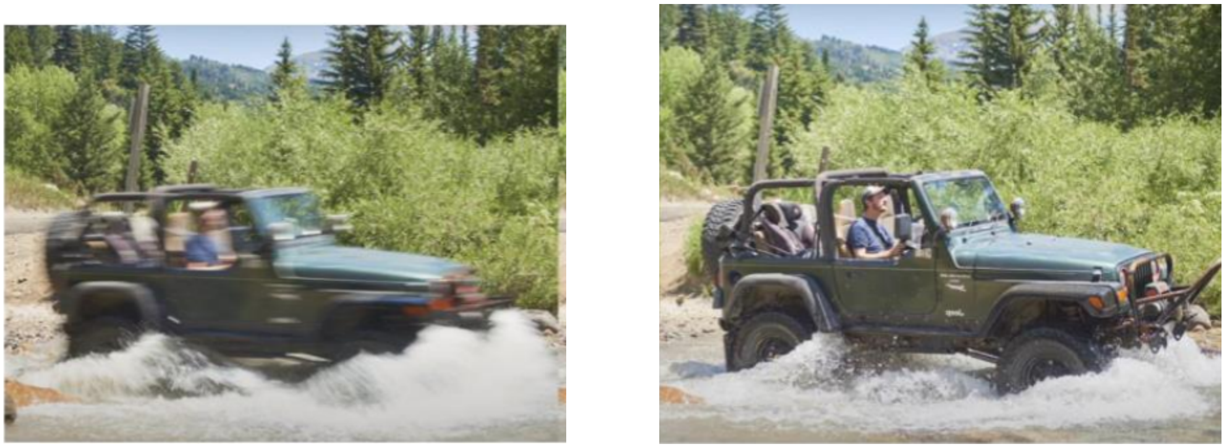
Comprehensive Project Report

Blur-Aware Camera Parameter Optimizer
An AI-Powered Mobile Photography Assistant

1. Introduction

This project introduces a smart camera assistant designed to **automatically predict optimal ISO and shutter speed** settings in real time, enhancing mobile photography in challenging conditions. Additionally, a **blur severity scoring model** quantifies image sharpness on a 0–100 scale. The system integrates handcrafted features, deep learning, and mobile deployment via Android and Flask APIs.



(a) Blur Photo

(b) Sharp Photo

Figure 1: Examples of Blur and Sharp Photos.

2. Objectives

- Automate camera parameter tuning to minimize motion blur.
- Enable real-time ISO and shutter speed predictions on mobile devices.
- Develop a perceptual blur scoring model.

- Deliver a full-stack prototype within hackathon constraints.

3. Dataset Preparation

A. ISO & Shutter Speed Prediction

- **Source:** ∼200 curated images with reliable EXIF metadata.
- **Filtering:** Removed corrupted entries (e.g., shutter speed = 0).
- **Augmentation:** Applied synthetic motion blur.
- **Features Extracted:** 7 handcrafted visual descriptors.

B. Blur Severity Prediction

- **Base Dataset:** 6,000 sharp images.
- **Synthetic Augmentation:** 18,000 blurred images (2:1 motion:Gaussian).
- **Label Normalization:** Scaled blur severity to 0–100.
- **Total:** 24,000 images with continuous blur labels.

