FINAL INTERNSHIP REPORT

Data Science Intern

Intern: Tarrush Saxena

Organization: NullClass

Duration: 21 Dec 2025 - 21 Jan 2026

NullClass Internship Report - Tarrush Saxena

1. Introduction

This report documents the activities, projects, and learning outcomes of my Data Science internship at NullClass. The internship, conducted from December 21, 2025, to January 21, 2026, focused on Computer Vision and Machine Learning. The primary goal was to apply theoretical concepts to real-world problems by building functional systems involving image processing, deep learning, and real-time detection.

2. Background

NullClass is an ed-tech platform providing practical project-based learning. This internship was designed to bridge the gap between academic learning and industry requirements. The domain of Data Science and Computer Vision was chosen due to its transformative potential in automation, surveillance, and human-computer interaction.

3. Learning Objectives

The key objectives of this internship were:
- To gain hands-on experience with Python, OpenCV, and Deep Learning frameworks (TensorFlow/Keras, PyTorch).
- To understand the end-to-end lifecycle of ML projects: data collection, preprocessing, model training, evaluation, and deployment.
- To implement real-time computer vision systems accessible via user-friendly GUIs.
- To improve problem-solving skills by addressing challenges in model accuracy and system performance.

4. Activities and Tasks (Project Summaries)

During the internship, I successfully developed and delivered six distinct projects:

4.1 Attendance System with Emotion Detection
Developed a smart attendance system using facial recognition (LBPH/CNN). It logs attendance with timestamps and analyzes student emotions (Happy, Neutral, etc.) to gauge classroom engagement.
Tech Stack: Python, OpenCV, TensorFlow, Tkinter.

4.2 Animal Detection System
Implemented a real-time object detection system using YOLOv8 to identify various animal species. This project has applications in wildlife monitoring and preventing animal-vehicle collisions.
Tech Stack: YOLOv8, OpenCV, Ultralytics.

4.3 Drowsiness Detection System
Built a driver safety system that monitors eye aspect ratio to detect drowsiness. It triggers alarms if the

driver's eyes remain closed for a threshold period. It also estimates the driver's age using a CNN.
Tech Stack: Keras, OpenCV, Dlib/FaceMesh.

4.4 Nationality Detection System
Created a multi-task learning system that predicts a person's nationality, age, emotion, and clothing color from a live video feed. It uses MobileNetV2 for feature extraction and custom classification heads.
Tech Stack: MobileNetV2, TensorFlow, K-Means.

4.5 Sign Language Detection System
Designed a real-time American Sign Language (ASL) alphabet recognizer. Using MediaPipe for hand tracking and a deep learning model for classification, it translates hand gestures into text.
Tech Stack: PyTorch, MediaPipe, OpenCV.

4.6 Car Colour Detection & Traffic Analysis
Developed a traffic monitoring tool that detects vehicles, classifies them by color (specifically highlighting blue cars), and counts vehicle density at traffic signals.
Tech Stack: YOLOv8, OpenCV, HSV Color Space.

5. Skills and Competencies

Throughout these projects, I acquired and refined the following skills:
- Programming: Advanced Python (NumPy, Pandas).
- Computer Vision: Image processing with OpenCV, Video stream handling, HSV color space manipulation.
- Deep Learning: Building and training CNNs with TensorFlow/Keras and PyTorch; Transfer Learning (MobileNetV2, YOLOv8).
- Tools: VS Code, Git/GitHub, Jupyter Notebooks.
- UI Development: Building desktop GUIs with Tkinter.
- Soft Skills: Project documentation, time management, and technical writing.

6. Feedback and Evidence

The projects were self-evaluated and tested against validation datasets.
- The Attendance System achieved high accuracy in controlled lighting.
- The YOLO-based detectors (Animal, Car) showed robust performance in real-time.
- Feedback from mentors (if applicable) and self-assessment highlighted the need for diverse training data to improve generalization across different environments.

7. Challenges and Solutions

Challenge 1: Variable Lighting Conditions
Issue: Face recognition and color detection were unreliable in poor lighting.

Solution: Applied histogram equalization and used HSV color space for more robust color filtering. Data augmentation was used during training.

Challenge 2: Real-time Performance Lag
Issue: Running deep learning models on CPU caused low FPS.
Solution: Optimized model sizes (using MobileNetV2/YOLOv8-Nano) and reduced frame processing resolution to maintain acceptable frame rates.

Challenge 3: False Positives in Drowiness Detection
Issue: Blinking was sometimes misclassified as drowsiness.
Solution: Adjusted the eye-closure time threshold (EAR) to distinguish between natural blinking and prolonged closure.

8. Outcomes and Impact

The internship resulted in a comprehensive portfolio of six functional computer vision applications. These projects demonstrate a strong command of modern AI tools and the ability to deliver end-to-end solutions. The codebases are clean, documented, and hosted on GitHub for public reference, contributing to the open-source community.

9. Conclusion

My internship at NullClass has been an invaluable learning experience. I have successfully transitioned from theoretical knowledge to building deployable AI systems. The challenges faced honed my debugging and optimization skills, preparing me for a professional role in Data Science and Computer Vision.

10. References

- GitHub Portfolio: https://github.com/TarrushSaxena/nullclass-internship-projects
- OpenCV Documentation: https://docs.opencv.org/
- TensorFlow/Keras Guide: https://www.tensorflow.org/
- Ultralytics YOLOv8: https://docs.ultralytics.com/