INSTITUTE OF ENGINEERING AND TECHNOLOGY, LUCKNOW

An Autonomous Constituent Institute of Dr. A.P.J. Abdul Kalam Technical University, Lucknow, U. P.

Session 2024-25

Internship Assessment

Report (KCS 752)

A report submitted in partial fulfilment of the requirements for the award of Degree of

BACHELOR OF TECHNOLOGY

COMPUTER SCIENCE AND ENGINEERING

Submitted By:

Name : Rajeev Kumar
Roll Number: 2100520200054
Submission Date: 23/12/2024

Submitted To:

Er. Shamsher Singh
Department of Computer Science & Engineering I. E. T. Lucknow

TABLE OF CONTENTS

CERTIFICATE

This is to certify that the “Internship report” submitted by **Rajeev Kumar** (Roll No.: **2100520200054**) is work done by him and submitted during the 2024 – 2025 academic year, in partial fulfilment of the requirements for the award of the degree of Bachelor of Technology in **Computer Science and Engineering** at Institute of Engineering and Technology (Lucknow, Uttar Pradesh).

Date: 23/12/2024

Place: Lucknow, U.P.

Er. Shamsher Singh
Course Coordinator
(KCS 752)

Prof. Girish Chandra
Head Of Department

LETTER OF INTERNSHIP

Dr. M. BRINDHA, M.E., Ph.D.
Associate Professor

राष्ट्रीय प्रौद्योगिकी संस्थान तिरुचिरापल्ली
NATIONAL INSTITUTE OF TECHNOLOGY TIRUCHIRAPPALLI
Department of Computer Science and Engineering
Tanjore Road, Tiruchirappalli - 620 015, Tamil Nadu, India.
Tel : +91 431 250 3218
Fax : +91 431 250 0133
E-mail : brindham@nitt.edu

30th July 2024

INTERNSHIP COMPLETION CERTIFICATE

This is to certify that the project titled **"Novel Vehicle Detection and Segmentation Techniques for Autonomous Vehicles"** is an internship record of the work done by:

RAJEEV KUMAR

B.Tech., Computer Science and Engineering, Institute of Engineering & Technology, Lucknow in partial fulfillment of the requirements for the completion of Summer Internship in **Computer Science and Engineering** of the **NATIONAL INSTITUTE OF TECHNOLOGY, TIRUCHIRAPPALLI,** during 1st June 2024 to 30th July 2024.

M. Brindha

With regards,

Dr. M. Brindha

Dr. M. BRINDHA, M.E., Ph.D.
Associate Professor
Department of Computer Science and Engineering
National Institute of Technology, Tiruchirappalli
Tamil Nadu - 620 015, India.

ACKNOWLEDGEMENT

This report is the outcome of a 2-months **Deep Learning Internship from 1st June 2024 to 31st July 2024**, successfully completed at **NIT Trichy** as part of the requirements for my Bachelor of Technology program in Computer Science and Engineering at I.E.T. Lucknow.

I wish to express my sincere and wholehearted gratitude to the entire team at **NIT Trichy** for providing me with this incredible opportunity. I thoroughly enjoyed the experience and learned a great deal. My heartfelt thanks extend to our esteemed **Prof. M. Brindha** for their visionary leadership and unwavering support throughout my journey. I am especially grateful to my mentors for their invaluable guidance and skilful management of my projects, which significantly contributed to my personal and professional growth.

I would also like to express my gratitude to my internship coordinator, **Er. SHAMSHER SINGH**, as well as our Head of Department, **DR. GIRISH CHANDRA**, for their support and encouragement.

Additionally, I am thankful to my seniors and peers for their pivotal role in helping me prepare for this enriching internship experience.

INTERNSHIP SUMMARY

Overview of the Internship

This report summarizes my experience during a **2-months internship** at **NIT Trichy**, undertaken as part of the requirements for my Bachelor of Technology program at **I.E.T. Lucknow**. The internship focused on implementing and optimizing deep learning models for **image recognition** and **natural language processing (NLP)** tasks. The internship provided hands-on experience with frameworks like TensorFlow and Keras, where I worked on designing and training neural network architectures such as **Convolutional Neural Networks (CNNs)** and **U-Net**. My role involved extensive data preprocessing, hyperparameter tuning, and improving model accuracy to meet research objectives.