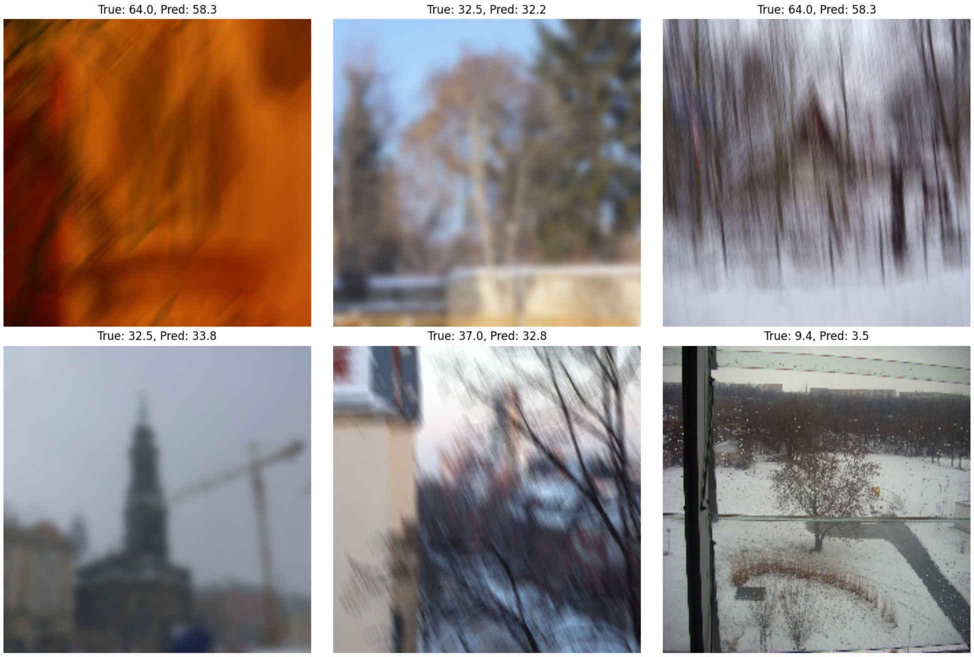


Figure 2: Dataset Generation And Labeling

4. Feature Engineering

Feature	Description
Laplacian Variance	Texture and focus detail indicator
Tenengrad Score	Gradient-based focus metric
Perceptual Blur Metric (PBM)	Measures edge width consistency
Edge Density	Ratio of edges via Canny detector
Brightness	Mean image luminance
Histogram Mean	Global luminance distribution
Histogram Variance	Luminance contrast

Table 1: Handcrafted features for ISO/Shutter Speed prediction

For blur severity scoring, we used **MobileNetV2** features directly on image tensors.

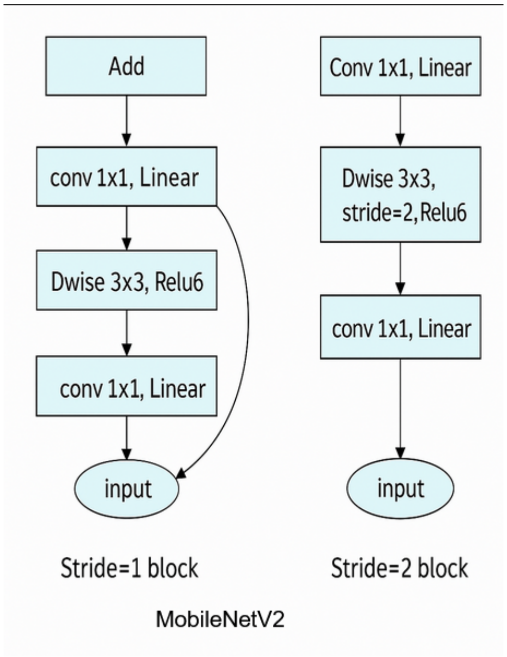


Figure 3: MobileNetV2 Architecture

5. Model Development

A. ISO Classification

- **Model:** MLP Classifier
- **Input:** 5 handcrafted features

- **Output:** Softmax over ISO levels (50–2500)
- **Loss:** Categorical Crossentropy
- **Training Insights:** High accuracy in ISO 50–800, resolved mid-range misclassifications

B. Shutter Speed Regression

- **Architecture:** MLP (64 → 32 → 1)
- **Target:** log(1/shutter speed)
- **Loss:** MSE, Optimizer: Adam
- **Regularization:** Dropout + BatchNorm
- **Training:** 150 epochs, early stopping

Metric	**Value**
MSE (log domain)	0.00091
MAE (shutter)	0.0082 s
R^2 Score	0.912
Median Abs. Error	0.0047 s

Table 2: Shutter Speed Regression Results

6. Blur Severity Regression Model

Architecture

- **Backbone:** MobileNetV2 (pretrained)
- **Head:** GlobalAvgPool → Dense(256, ReLU) → Dense(1)
- **Loss:** MSE **Metrics:** MAE, Pearson correlation

Training Parameters

The model predicts a **blur score from 0–100**, integrated into the app UI and future ISO/SS tuning pipelines.

