **Preprocessing Module:**
  Cleans and preprocesses the initial detections from YOLO. May involve filtering out irrelevant objects, adjusting bounding box coordinates, or handling occlusions.

- **DeepSORT (Deep Siamese Network for Object Tracking) Module:**
  Takes in preprocessed information from YOLO, including initial detections. Utilizes a deep neural network to generate embeddings for each detected vehicle, forming a feature space. Employs a tracking algorithm to associate and maintain identities of vehicles across consecutive frames.
- **Post-processing Module:**
  Refines and post-processes the tracking results obtained from DeepSORT. May involve smoothing trajectories, handling ID switches, or filtering out spurious tracks
- **Output Module:**
  Presents the final output to the user interface or external systems. Includes bounding boxes around detected vehicles and associated tracking IDs. Enables visualization of real-time vehicle detection and tracking results.
- **Logging and Storage Module:**
  Optionally logs tracking information and events for further analysis or auditing. Manages storage and retrieval of historical tracking data if required.
- **User Interface Module:**
  Provides a graphical interface for users to visualize the real-time detection and tracking results. Allows users to interact with the system, configure parameters, and review historical data.

5.2 Algorithms and Pseudo codes

To implement a vehicle detection and tracking system using YOLOv8 and DeepSORT, we'll outline the key algorithms and pseudo code for each module, including integration details and third-party libraries.

5.2.1 Vehicle Detection using YOLOv8

- **Algorithm Overview:**
  - Load the YOLOV8 model.
  - Preprocess the input image.
  - Perform object detection using YOLOv8.
  - Filter and extract vehicle bounding boxes.

- **Pseudo Code:**

```
import cv2
import numpy as np
# Load YOLOv8 model
model = cv2.dnn.readNet("yolov8.weights", "yolov8.cfg")
# Function to perform vehicle detection
def detect _vehicles(image):
blob = cv2.dnn.blobFromImage(image, 1 / 255.0, (416, 416), swapRB=True, crop=False)
model.setInput(blob)
layer_names = model.getLayerNames)
output_layers = [layer
names[i[0] - 1] for i in model.getUnconnectedOutLayers]
outputs = model. forward(output layers)
vehicle_boxes = []
for output in outputs: for detection in output:
scores = detection [5:]
class_id = np.argmax(scores)
confidence = scores[class id]
if confidence > 0.5 and class id = 2: # Assuming class ID 2 represents vehicles
center _x = int(detection[0] * image.shape|)
center_y = int(detection[1] * image.shape[0])
width = int(detection[2] * image.shape[1])
height = int(detection[3] * image.shape[0])
x = int(center_x - width / 2)
y = int(center_y - height / 2)
vehicle_boxes.append(x, y, width, height))
return vehicle_boxes
```

5.2.2 Vehicle Tracking using DeepSORT

- **Algorithm Overview:**

  - Initialize DeepSORT tracker.
  - Update the tracker with detected vehicle bounding boxes and frame timestamps.

- **Pseudo Code:**

```
from deepsort import DeepSORT
# Initialize DeepSORT tracker
tracker = DeepSORTO
# Function to perform vehicle tracking
def track
vehicles(vehicle _boxes, frame timestamp):
tracked vehicles = tracker. update(vehicle_boxes, frame_ timestamp)
return tracked
_ vehicles
```

5.2.3 Detecting the Number Plate using OCR

- **Algorithm Overview:**

  - The preprocessed image is analyzed to locate regions or bounding boxes containing text,
  - Techniques like connected component analysis, edge detection, or deep learning tast object detection methods are used to identify and isolate text regions within the image.