During my internship, The paper delved into advanced techniques for improving model efficiency and accuracy in specific use cases, and I collaborated with researchers to analyze results and document findings. This experience not only enhanced my technical skills in deep learning but also strengthened my ability to conduct structured research and present technical insights effectively.

My Internship Experience at NIT Trichy.

During my internship at NIT Trichy, I worked on a research project focusing on **deep learning** for **image recognition** and **NLP**. I implemented advanced architectures like **CNNs** and **U-Net** using TensorFlow and Keras, optimizing performance through hyperparameter tuning and data augmentation. Collaborating with researchers, I addressed challenges in handling large datasets and contributed to a **research paper** documenting novel approaches to improving model accuracy and efficiency.

Key Projects and Contributions

- Developed a real-time object detection and segmentation system for autonomous vehicles using **CNN** and **Mask R-CNN**.
- Trained models on large urban driving datasets, optimizing for high accuracy and performance under varying conditions such as lighting and weather.
- Implemented precise segmentation of objects like vehicles and pedestrians to enhance safety and decision-making in autonomous driving.
- Addressed challenges in model optimization, computational efficiency, and real-time deployment, ensuring robust performance in diverse scenarios.

Technologies and Tools Used

- **Programming Languages:** Python, R
- **Frameworks and Libraries:** TensorFlow, Keras, Mask R-CNN
- **Tools:** Google Colab, Jupyter Notebooks
- **Data Management:** Pandas, NumPy
- **Visualization:** Matplotlib, Seaborn
- **Dataset Handling:** Large urban driving datasets with preprocessing and augmentation techniques

Challenges and Solutions

- **Challenge:** Handling large urban driving datasets, which required extensive preprocessing and augmentation.
  Solution: Implemented data normalization, augmentation techniques, and efficient batch processing to streamline training and improve model performance.
- **Challenge:** Achieving high accuracy under varying conditions such as lighting and weather.
  Solution: Trained the model on diverse datasets to improve generalization and applied regularization techniques to prevent overfitting.
- **Challenge:** Ensuring real-time detection and segmentation performance.
  Solution: Optimized the model using lightweight architectures and asynchronous processing to reduce inference time without compromising accuracy.
- **Challenge:** Managing computational limitations during training and testing.
  Solution: Utilized **Google Colab** with GPU support for training and implemented modular code to make resource usage more efficient.

Learning and Growth

The internship was a transformative experience that enabled me to:

- Gained a deep understanding of **deep learning architectures**, including CNN and Mask R-CNN, and their real-world applications in autonomous systems.
- Improved skills in **data preprocessing**, augmentation, and handling large datasets, ensuring model robustness and scalability.
- Enhanced problem-solving abilities by addressing challenges in accuracy, computational efficiency, and real-time performance.
- Strengthened expertise in tools like TensorFlow, Keras, and Google Colab, and learned how to optimize workflows for high-performance computing.
- Developed a systematic approach to research and implementation, balancing technical innovation with practical deployment needs.

I. INTRODUCTION

Background

As urban areas grow and traffic congestion becomes more prevalent, ensuring road safety and efficient transportation systems has become a critical challenge. Vehicle detection and segmentation are essential technologies in addressing these challenges, playing a pivotal role in applications such as autonomous driving, traffic monitoring, and intelligent transportation systems.Traditional vehicle detection methods often rely on basic image processing techniques, which struggle to cope with the dynamic and complex conditions of real-world traffic. These methods lack the robustness and precision required for high-stakes applications, especially under varying lighting conditions, dense traffic scenarios, and diverse vehicle types.

In recent years, advancements in deep learning, particularly Convolutional Neural Networks (CNNs), have revolutionized image analysis tasks. CNNs excel at extracting meaningful features from complex datasets, making them ideal for vehicle detection. Complementary architectures like U-Net enhance segmentation capabilities by accurately identifying object boundaries, which is crucial for pixel-level tasks in autonomous systems.

The integration of CNNs and U-Net, supported by robust frameworks like TensorFlow, has paved the way for real-time, high-accuracy vehicle detection and segmentation. This approach not only addresses current challenges in traffic management but also contributes significantly to the development of safe and efficient autonomous vehicles.

