Data Science Internship Final Report
Prepared by: Mandar Kajbaje
Internship Duration: 24-09-2025 to 24-11-2025

1. Introduction

This internship focused on the design, development, and deployment of six distinct machine learning models centered around real-world computer vision applications. Each model addressed a unique problem domain — from attendance systems with emotion recognition to nationality detection, drowsiness monitoring, animal classification, sign language interpretation, and traffic analysis. The projects were implemented using Python, leveraging libraries such as OpenCV, TensorFlow/Keras, MediaPipe, scikit-learn, pandas, and tkinter for GUI components. All models were trained from scratch or fine-tuned using custom or publicly available datasets, ensuring hands-on experience in end-to-end ML pipeline development.

2. Background

Computer vision has emerged as one of the most transformative fields within artificial intelligence, enabling machines to interpret and understand visual data from the world. Applications span across industries — education, automotive safety, wildlife conservation, accessibility, and urban planning. This internship aimed to bridge theoretical knowledge with practical implementation by building functional, constraint-aware models that solve specific problems under realistic conditions (e.g., time-bound operation, multi-object detection, pop-up alerts, GUI integration).

The motivation behind these projects was to simulate industry-grade problem-solving scenarios where accuracy, efficiency, user interaction, and contextual awareness are critical.

3. Learning Objectives

The primary objectives of this internship were:

- To gain proficiency in designing and training custom machine learning models for computer vision tasks.
- To understand preprocessing, augmentation, model selection, evaluation, and deployment pipelines.
- To integrate models into interactive applications using GUI frameworks.
- To handle constraints such as time-based activation, multi-class detection, and conditional logic based on predictions.
- To develop skills in documentation, debugging, performance optimization, and result visualization.
- To learn how to structure and deliver an end-to-end AI project including reporting and evidence collection.

4. Activities and Tasks

The following six core projects were completed during the internship:

Task 1: Attendance System Model

- Trained a face recognition model using embeddings (FaceNet) combined with SVM/MLP classifiers.
- Integrated emotion detection using FER2013-trained CNN.
- Implemented time-gating logic (9:30 AM – 10:00 AM).
- Logged attendance + emotions + timestamps into CSV files.

Task 2: Animal Detection Model

- Built YOLOv5 or SSD-based object detector trained on COCO or custom animal dataset.
- Classified animals and flagged carnivores with red bounding boxes.
- Displayed pop-up count of carnivorous animals.
- Developed GUI supporting image/video upload with preview pane.

Task 3: Drowsiness Detection Model

- Used facial landmark detection (MediaPipe/Dlib) to extract eye/mouth features.
- Trained binary classifier (CNN/LSTM) for drowsiness prediction.
- Estimated age via regression/classification head.
- Marked sleeping persons in red, showed pop-up with counts and ages.
- GUI enabled image/video input with real-time preview.

Task 4: Nationality Detection Model

- Fine-tuned ResNet/ViT for ethnicity/nationality classification (custom dataset).
- Integrated emotion recognition (FER), age estimation, and color detection (OpenCV HSV masking).
- Conditional output logic: Indian → age + dress color + emotion; US → age + emotion; African → dress color + emotion; Others → nationality + emotion.
- GUI included image upload, preview, and dynamic output panel.

Task 5: Sign Language Detection

- Collected/gathered ASL dataset or created synthetic gestures.
- Trained CNN or MediaPipe Hands + LSTM model for gesture classification.
- Restricted operation between 6 PM – 10 PM using system clock checks.
- GUI supported both static image uploads and live webcam feed.

Task 6: Car Colour Detection Model

- Detected vehicles using pre-trained object detectors (YOLO/Cascade).
- Applied K-means clustering or histogram analysis for dominant car color extraction.
- Blue cars → red rectangle; others → blue rectangle.
- Counted pedestrians near traffic signals using pose/human detection.
- GUI displayed annotated images with counters.

All tasks were implemented in Jupyter Notebooks (.ipynb), later modularized for reusability and packaged together in a ZIP archive alongside trained weights, sample data, and dependency lists.

