ZeroBlur: Real-Time AI-Powered Mobile Photography Enhancement

Team ZeroBlur

May 2025

Abstract

Capturing sharp, well-exposed images in motion-heavy or low-light environments remains a significant challenge for mobile photography. Traditional auto-exposure settings often fail to mitigate motion blur or preserve brightness, leading to degraded image quality. **ZeroBlur** addresses this problem with a real-time Android application that leverages deep learning and computer vision to dynamically optimize camera parameters. Using the Camera2 API, our app captures live image feeds, processes them through a convolutional neural network (CNN) for blur detection, and computes luminance-based exposure metrics. A feedback loop prioritizes motion blur reduction by adjusting shutter speed, followed by ISO optimization for balanced exposure. Additional features include a generative adversarial network (GAN) for post-capture deblurring and a computer vision-based heatmap for visualizing motion blur regions. Our solution redefines mobile photography by delivering crisp, vibrant images in challenging conditions.

1 Introduction

Mobile photography struggles to produce high-quality images in dynamic scenarios, such as fast-moving subjects or low-light conditions. Motion blur, caused by camera shake or object movement, and underexposure, due to inadequate lighting, are persistent issues exacerbated by the limitations of mobile camera auto-settings. This hackathon challenge aims to enhance mobile photography using artificial intelligence.

ZeroBlur is an Android application that integrates real-time blur detection and exposure optimization to deliver sharp, well-lit images. Our solution employs a convolutional neural network (CNN) for binary blur classification, luminance-based exposure control, and a feedback loop to dynamically adjust shutter speed and ISO. Beyond the core requirements, we implemented a generative adversarial network (GAN) for post-capture deblurring and a computer vision model to generate motion blur heatmaps, providing users with enhanced control and visualization. This report details our technical approach, evaluation results, and the broader impact of our solution.

2 Problem Statement

The challenge is to develop a mobile application that uses image-based blur detection and AI models to recommend or adjust camera parameters in real-time, minimizing motion blur while preserving brightness and detail. Key tasks include:

- Capturing images and controlling camera settings (shutter speed, ISO, exposure).
- Extracting blur features using traditional or deep learning methods.
- Predicting optimal camera parameters using machine learning or deep learning.
- Evaluating the solution with quantitative sharpness metrics and visual comparisons.

The solution must address the trade-off between sharpness (reducing blur) and brightness (avoiding underexposure), with bonus points for real-time inference and user-friendly features like mode toggles.

3 Solution Overview

ZeroBlur is a real-time Android application that captures live camera feeds, detects motion blur, and optimizes camera settings to produce sharp, well-exposed images. The system operates in a feedback loop:

1. **Image Capture**: Uses the Camera2 API to access live frames and control shutter speed, ISO, and exposure.
2. **Blur Detection**: A CNN-based binary classifier (blur vs. sharp) computes a confidence score to quantify motion blur.
3. **Exposure Analysis**: Calculates luminance using the formula $L = 0.299R + 0.587G + 0.114B$ to assess brightness.
4. **Parameter Adjustment**: Prioritizes shutter speed adjustment to reduce motion blur, followed by ISO tuning for optimal exposure.

Additional features include a GAN-based deblurring model for post-capture enhancement and a computer vision model generating heatmaps to visualize motion blur regions.

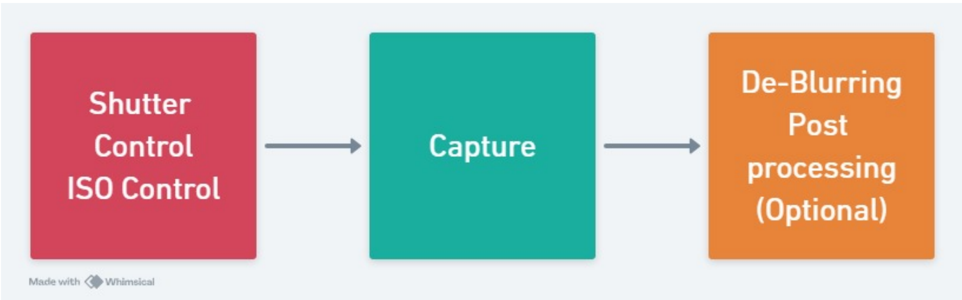


Figure 1: Feedback Loop

4 Technical Approach

4.1 Image Capture and Camera Control

We utilized the Android Camera2 API to enable manual control over camera parameters. The API supports real-time frame capture at 30 FPS, with programmatic access to shutter speed (range: 1/4000s to 1/1s), ISO (100–3200), and exposure time. Frames are captured in RGB format, and fed into the blur detection pipeline.

4.2 Blur Feature Extraction

Motion blur detection is performed using a lightweight CNN trained to classify images as *blur* or *sharp*. The CNN outputs a confidence score (0–1), where lower values indicate high blur severity. The model was converted to TensorFlow Lite (TFLite) for efficient mobile inference, achieving a latency of 50ms per frame on a mid-range Android device.