Objectives

The primary objectives of my internship were as follows:

1. Design and implement a vehicle detection and segmentation system using CNN and U-Net architectures.
2. Achieve high accuracy and efficiency in detecting and segmenting vehicles in diverse traffic conditions.
3. Utilize TensorFlow for effective training, testing, and deployment of the model.
4. Develop a versatile solution capable of handling various vehicle types and environmental scenarios.
5. Contribute to intelligent transportation systems for improved road safety and traffic management.

II. LITERATURE REVIEW

1. Advancements in Deep Learning for Vehicle Detection

Geng et al. explored the use of Fully Convolutional Networks (FCNs) and U-Net for semantic segmentation. FCNs provide a framework for pixel-wise predictions, enabling precise object boundary identification. U-Net, with its encoder-decoder structure and skip connections, enhances performance by integrating localization and contextual information. These architectures have become foundational in tasks requiring high accuracy, such as vehicle detection and segmentation in urban traffic conditions.

- **Key Takeaway:** The introduction of U-Net marked a significant improvement in localization and segmentation accuracy, making it suitable for complex traffic scenarios.

2. Integration of Object Detection and Segmentation

Zhu et al. proposed a hybrid model that combines YOLO (You Only Look Once) and U-Net for robust vehicle recognition. YOLO's real-time object detection capability complements U-Net's strength in pixel-level segmentation, resulting in an efficient and accurate system. This hybrid model has demonstrated resilience in challenging scenarios, such as occluded vehicles and varying environmental conditions.

- **Key Takeaway:** Combining detection and segmentation frameworks enhances overall system performance, making them reliable for real-world applications like autonomous driving.

3. Use of Mask R-CNN for Instance Segmentation

Mask R-CNN, introduced by Kaiming He et al., extends the Faster R-CNN framework by adding a branch for segmentation masks. This technique enables instance-level segmentation while maintaining high detection accuracy. It has achieved state-of-the-art performance in object instance segmentation, including vehicles, under diverse traffic and lighting conditions.

- **Key Takeaway:** Mask R-CNN is highly effective for both detection and segmentation, providing detailed insights at the pixel level for autonomous sys.

4. Role of Dataset Diversity

The availability of large, annotated datasets has been pivotal in training robust vehicle detection and segmentation models. Datasets such as the one used by Geng et al. include images captured under varying traffic conditions, lighting scenarios, and vehicle types. These datasets enable models to generalize better and perform well across different environments. Advances in hardware, such as GPUs and distributed computing, have further supported the training of these models on large-scale data.

- **Key Takeaway:** Dataset diversity and hardware advancements are critical to achieving high robustness and reliability in vehicle detection and segmentation models.

5. Applications Beyond Autonomous Vehicles

While vehicle detection and segmentation are essential for autonomous driving, their applications extend to other domains like traffic monitoring, surveillance, and road safety analysis. For instance, Bolya et al. developed YOLACT, a real-time segmentation method, which has been utilized for traffic analysis in smart city infrastructure.

- **Key Takeaway:** Vehicle detection and segmentation are versatile technologies with applications that extend far beyond autonomous vehicles, contributing to safer and smarter transportation systems.

The reviewed literature highlights the evolution of vehicle detection and segmentation technologies, emphasizing the role of advanced architectures like FCNs, U-Net, and Mask R-CNN. Hybrid models combining detection and segmentation capabilities further enhance performance, especially in challenging conditions.

III. METHODOLOGY

Materials and Methods

1. **Dataset:** Kaggle Vehicle-Image-Detection-Data with 17,762 labeled images, split into training (70%) and validation (30%).
2. **Frameworks:** TensorFlow for model implementation, Python for coding, and Matplotlib for data visualization.
3. **Hardware:** GPU-enabled systems for efficient training.
4. **Preprocessing:** Resize and normalize images; apply data augmentation **Hybrid.**
5. **Approach:** Combines CNN for feature extraction and U-Net for boundary refinement.
6. **Training and Testing:** Model is trained using the Adam optimizer, evaluated with accuracy and IoU metrics.

