Characterizing and Taming Model Instability Across Edge Devices

Jenya Pergament

Joint work with: Eyal Cidon*, Zain Asgar, Asaf Cidon, Sachin Katti
To appear in MLSys 21'
Burst Mode

What does MobileNetV2 predict?

Tub
Hotdog
Pineapple
Butternut squash
Camera systems are designed for humans

Image credit: https://www.apple.com/iphone-12-pro/
Digital Camera Pipeline

ISP (Image Signal Processor)

- Gamma Correction
- White Balancing
- Demosaicing
- Compression
- Color Transformations
- Noise Reduction
Camera systems are highly complex and highly heterogeneous.
# Cameras

- **Pro camera system**
  - Ultra Wide, Wide, Telephoto
  - Image from: https://www.apple.com/iphone/

- **Advanced dual-camera system**
  - Ultra Wide, Wide

- **Single-camera system**
  - Wide

---

## Lenses

---

## Sensors

- **ISOCELL HM3**
  - 108MP Ultra-high resolution

- **ISOCELL GNI**
  - 50Mp with Dual Pixel autofocus

- **ISOCELL GM5**
  - 48MP with 0.7μm pixels

---

## ISPs

Beyond Human Vision
Photography Noise Pattern is Non-Standard

Water Bottle

Bubble
(NOT) Adversarial Attack

\[ x + 0.007 \times \text{sign}(\nabla_x J(\theta, x, y)) \]

“panda”
57.7% confidence

“nematode”
8.2% confidence

Image credit: Explaining and Harnessing Adversarial Examples, Goodfellow et al, ICLR 2015

Image credit: Robust Physical-World Attacks on Deep Lear

Image credit: One Pixel Attack for Fooling Deep Neural Networks, Su et al,
RAW vs. Processed formats
Dataset Problem

- Publicly available dataset
- RAW images
- Metadata
- Classification labels
- Images from multiple smartphones
Dataset Problem

- Publicly available dataset ✓
- RAW images ✓
- Metadata ✓
- Classification labels ✗
- Images from multiple smartphones ✗
Dataset Problem

- Publicly available dataset
- RAW images
- Metadata
- Classification labels
- Images from multiple smartphones
Our Dataset Collection

1. HTTP request
2. Photo presented on the monitor
3. Image taken by phone
Devices used for Data Collection

Motorola Moto G5
HTC Desire 10 Lifestyle
LG K10 LTE
Samsung Galaxy S10
iPhone XR

JPEG
HEIF
Android
iOS
Sample of the Collected Dataset

Images (1537):
- Taken by friends and family
- Amazon
- Flickr

We took photos (68125):
- 5 phones
- Different angles relative to the screen

---

**Backpack**

**Purse**

**Wine bottle**

**Water bottle**

**Beer bottle**
Unstable Image

Environment A

Environment B

Fixed Model

Environment A
CORRECT
backpack

Environment B
INCORRECT
beer bottle

Fixed Conditions: lighting, angle, distance, etc.
Instability Metric

# Correct prediction under Env A

# Correct prediction under Env B

Instability = % out of
End-to-End Experiment

Image

HTC (correct): Backpack

MobileNetV2

Samsung (incorrect): Quilt
Instability > Accuracy

Accuracy

Instability
Prediction Score

Stable Images

Unstable Images
ISP Experiment

RAW Image
ISP Experiment Results

<table>
<thead>
<tr>
<th>Metric</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adobe Photoshop Accuracy</td>
<td>49.96%</td>
</tr>
<tr>
<td>ImageMagick Accuracy</td>
<td>54.75%</td>
</tr>
<tr>
<td>Instability</td>
<td>14.11%</td>
</tr>
</tbody>
</table>

Adobe Photoshop

ImageMagick

Vestment

Backpack

Wine Bottle

Whiskey Jag
Compression Experiment

RAW Image

Compression Format

JPEG
HEIC
webp
PNG

Compression Quality

JPEG 100
JPEG 85
JPEG 50
## Compression Experiment Results

### Compression Format

<table>
<thead>
<tr>
<th>Metric</th>
<th>JPEG</th>
<th>PNG</th>
<th>WebP</th>
<th>HEIF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avg. Size [MB]</td>
<td>1.54</td>
<td>6.49</td>
<td>0.29</td>
<td>0.57</td>
</tr>
<tr>
<td>Accuracy</td>
<td>53.9%</td>
<td>53.9%</td>
<td>55.2%</td>
<td>54.4%</td>
</tr>
<tr>
<td>Instability</td>
<td></td>
<td></td>
<td></td>
<td><strong>9.66%</strong></td>
</tr>
</tbody>
</table>