Parameter	Value
Epochs	50
Batch Size	32

Table 3: Blur Severity Model Training Parameters

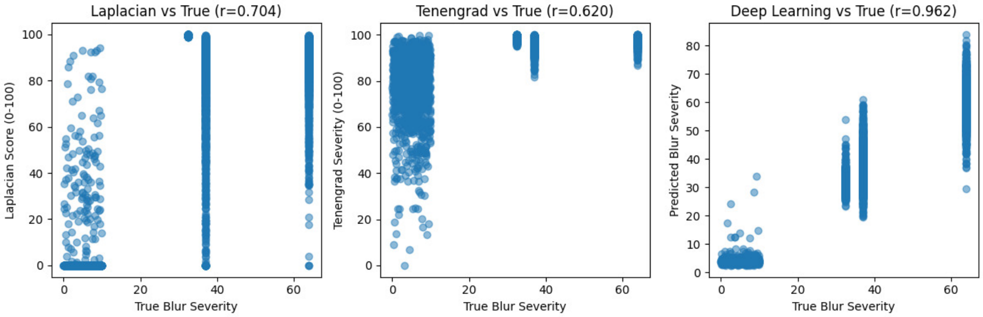


Figure 4: Model Evaluation

7. Deployment & API Architecture

A. Flask API (ISO + Shutter Speed)

- `/recommend_settings` – Returns ISO and shutter speed
- `/generate_heatmap` – Returns blur heatmap

B. Flask API (Blur Severity)

- `/predict_blur_score` – Returns blur score [0–100]
- Deployed locally for low latency

C. Android Integration

- Built with CameraX (Android Studio)
- Real-time blur detection using Laplacian Variance
- UI displays recommended settings + blur score
- Auto-adjusts ISO and shutter speed