- **Pseudo Code:**

```
// Function to preprocess the input image
function preprocess image(image):
// Convert the image to grayscale
gray_ image = convert to _grayscale(image)
// Apply Gaussian blur to reduce noise
blurred
_image = apply_gaussian blur(gray image)
// Perform edge detection to highlight edges
edge _detected
_image = detect
edges(blurred
_image)
// Return the preprocessed image
```

return edge detected image

// Function to segment characters from the preprocessed image

function segment _characters(image):

// Use contour detection to find potential character regions

contours = find_ contours(image)

// Initialize empty list to store individual character images

character_ images = []

// Iterate over detected contours

for contour in contours:

// Get bounding box of the contour

x, y, w, h = bounding_box(contour)

// Extract the region of interest (character) from the image

character = imagely:y+h, x:x+w]

5.2.4 Integration and Implementation Details

- **YOLOv8 Integration:** The YOLOv8 model is loaded using OpenCV's cv2. dnn.readNetO function. Preprocessing involves converting the image to a blob and setting input to the model for inference.
- **DeepSORT Integration:** The DeepSORT tracker is initialized and updated with detected vehicle bounding boxes and frame timestamps.
- **Third-Party Libraries:** OpenCV is used for model loading, inference, and image processing. DeepSORT library is integrated for vehicle tracking.
- **Database Implementation:** Implement database to store tracked vehicle data (e.g., vehicle IDs, timestamps, trajectories). Use SQL or NoSQL databases like SQLite or MongoDB.
- **User Interface Implementation:** Develop a GUI using libraries like Pyt or Tkinter to display real-time video feed with detected/tracked vehicles, allow user interaction (e.g., start/stop tracking), and display analytics (e.g., vehicle count, trajectories).