### Compression Quality

<table>
<thead>
<tr>
<th>Metric</th>
<th>JPEG 100</th>
<th>JPEG 85</th>
<th>JPEG 50</th>
</tr>
</thead>
<tbody>
<tr>
<td>Avg. Size [MB]</td>
<td>3.05</td>
<td>0.65</td>
<td>0.25</td>
</tr>
<tr>
<td>Accuracy</td>
<td>54.0%</td>
<td>54.3%</td>
<td>54.5%</td>
</tr>
<tr>
<td>Instability</td>
<td></td>
<td></td>
<td><strong>7.6%</strong></td>
</tr>
</tbody>
</table>
Compression Examples

(a) HEIF: backpack  
(b) JPEG: bonnet  
(c) PNG: bonnet

(d) HEIF: wine bottle  
(e) JPEG: wine bottle  
(f) PNG: beer bottle

(g) HEIF: safety pin  
(h) JPEG: purse  
(i) PNG: stethoscope

(j) JPEG-q100: beer bottle  
(k) JPEG-q85: lighter  
(l) JPEG-q50: lighter
So Far

- End-to-End experiment: average Instability 14% at worst 20%
- ISP: 14% Instability
- Compression: 7-10% Instability
- JPEG decoding adds minor instability > 1%
Stability Training

Our solution is inspired by the paper:

Improving the Robustness of Deep Neural Networks via Stability Training

Loss Function

\[ L(x, x', \theta) = L_0(x, \theta) + \alpha L_{stab}(x, x') \]

**Probabilities**

Classification Cross Entropy:

\[ L_0(x, \theta) = -\sum_j \hat{y}_j \log \left( P(y_j | x, \theta) \right) \]

Matching Probability Distributions (Relative Entropy):

\[ L_{stab}(x, x', \theta) = -\sum_j P(y_j | x, \theta) \log \left( \frac{P(y_j | x', \theta)}{P(y_j | x, \theta)} \right) \]

**Embeddings**

Classification Cross Entropy:

\[ L_0(x, \theta) = -\sum_j \hat{y}_j \log \left( P(y_j | x, \theta) \right) \]

Matching Embeddings:

\[ L_{stab}(x, x', \theta) = \| f(x, \theta) - f(x', \theta) \|_2 \]
Extended Stability Training

Distortion noise

- Simulate image processing noise, randomly distort (epsilon):
  - Hue
  - Contrast
  - Brightness
  - JPEG compression qualities
  - ...

![Diagram of DNN with distortion noise](image)
Extended Stability Training
Two images noise

- Two images noise. Use:
  - Image I from phone A
  - Image I’ from phone B
  - I and I’ are the same object
Extended Stability Training

Subsample noise

- Two images noise. Use:
  - Image $I$ from phone A
  - Image $I'$ from phone B
  - $\text{label}(I) == \text{label}(I')$
## Extended Stability Training Results

<table>
<thead>
<tr>
<th>Embedding Matching</th>
<th>Relative Entropy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Noise</td>
<td>Instability</td>
</tr>
<tr>
<td>Two images</td>
<td>3.91%</td>
</tr>
<tr>
<td>Subsample</td>
<td>4.22%</td>
</tr>
<tr>
<td>Distortion</td>
<td>5.12%</td>
</tr>
<tr>
<td>Gaussian</td>
<td>5.12%</td>
</tr>
<tr>
<td>No Noise</td>
<td>7.22%</td>
</tr>
</tbody>
</table>
Current and Future Work

- Should we use the ISP for ML on the edge?
- Can we create a pipeline (similar to ISP) optimized for ML?
- Can we replace the ISP with an ML model?
- Can we create an intermediate image representation?
- Collecting a dataset for edge devices
THANK YOU!