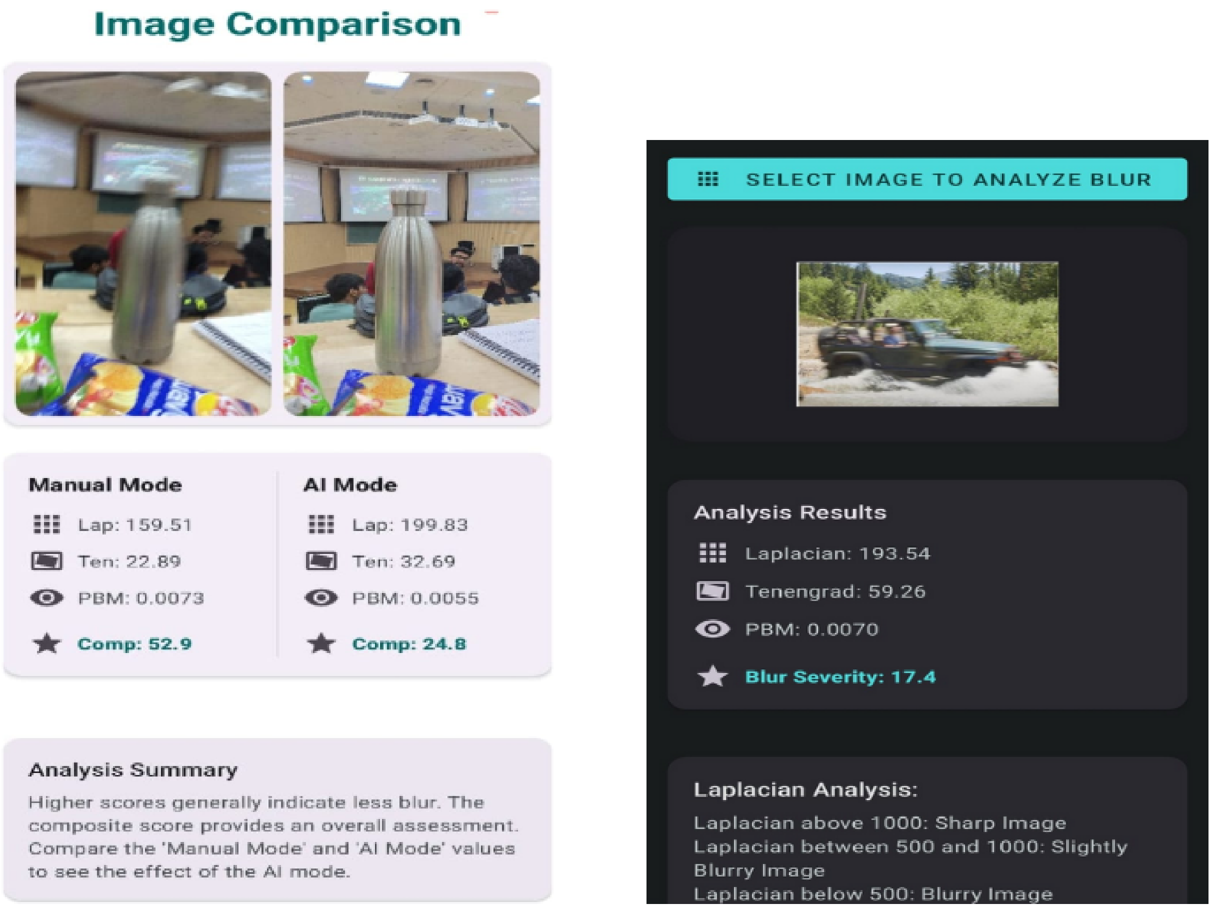


Image Comparison

Manual Mode

Lap: 159.51

Ten: 22.89

PBM: 0.0073

★ Comp: 52.9

AI Mode

Lap: 199.83

Ten: 32.69

PBM: 0.0055

★ Comp: 24.8

Analysis Summary

Higher scores generally indicate less blur. The composite score provides an overall assessment. Compare the 'Manual Mode' and 'AI Mode' values to see the effect of the AI mode.

SELECT IMAGE TO ANALYZE BLUR

Analysis Results

Laplacian: 193.54

Tenengrad: 59.26

PBM: 0.0070

★ Blur Severity: 17.4

Laplacian Analysis:

Laplacian above 1000: Sharp Image
Laplacian between 500 and 1000: Slightly Blurry Image
Laplacian below 500: Blurry Image

(a) Image Comparision

(b) Image Analysis

Figure 5: Working of our App

8.Blur Heatmap Visualization

- Converts image into a heatmap indicating sharp vs blurred regions
- Provides intuitive visual feedback

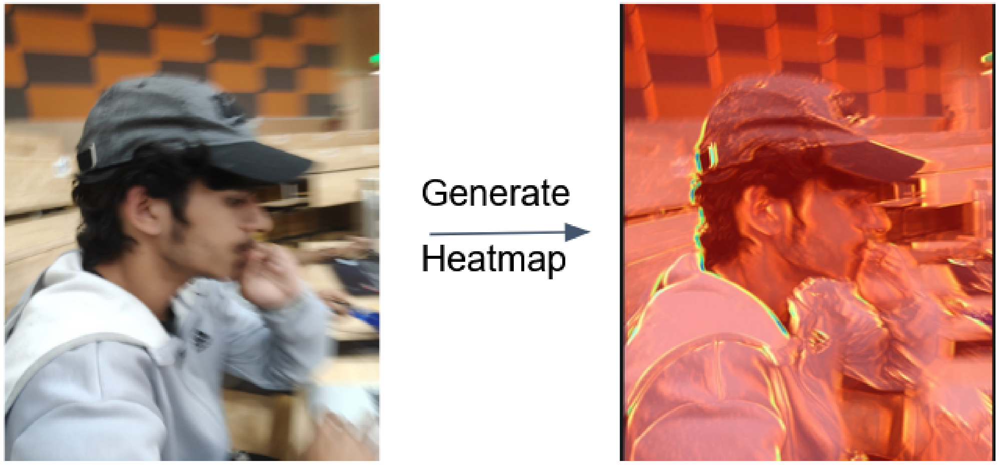


Figure 6: Generating Heatmap

9. Challenges & Resolutions

Challenge	Resolution
EXIF from HEIC files	Focused on JPG/PNG only
Model overfitting	Used dropout + early stopping
Deployment scaling bugs	Fixed scaler mismatch and exponent inversion
Real-time performance on Android	Used handcrafted features; offloaded CNN

Table 4: Key Challenges and Solutions

10. Alternate Approaches Explored

Approach	Status	Reason for Drop
CNN-based ISO/SS prediction	Dropped	Slow training, low interpretability
Continuous ISO regression	Dropped	Noisy, unstable results
Large-scale EXIF dataset (15k+)	Dropped	Metadata inconsistency
Hybrid CNN + handcrafted	Deferred	Infeasible within hackathon time

Table 5: Abandoned or Deferred Approaches

11. Final Outcomes

- Real-time mobile app with AI-based blur detection and tuning
- Flask API serving ISO, shutter speed, and blur heatmaps
- Blur severity model with perceptual scoring
- Lightweight, stable, and deployable solution

12. Future Roadmap

- Expand dataset to 10K+ images across blur/light variations
- Convert models to TensorFlow Lite for on-device inference
- Integrate aperture prediction for full exposure triangle
- Add user feedback loop for adaptive tuning
- Extend to video-based blur detection and auto-tuning