# GSoC 2025 Proposal: Multi-Camera Calibration Test

## Personal Information

Name: Shivam Prasad

Email: shivamprasad10001@gmail.com

GitHub: https://github.com/shivamprasad1001

LinkedIn: https://www.linkedin.com/in/shivam-prasad1001/

Portfolio: https://shivamprasad.netlify.app

University: Dr. A. P. J. Abdul Kalam Technical University, Lucknow

Degree Program: B.Tech, Computer Science (Artificial Intelligence and Machine Learning)

## Synopsis

This project aims to enhance the camera calibration module in OpenCV by collecting high-quality calibration data from both public datasets and real-world setups. It includes writing test scripts for existing calibration functions, generating calibration visualizations, performing statistical analyses, and improving documentation. The goal is to establish a standardized calibration testbed to assess and demonstrate the performance of OpenCV's calibration pipeline.

## Benefits to the Community

This project will significantly improve the robustness, usability, and documentation of OpenCV's camera calibration tools. By establishing a standard testbed with diverse fiducial types and camera setups, the project ensures consistency and reliability across OpenCV calibration methods. Enhanced tests and statistical evaluations will help users choose optimal methods for their specific applications. Clear documentation and educational material will lower the entry barrier for new users and researchers.

## Deliverables

- Curated calibration data from public sources
- Calibration data collected using OpenCV fiducials and diverse cameras
- Ready-to-use scripts to generate and visualize calibration results
- New test functions for the calibration module

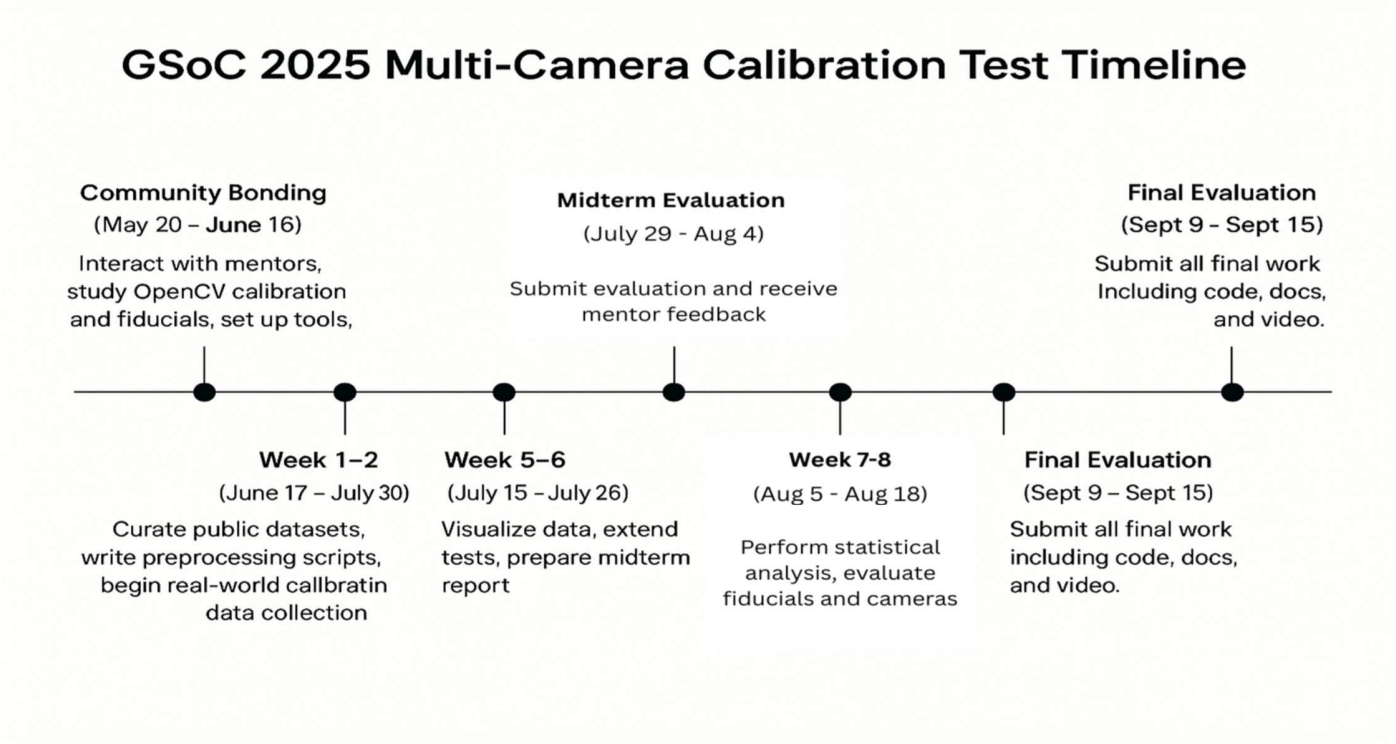
- Improved or new documentation for camera calibration
- Statistical report on accuracy and variance of calibration setups
- A demo YouTube video explaining the calibration pipeline and tests

### Technical Details

Languages: Python, C++
Tools/Libraries: OpenCV(cv2), NumPy, Matplotlib, Pandas

Dataset Sources: ETH3D, Open Images, TUM RGB-D, Custom recordings
Statistical Analysis: RMSE, reprojection error, accuracy/variance metrics
Visualization: Matplotlib, OpenCV drawing functions, potential use of Plotly for web-based plots
Video Editing: OpenCV or external tools for recording screen demos and processing

## Timeline



# Detailed explanation

### 🟢 Community Bonding (May 20 – June 16)

**Goals:**

- Engage with mentors and the OpenCV community.
- Understand the project scope and expectations.
- Study OpenCV's camera calibration modules, focusing on different fiducial markers (like ArUco and Charuco).
- Set up the development environment, including tools, libraries, and calibration data pipelines.
- Research and explore publicly available datasets for calibration tasks.