Project Design

1. **Objective:** Develop a vehicle detection and segmentation system using CNN and U-Net architectures.
2. **Architecture:**
    - **CNN (Convolutional Neural Network):** Used for feature extraction from inputs.
    - **U-Net:** Employed for precise pixel-level segmentation of vehicles, utilizing its encoder-decoder structure.
3. **Hybrid Model:**
    - Combines the strengths of CNN for detecting vehicles and U-Net for accurate segmentation.
4. **Input Data:**
    - Preprocessed images from the Kaggle dataset, including vehicle and non-vehicle categories, with pixel-level segmentation masks.
5. **Training:**
    - The model is trained using the Adam optimizer and binary cross-entropy loss, evaluated on a separate validation set.
6. **Output:**
    - The final output includes accurate vehicle detection maps with pixel-level boundaries, usable for intelligent transportation systems.

IV. Implementation

Steps Taken

1. **Understanding the Problem Statement:**
- Reviewed literature and existing methodologies on vehicle detection and segmentation.
- Identified gaps and areas for improvement in current practices.
2. **Setting Up Tools and Environment:**
- Installed and configured TensorFlow and supporting libraries.
- Set up a development environment for handling large datasets and intensive computations.
3. **Model Design:**
- Designed the CNN layers for hierarchical feature extraction.
- Integrated U-Net for pixel-level segmentation, ensuring precise vehicle boundary detection.
4. **Preprocessing Data:**
- Resized and normalized the dataset images.
- Performed data augmentation to account for varying conditions.
5. **Model Training and Testing:**
- Split data into training and validation sets to evaluate performance iteratively.
- Fine-tuned hyperparameters like learning rate and number of filters for optimal results.
6. **Result Analysis and Documentation:**
- Evaluated model performance using metrics like accuracy and loss.

Tools, Frameworks, and Technologies Used

- **Tools:**
  - TensorFlow for deep learning model development and training.
  - Python as the primary programming language for scripting and implementation.
  - Kaggle Datasets for accessing high-quality annotated vehicle and non-vehicle image data.
  - Image processing libraries like OpenCV for preprocessing

    - Matplotlib and Seaborn for visualizing performance metrics and model outputs.
- **Frameworks:**
    - **CNNs (Convolutional Neural Networks):** For hierarchical feature extraction to detect vehicles accurately.
    - **U-Net Architecture:** For precise pixel-level segmentation of vehicles in images.
- **Technologies:**
    - Advanced deep learning techniques for feature extraction and classification.
    - Image augmentation (e.g., rotation, scaling) to enhance model generalization.
    - GPU-based training for faster computations.

Optimizations and Accuracy Improvement

- **Hybrid Model:** Integrated CNNs and U-Net to balance detection accuracy and segmentation precision.
- **Hyperparameter Tuning:** Adjusted learning rates, batch sizes, and filter configurations to minimize validation loss.
- **Data Augmentation:** Performed image transformations like flipping, rotation, and contrast adjustment to create a more diverse training set.
- **Regularization:** Implemented dropout layers in CNNs to prevent overfitting.
- **Model Refinement:** Optimized the U-Net's encoder-decoder paths and used skip connections for better localization and segmentation.

V. Results and Discussion

Outcomes

- **Accuracy:** The hybrid CNN-U-Net model achieved a detection accuracy of 99.04%, demonstrating exceptional reliability in identifying and segmenting vehicles under varied traffic and environmental conditions.
- **Performance:** The model delivered consistent results in real-time applications, effectively distinguishing vehicles in scenarios ranging from heavy traffic to low visibility.
- **Learning Outcome:** This project provided practical experience in applying deep learning techniques, optimizing model architectures, and solving real-world challenges in intelligent transportation systems.

Challenges and Solutions

- **Challenge:** Handling occlusions, varying lighting, and heavy traffic densities.
- **Solution:** Data augmentation and extensive training on diverse datasets.
- **Challenge:** High training time and resource consumption.
- **Solution:** Optimized model architecture with skip connections and bottleneck layers to reduce redundancy.
- **Challenge:** Ensuring fast detection without compromising accuracy.
- **Solution:** Applied TensorFlow's distributed training and model trimming techniques.