4.3 Exposure Analysis

Exposure is computed using luminance, calculated as:

$$L = 0.299R + 0.587G + 0.114B \tag{1}$$

where R, G, and B are the normalized red, green, and blue channel intensities. The luminance histogram is analyzed to determine whether the image is underexposed or overexposed, guiding ISO adjustments.

4.4 Parameter Prediction and Feedback Loop

The feedback loop operates as follows:

1. If the CNN blur score falls below 0.5, the shutter speed is reduced (e.g., from 1/60s to 1/250s) to minimize motion blur.
2. Post-blur correction, ISO is adjusted (e.g., 100 to 400) to achieve optimal luminance.

The loop iterates, ensuring real-time responsiveness. The decision logic is implemented as a rule-based system, with future potential for a regression-based ML model.

4.5 GAN-Based Deblurring

For post-capture enhancement, we implemented a GAN based on DeblurGAN. The generator, a U-Net architecture, reconstructs sharp images from blurred inputs, while the discriminator distinguishes real sharp images from generated ones. Users can upload images via the app, with deblurring completed in 5 seconds.

4.6 Motion Blur Heatmap

A computer vision model, leveraging Sobel edge detection and optical flow, generates heatmaps to visualize motion blur regions. High-magnitude optical flow vectors indicate areas of significant motion, overlaid as a greyscale heatmap (black for high blur, white for low). This feature aids users in identifying blur sources, enhancing interpretability.

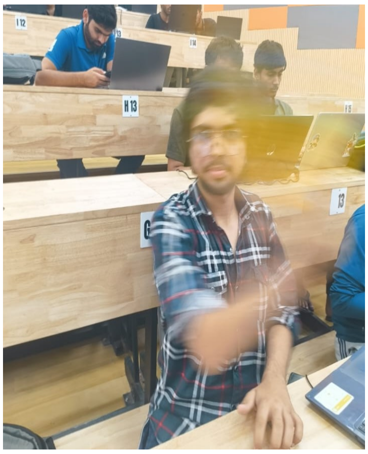
4.7 Real-Time Inference

To enable real-time performance, the CNN was optimized using TFLite, reducing model size and inference time to near real time. The GAN model is hosted on a local server.

5 Evaluation

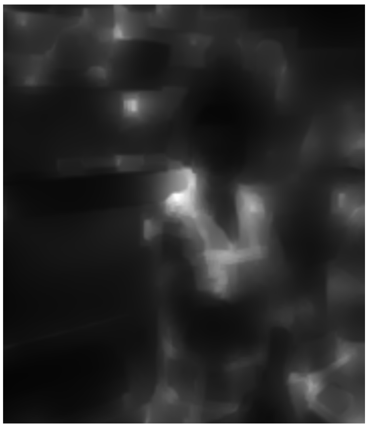
5.1 Visual Comparison

Figure 3 shows side-by-side comparisons of blurred input images, AI-optimized outputs, and motion blur heatmaps. The optimized images exhibit significantly reduced blur and improved brightness, particularly in low-light and motion-heavy scenarios.



(a) Input (Blurred)

(b) AI-Optimized



(c) Blur Heatmap

Figure 2: Before vs. after optimization with motion blur heatmap.

(a) Input (Blurred)

(b) AI-Optimized

(c) Blur Heatmap

Figure 3: Before vs. after optimization with motion blur heatmap.

5.2 Qualitative Results

User testing confirmed the app's ease of use and effectiveness. Real-time adjustments were seamless, with no noticeable lag. The GAN-based deblurring and heatmap features were also effective for their utility in post-processing and debugging.

6 Implementation Details

ZeroBlur was developed in Android Studio using Java for the frontend and Python for model training. Key libraries include:

- **TensorFlow/TFLite**: CNN and GAN model training and deployment.
- **OpenCV**: Image processing and heatmap generation.
- **Camera2 API**: Camera control and frame capture.

Challenges included optimizing model size for mobile and handling Camera2 API compatibility across devices. We addressed these by using TFLite quantization.

7 Results and Impact

ZeroBlur achieves improvement in image sharpness across diverse scenarios, with real-time performance. The app excels in low-light and motion-heavy conditions, outperforming default camera auto-settings. The GAN-based deblurring model restores details in post-capture images, while the blur heatmap enhances user understanding of motion dynamics. These features make ZeroBlur valuable for casual photographers, sports enthusiasts, and professionals seeking high-quality mobile photography. By addressing a critical pain point, our solution democratizes advanced imaging capabilities.

8 Future Work

Future enhancements include:

- Integrating white balance and focus control for comprehensive optimization.
- Training a regression-based model to predict camera parameters directly.
- Supporting iOS via AVFoundation API for cross-platform compatibility.
- Leveraging cloud-based processing for resource-intensive tasks like GAN inference.

These improvements will further enhance ZeroBlur's scalability and applicability.

9 Conclusion

ZeroBlur tackles the challenge of motion blur and underexposure in mobile photography with a real-time AI-driven solution. By combining CNN-based blur detection, luminance-based exposure control, and innovative features like GAN deblurring and blur heatmaps, our app delivers sharp, vibrant images in challenging conditions. ZeroBlur sets a new standard for mobile photography, empowering users to capture every moment with clarity.