# Forensic Pipeline Critical Review & Remediation Plan

Target System: Hybrid OSINT Deepfake Detector

Review Date: December 12, 2025

Reviewer: Senior Forensic Expert

Status: CRITICAL FAILURES DETECTED

## 1. Critical Code Failures (High Priority)

*These errors render the tool scientifically invalid against social media imagery and must be fixed immediately.*

### Failure 1: Destructive Resizing in FFT Analysis

File: forensics.py

Method: generate\_fft\_preprocessed (Line ~322)

Severity: Critical (Destroys Evidence)

* **The Pitfall:** The code uses cv2.resize with cv2.INTER\_LINEAR to standardize the image size to 512x512.
  + *Why this fails:* AI generation artifacts (checkerboard patterns from Transpose Convolutions) exist at the **pixel-to-pixel high frequency** level. Linear interpolation acts as a low-pass filter, mathematically smoothing out these pixels and erasing the specific fingerprints we are trying to detect.
* **The Fix:** **NEVER resize.** You must implement a **Center Crop** logic.
  + If image > 512x512: Crop the center 512x512 square.
  + If image < 512x512: Pad with cv2.BORDER\_REFLECT to reach 512x512.

### Failure 2: Unmasked Social Media Artifacts (The "JPEG Cross")

File: forensics.py

Method: generate\_fft and generate\_fft\_preprocessed

Severity: High (Blinds the VLM)

* **The Pitfall:** The FFT magnitude spectrum is generated without masking the central axes.
  + *Why this fails:* Images from social media (Twitter/Telegram) have sharp rectangular borders. In the frequency domain, these borders create a massive, high-energy **white cross (+)** running through the center of the spectrum. This signal is 100x stronger than any AI artifact, causing the VLM to either fixate on the cross or hallucinate anomalies.
* **The Fix:** Mechanically **mask (zero out)** the central DC component (the center dot) and the central vertical/horizontal axes (1-2 pixels wide).

## 2. Logic & Prompt Failures (Medium Priority)

*These errors increase the False Positive/Negative rates in specific OSINT contexts.*

### Failure 3: ELA Variance Misinterpretation

File: forensics.py

Context: compute\_ela\_variance and associated docstrings

Severity: Medium (False Positives)

* **The Pitfall:** The documentation claims Low variance (<2.0) indicates uniform compression (AI indicator).
  + *Why this fails:* Platforms like WhatsApp and Facebook aggressively re-compress *all* images, crushing the quantization tables. A real photo from WhatsApp will have a variance of ~0.5. Relying on this float value alone will flag every social media image as "AI".
* **The Fix:**
  + Update docstrings to clarify that **low variance is inconclusive** on social media.
  + The VLM must be instructed to look for **local inconsistencies** (e.g., a bright patch on a dark background), not global uniformity.

### Failure 4: Ambiguous "Grid" Instruction in Case A

File: prompt engineering strategy.md (and detector.py system prompts)

Context: CASE A: Uniforms / Parades / Formations

Severity: Medium (False Negatives)

* **The Pitfall:** The prompt instructs the VLM: *"Filter: IGNORE repetitive grid artifacts... These are caused by marching columns."*
  + *Why this fails:* This is too broad. It teaches the model to ignore **GAN artifacts** (which also look like grids).
  + *Differentiation:*
    - **Real Formation Grid:** Macro-scale, organic, imperfect alignment, low frequency.
    - **AI/GAN Grid:** Micro-scale, pixel-perfect, high frequency, often visible in the sky/noise.
* **The Fix:** Refine the prompt to explicitly distinguish between **Macro-Repetition** (Safe) and **Micro-Frequency/Pixel Grids** (Suspicious).

## 3. Recommended Remediation Code Snippets

### Fix for FFT (Center Crop + Masking)

# In forensics.py -> generate\_fft\_preprocessed  
  
# 1. CENTER CROP (Replaces Resize)  
h, w = gray.shape  
crop\_size = 512  
start\_y = max(0, (h - crop\_size) // 2)  
start\_x = max(0, (w - crop\_size) // 2)  
gray\_crop = gray[start\_y : start\_y+crop\_size, start\_x : start\_x+crop\_size]  
  
# Pad if too small  
h\_c, w\_c = gray\_crop.shape  
if h\_c < crop\_size or w\_c < crop\_size:  
 gray\_crop = cv2.copyMakeBorder(gray\_crop, 0, crop\_size-h\_c, 0, crop\_size-w\_c, cv2.BORDER\_REFLECT)  
  
gray = gray\_crop  
  
# ... (High pass filter logic) ...  
  
# 2. AXIS MASKING (After log transform)  
rows, cols = magnitude\_log.shape  
crow, ccol = rows // 2, cols // 2  
  
# Mask DC component (Center Dot)  
cv2.circle(magnitude\_log, (ccol, crow), 5, 0, -1)  
  
# Mask Axes (The Cross)  
magnitude\_log[crow-1:crow+1, :] = 0  
magnitude\_log[:, ccol-1:ccol+1] = 0

### Fix for Prompt (Case A)

# In prompt engineering strategy.md  
  
CASE A: Uniforms / Parades / Formations (Military Context)  
- Filter: IGNORE MACRO-scale repetitive patterns (e.g., lines of soldiers, rows of tanks).  
- Focus: Strictly FLAG MICRO-scale, perfect pixel-grid anomalies or symmetric 'star patterns' in the noise floor.