**Outcome:**
A strong foundational understanding of OpenCV's calibration ecosystem and a clear strategy for executing the project efficiently.

---

### 🗓️ Week 1–2 (June 17 – June 30)

**Goals:**

- Curate high-quality, publicly available calibration datasets.
- Write data preprocessing scripts (e.g., image resizing, fiducial detection, metadata extraction).
- Begin collecting calibration data using physical setups with ArUco and Charuco markers and different camera types (monocular, stereo, webcams, etc.).

**Outcome:**
A diverse and organized calibration dataset repository ready for model generation and testing.

---

### 🔧 Week 3–4 (July 1 – July 14)

**Goals:**

- Start generating camera calibration models using the collected data.
- Write scripts to automate camera matrix, distortion coefficients, and reprojection error calculations.

- Create initial test scripts for OpenCV's existing calibration functions (e.g., cv2.calibrateCamera, cv2.stereoCalibrate, etc.).

**Outcome:**
A working pipeline that takes in calibration data and produces calibration outputs along with basic test validations.

---

### 📊 Week 5–6 (July 15 – July 28)

**Goals:**

- Visualize calibration outputs using plots (e.g., error maps, overlayed points).
- Expand the test coverage to include edge cases and various camera-marker combinations.
- Prepare and document the Midterm Report, ensuring all progress is captured clearly.
- Validate the quality of data using performance metrics like reprojection error and detection consistency.

**Outcome:**
Robust test coverage and validated calibration workflows, all documented and ready for feedback.

---

### 📌 Midterm Evaluation (July 29 – Aug 4)

**Goals:**

- Submit a working prototype with test functions and visualization scripts.
- Deliver detailed documentation and code samples.
- Gather feedback from mentors to realign or refine the second-half goals.

**Outcome:**
Checkpoint validation of progress and direction, plus mentor guidance for further improvements.

---

### 📈 Week 7–8 (Aug 5 – Aug 18)

**Goals:**

- Write statistical analysis scripts to measure accuracy, variance, and performance of calibration functions.
- Compare different fiducial marker types (ArUco vs. Charuco) and camera types (monocular, stereo).

- Collect metrics like reprojection error, calibration stability, and detection reliability under different conditions.

**Outcome:**
Comprehensive performance benchmark of OpenCV calibration tools and marker configurations.

---

### 📘 Week 9–10 (Aug 19 – Sept 1)

**Goals:**

- Finalize all documentation, including improvements to OpenCV tutorials if needed.
- Create a demo video script and begin collecting visual assets (screen recordings, diagrams, voiceover plans).
- Make documentation beginner-friendly for wider OpenCV adoption.

**Outcome:**
Polished learning materials and assets ready for public consumption and presentation.

---

### 📽️ Week 11 (Sept 2 – Sept 8)

**Goals:**

- Record the demonstration video covering the complete workflow: setup, data collection, calibration, visualization, and statistical analysis.
- Edit and produce a high-quality YouTube video walkthrough.
- Finalize codebase: remove redundant files, organize folders, add comments, and perform final tests.

**Outcome:**
Complete public-facing deliverables and clean, ready-to-use source code.

---

### 🏁 Final Evaluation (Sept 9 – Sept 15)

**Goals:**

- Submit the final report, demo video, and code repository.
- Ensure that all documentation, test scripts, and outputs are published as per GSoC guidelines.
- Respond to any final mentor queries and wrap up remaining tasks.

**Outcome:**
A successful conclusion to the GSoC project with valuable contributions to OpenCV's calibration tools and community.

## About Me

I am a Computer Science (AI & ML) undergraduate with strong interests in computer vision and open-source development. I've been actively exploring OpenCV and have used it in projects involving facial recognition, object detection, and gesture control. I'm comfortable working with both Python and C++, and I enjoy building tools that combine mathematical depth with practical applications.

Contributing to OpenCV through GSoC is a dream opportunity for me to work with industry-grade vision tools, collaborate with experienced mentors, and contribute meaningfully to a globally recognized project. I'm particularly drawn to calibration and 3D reconstruction topics, and this project aligns perfectly with my passion and skills.

## Availability

I am fully committed to this project and can dedicate 25–30 hours per week during the coding period. I will treat this like a professional internship and prioritize it alongside any academic responsibilities.

## Commitment Statement

I understand the responsibilities involved in participating in GSoC and I am committed to delivering quality contributions to OpenCV. I will maintain regular communication with my mentors and provide weekly progress updates.