5. Skills and Competencies Acquired

Through this internship, I developed and enhanced the following technical and soft skills:

Technical Skills:

- Deep Learning Frameworks: TensorFlow, Keras, PyTorch
- Computer Vision Libraries: OpenCV, MediaPipe, Dlib
- Data Handling: Pandas, NumPy, Scikit-learn
- GUI Development: Tkinter, PIL, Matplotlib embedding
- Model Training & Evaluation: Transfer learning, hyperparameter tuning, confusion matrices, ROC curves
- Deployment Logic: Time-based triggers, conditional rendering, file logging

Soft Skills:

- Problem Decomposition: Breaking complex requirements into manageable modules
- Debugging & Optimization: Handling memory leaks, latency issues, false positives
- Documentation: Commenting code, maintaining clean notebooks, writing reports
- Project Management: Meeting deadlines, version control (Git), organizing deliverables

6. Feedback and Evidence

Each project was validated through:

- Quantitative Metrics: Accuracy, precision, recall, FPS (for video), inference time.
- Qualitative Testing: Manual verification of outputs against ground truth images/videos.
- User Interface Usability: Smooth navigation, responsive buttons, clear previews and outputs.
- Edge Case Handling: Low-light images, occluded faces, multiple overlapping objects, varying scales.

Evidence Provided:

- Screenshots of GUIs and detection results embedded in notebooks.
- Sample output CSV/logs generated by Attendance System.
- Pop-up message triggers captured during runtime.
- Confusion matrices and loss/accuracy graphs for trained models.
- Video demos (optional but recommended) showing real-time functionality.

All source code is fully executable and reproducible given dependencies and hardware (GPU recommended for faster training/inference).

7. Challenges and Solutions

Challenge	Solution
Real-time processing lag in video streams	Optimized frame skipping, used lightweight models (MobileNet, MediaPipe), multithreading
Poor lighting affecting emotion/color detection	Applied CLAHE, gamma correction, adaptive thresholding
Overlapping bounding boxes in crowd scenes	Implemented Non-Max Suppression (NMS) and confidence thresholds
Limited labeled data for nationality/emotion	Used data augmentation (rotation, flip, brightness), transfer learning, and synthetic generation
Time-based restriction logic not working across timezones	Used datetime.now().time() with fixed string comparison (“09:30” <= current <= “10:00”)
Memory overload during batch processing	Cleared GPU cache after each prediction, used generators for large datasets

These solutions ensured robustness and scalability across different environments and inputs.

8. Outcomes and Impact

The successful completion of all six projects demonstrates comprehensive mastery over applied computer vision and machine learning concepts. Key outcomes include:

- Six fully functional, requirement-compliant models ready for prototyping or further scaling.
- Modular, reusable code architecture facilitating easy adaptation to new use cases.
- Enhanced understanding of integrating business logic (conditional outputs, timing, alerts) into AI systems.
- Practical exposure to deploying models in constrained, real-world settings.
- Creation of an impressive portfolio showcasing diverse capabilities in AI/ML.

Potential impact areas:

- Educational institutions can adopt the Attendance System for automated roll calls.
- Wildlife researchers may utilize the Animal Detection tool for field monitoring.
- Automotive companies could integrate Drowsiness Detection for driver safety.
- Accessibility tools like Sign Language Recognition empower hearing-impaired communities.
- Urban planners might leverage Car Color & Pedestrian Counting for smarter traffic management.

9. Conclusion

This internship served as a pivotal milestone in my journey toward becoming a proficient machine learning engineer specializing in computer vision. By tackling six challenging, real-world problems — each with its own set of constraints and expectations — I gained invaluable hands-on experience that goes beyond textbook theory. From conceptualization to deployment, every step reinforced best practices in software engineering, model development, and user-centric design.

I am now confident in building intelligent vision systems capable of solving niche yet impactful problems. These projects not only reflect technical competence but also creativity, persistence, and attention to detail — qualities essential for success in the rapidly evolving field of artificial intelligence.

Looking ahead, I aim to extend these models with cloud deployment, mobile app integration, and edge device optimization to maximize their reach and utility.

Submitted By:

Mandar Kajbaje

Mandarkajbaje@gmail.com

Internship Duration: 24-09-2025 to 24-11-2025