

CV25 - VisionAR Project: Final Report

Title: Real-Time Artifact Removal for Passthrough AR

Team VisionAR:

- Azizbek - Project Lead
- Azim - Model & Evaluation
- Ikromjon - Data & Integration

1. Motivation & Architecture

Problem Statement:

Augmented Reality (AR) passthrough video often contains sensor noise, lighting artifacts, and temporal flickering, which degrade visual quality, cause user discomfort, and break immersion. The challenge is removing these artifacts in real time (<30ms per frame) while preserving important visual details.

Design Choices:

- Baseline denoising: Gaussian blur + bilateral filter (fast, interpretable, real-time capable, may over-smooth)
- CNN-based denoising: Pre-trained lightweight CNN (learns noise vs structure, preserves edges/textures, reduces flicker)
- Loss function (if applicable): MSE/L1 for pixel-level reconstruction

2. Baselines & Comparison

Method Comparison:

Method	Description	Advantages	Disadvantages
--------	-------------	------------	---------------

-----	-----	-----	-----
-------	-------	-------	-------

Gaussian + Bilateral	Classical filters	Fast, simple, interpretable	Removes fine details, blurry
----------------------	-------------------	-----------------------------	------------------------------

CNN (pre-trained)	Lightweight CNN	Preserves edges, reduces flicker, better quality	Slightly slower than baseline, requires GPU
-------------------	-----------------	--------------------------------------------------	---------------------------------------------

Comparison with other methods:

- ResNet variants are heavy for real-time
- Canny Edge detects edges but does not remove noise

3. Dataset

- Single 720p indoor public-domain video from Pexels
 - Natural sensor noise, lighting variation, motion
- Justification: realistic AR passthrough conditions, lightweight, suitable for iterative testing

4. Quantitative Evaluation

Metric	Baseline	CNN	Original
FPS	31.8	39.25	N/A
Latency (ms)	28.2	25.45	N/A
PSNR	N/A	28.91	N/A
SSIM	0.75	0.84	0.70*

*Approximate reference for original video

Analysis: CNN achieves higher FPS, lower latency, and better visual quality; baseline is fast but over-smooths textures

5. Qualitative Analysis

- Success Cases:
- Static indoor scenes
 - Well-lit areas
 - CNN reduces flicker and preserves edges

Failure Cases (example placeholders):

Frame Example	Issue	Explanation
Dark shadow	Slight over-smoothing	CNN may misinterpret low-contrast textures as noise
Fast motion	Minor temporal blur	Lightweight CNN cannot fully track motion
Low contrast object	Some detail lost	CNN prioritizes noise removal over faint textures

Takeaway: CNN works well most of the time; failures occur in low-light, low-contrast, or fast-motion regions

6. Architecture Diagram

(Insert a diagram showing pipeline: Input -> Baseline / CNN -> Output -> Evaluation)

7. Conclusion & Future Work

Conclusion:

- Baseline: fast but detail loss
- CNN: better visual quality + lower latency
- Real-time feasibility confirmed (<30ms latency)

Future Work:

- Temporal CNNs for motion consistency
- Larger AR video datasets
- Integration on AR hardware (glasses/headset)

8. References / Dataset License

- Pexels videos: <https://www.pexels.com/videos/>
- Open-source libraries: OpenCV, PyTorch, NumPy

License: Free to use for academic projects