VI. Conclusion

Summary of Findings

The hybrid CNN-U-Net model effectively detects and segments vehicles in diverse conditions. Data augmentation and regularization techniques contributed significantly to improved accuracy and model generalization. The use of TensorFlow enhanced the scalability and adaptability of the solution.

Limitations

Several limitations were encountered during the development and deployment of the system:

- **High Computational Requirements:**Training the hybrid CNN-U-Net model requires significant computational resources, including GPUs or TPUs. This makes it challenging to deploy on low-power devices such as mobile processors or embedded systems commonly used in autonomous vehicles.
- **Dependence on Dataset Quality:** The model's performance heavily relies on the quality and diversity of the training dataset. If the dataset does not adequately represent rare traffic conditions (e.g., extreme weather, unusual vehicle types), the model may struggle to generalize effectively.
- **Real-Time Challenges:** Although optimized, the hybrid model's computational complexity might still cause latency in scenarios demanding ultra-fast processing, such as high-speed autonomous driving in densely populated urban areas.
- **Limited Scope of Application:** The model focuses on camera-based detection, which can be affected by occlusions, poor lighting, or adverse weather. Without integration with other sensor technologies like LiDAR or radar, its effectiveness may be reduced in such conditions.

Future Recommendations

To enhance the system, the following steps are recommended:

- **Optimize for Edge Devices:** Simplify the model for real-time deployment on low-power devices, such as IoT cameras or mobile processors.
- **Expand Dataset:** Include more varied traffic and environmental conditions to improve model generalization.

- **Integration with Other Sensors:** Combine camera-based detection with LiDAR or radar for improved accuracy in adverse conditions.
- **Focus on Speed:** Develop lightweight models to balance accuracy and computational efficiency for real-time applications.

VII. REFERENCES

[1] Zhao X, Sun P, Xu Z, Min H (2020) “Fusion of 3D LIDAR and camera data for object detection in autonomous vehicle applications”. IEEE Sensors J 20(9):4901–4913.

[2] Geng K, Dong G, Yin G (2020) “Deep dual-modal traffic objects instance segmentation method using camera and LIDAR data for autonomous driving”. Remote Sens 12(20):3274

[3] Kaiming H , Georgia G, Piotr D, Girshick R (2018) “Mask R-CNN. arXiv:1703.06870v3” [cs.CV].

[4] Ronneberger et al. (2015) “Encoder-Decoder Methods for Semantic Segmentation: Efficiency and Robustness Aspects”.IEEE Sensors.

[5] Zhu et al. (2018) “Integrates detection and segmentation frameworks for robust performance Granada,” Spain, pp 126–133

[6] Krizhevsky et al. (2012) “Hierarchical feature learning from data for object detection”.IEEE Sensors.

[7] Sun J, Wan C, Cheng J, Yu FL, Liu J (2017) Retinal image quality classification using fine-tuned CNN. In: International workshop on fetal and infant image analysis (FIFI) / 4th international workshop on ophthalmic medical image analysis. Granada, Spain, pp 126–133

[8]Rezaei M, Ravanbakhsh E, Namjoo E, Haghighat M (2019) “Assessing the effect of image quality on SSD and faster R-CNNnetworks for face detection. In: 27th Iranian conference on electrical engineering”.Mashhad, Iran, pp 1589–1594.

[9]Bolya D, Zhou C, Xiao F, Lee YJ (2019) YOLACT: “real-time instance segmentation. In: IEEE/CVF international conference on computer vision”. Seoul, South Korea, pp 4493–4497

[10]“Kaggle Vehicles-Image-Detection-Data” (Dataset).

VIII. APPENDICES

Appendix A: Dataset Details

- **Source:** Kaggle and institutional datasets.
- **Contents:** Images for recognition and labeled text for NLP tasks.
- **Size:** Approximately 13000 data points.
- **Preprocessing:** Image augmentation, normalization, and tokenization for text.

Appendix B: Tools and Frameworks

- **Frameworks:** TensorFlow, Keras.
- **Languages:** Python for scripting and data preprocessing.
- **Platforms:** Google Colab, Jupyter Notebooks.
- **Libraries:** NumPy, Pandas, Matplotlib, Seaborn, Scikit-learn.