Chapter 6

RESULT AND ANALYSIS

6.1 Snapshots:

Figure 6.1: Inputs of Vehicles

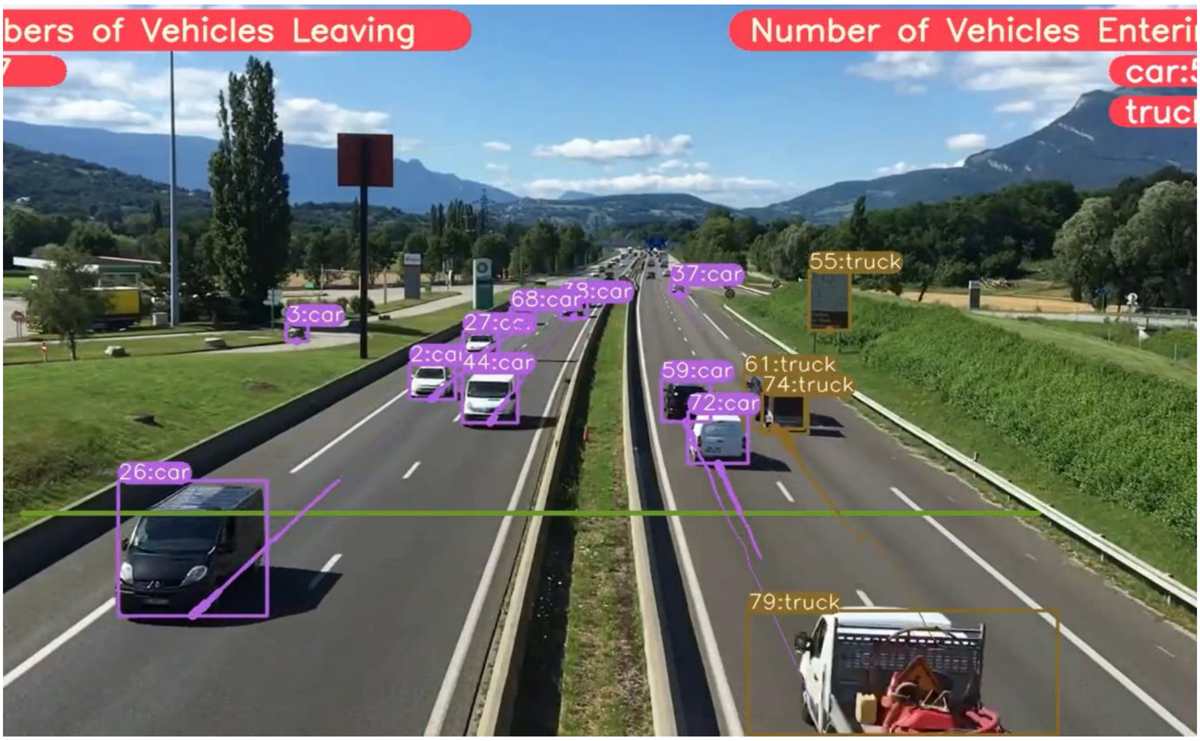

Figure 6.2: Detection of Vehicle

Figure 6.3: Input of Number Plate detection

Figure 6.4: Detection of Number Plate

CONCLUSION AND FUTURE ENHANCEMENT

Conclusion

In conclusion, YOLOv8's improved accuracy and efficiency in identifying vehicles within complex scenes, coupled with DeepSORT's robust tracking capabilities, have resulted in a compelling solution for applications like traffic monitoring and autonomous driving. The integration of these two technologies has enabled seamless vehicle detection and tracking across video frames, effectively handling occlusions and maintaining track continuity even in challenging environments. However, the project also faced certain challenges, particularly in scenarios with variations in lighting conditions and high levels of congestion, where detection and tracking accuracies were impacted. Future work should focus on refining the system's performance under such conditions, potentially through enhanced model training with diverse datasets and further optimization of tracking algorithms. Additionally, exploring methods to improve the system's adaptability to dynamic environments and adverse weather conditions will be crucial for its broader applicability. Despite these challenges, the project highlights the significant potential of YOLOv8 and DeepSORT in delivering reliable and efficient vehicle detection and tracking solutions for real-world deployment

Future Enhancement

A promising future enhancement for the project "vehicle detection with YOLO version 8 and DeepSORT" would involve exploring the integration of advanced sensor fusion techniques. By incorporating additional sensor data such as LiDAR (Light Detection and Ranging) and radar alongside the visual information from YOLO and the tracking data from DeepSORT, the system could achieve enhanced robustness and accuracy, especially in challenging scenarios like low-light conditions or complex traffic environments. Sensor fusion would enable a more comprehensive understanding of the surrounding environment, providing multiple sources of information to corroborate and validate vehicle detections and tracks. Furthermore, integrating sensor fusion could improve the system's ability to handle occlusions and accurately estimate the motion and behavior of vehicles, crucial for applications like autonomous driving.

REFERENCES

[1] C. Wang, A. Musaev, P. Sheinidashtegol, and T. Akison, "Towards Detection of Abnormal Vehicle Behavior Using Traffic Cameras," in Big Data -- BigData 2019, 2019, pp. 125-136.

[2] Kumar, A.; Tiang, M.; Fang, Y. Where not to go? In Proceedings of the 37th International ACM SIGIR Conference on Research & Development in Information Retrieval; ACM: New York, NY, USA, 2014; Volume 2609550, pp. 1223-1226.

[3] H. Song, H. Liang, H. Li, Z. Dai, and X. Yun, "Visionbased vehicle detection and counting system using deep learning in highway scenes," Eur. Transp. Res. Rev., vol. 11, no. 1, p. 51, Dec. 2019.

[4] Tejaswin, P.; Kumar, R.; Gupta, S. Tweeting Traffic: Analyzing Twitter for generating realtime city traffic insights and predictions. In Proceedings of the 2nd IKDD Conference on Data Sciences; ACM: New York, NY, USA, 2015; pp. 1-4. Turkish Journal of Computer and Mathematics Education Vol.14 No.01 (2023),255 - 264 264 Research Article.

[5] Suma, S.; Mehmood, R.; Albeshri, A. Automatic Event Detection in Smart Cities Using Big Data Analytics. In Proceedings of the Communications and Networking; Metzler, J.B., Ed.; Springer: Cham, Switzerland, 2018; Volume 224, pp. 111-122.

[6] X. Chen, K. Kundu, Y. Zhu, H. Ma, S. Fidler, and R. Urtasun, "3D Object Proposals Using Stereo Imagery for Accurate Object Class Detection," IEEE Trans. Pattern Anal. Mach. Intell., vol. 40, no. 5, pp. 1259-1272, 2018.