Multi-Patch Aggregation Models For Resampling Detection

Computational Imaging Lab
Indian Institute of Technology Madras
Mohit Lamba and Kaushik Mitra
Malicious Intents !!!
Related Works

Related Works


Suitable for UNCOMPRESSED Images
Related Works

- Bayar, B., & Stamm, M. C., TIFS(2018)
- Sahu, S., & Okade, M., WIFS(2018)
Related Works

- Bayar, B., & Stamm, M. C., TIFS(2018)
- Sahu, S., & Okade, M., WIFS(2018)

JPEG + Resampling + JPEG
Related Works


JPEG + Resampling + JPEG
(Image Acquisition) (Forgery) (Re-Save)
<table>
<thead>
<tr>
<th>Work</th>
<th>Image Size</th>
<th>Patch Size</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sahu et al., WIFS(2018)</td>
<td>1024 × 1024</td>
<td>512 × 512</td>
</tr>
<tr>
<td>MISLnet (Bayar et al., TIFS 2018)</td>
<td>256 × 256</td>
<td>256 × 256</td>
</tr>
<tr>
<td>Li et al., TCSVT(2018)</td>
<td>512 × 512</td>
<td>512 × 512</td>
</tr>
<tr>
<td>Verma et al., Elsevier(2018)</td>
<td>512 × 384</td>
<td>128 × 128</td>
</tr>
<tr>
<td>Kirchner et al., WIFS(2009)</td>
<td>1024 × 1024</td>
<td>512 × 512</td>
</tr>
<tr>
<td>Bianchi et al., WIFS(2012)</td>
<td>1024 × 1024</td>
<td>512 × 512</td>
</tr>
<tr>
<td>Quan et al., TIFS(2018)</td>
<td>&lt;1024 × 1024</td>
<td>233 × 233</td>
</tr>
</tbody>
</table>
Images occur in all kinds of sizes and dimensions.
Images occur in all kinds of sizes and dimensions..
## Experiments on MISLnet (Bayar et al., TIFS 2018)

<table>
<thead>
<tr>
<th>Patch size / Img Resolution</th>
<th>Resampling Factors</th>
<th>Avg Acc %</th>
</tr>
</thead>
<tbody>
<tr>
<td>256 × 256/1024 × 1024</td>
<td>0.6</td>
<td>99.3</td>
</tr>
<tr>
<td></td>
<td>0.8</td>
<td>98.8</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>99.1</td>
</tr>
<tr>
<td></td>
<td>1.2</td>
<td>99.4</td>
</tr>
<tr>
<td></td>
<td>1.4</td>
<td>99.0</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>99.0</strong></td>
</tr>
<tr>
<td>256 × 256/Variable</td>
<td>0.6</td>
<td>89.0</td>
</tr>
<tr>
<td></td>
<td>0.8</td>
<td>73.5</td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>97.5</td>
</tr>
<tr>
<td></td>
<td>1.2</td>
<td>96.6</td>
</tr>
<tr>
<td></td>
<td>1.4</td>
<td>93.4</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>89.90</strong></td>
</tr>
</tbody>
</table>
Resampling Detection for images of varying sizes & dimensions

- With No Priors (Blind Technique)
- With Some Prior Knowledge (Non Blind Technique)
Resampling Detection for images of varying sizes & dimensions

- With No Priors (Blind Technique)
  - Absolutely NO idea what the base dimension is...
- With Some Prior Knowledge (Non Blind Technique)
  - A rough guess about the base resolution...
A comprehensive guide to convolutional neural networks - the eli5 way

Iterative Pooling Strategy - A Blind Technique w/o priors

INPUT → CONVOLUTION + RELU → POOLING → CONVOLUTION + RELU → POOLING → FLATTEN → FULLY CONNECTED → SOFTMAX

FEATURE LEARNING → CLASSIFICATION

2^{i}H x 2^{i}W x C (Input Tensor) → After 1^{st} step → 3 x 3 conv stride = 2 padding = 1 → At log_{2}(2^{i}) step → H x W x C (Output Tensor)
Iterative Pooling Strategy - A Blind Technique w/o priors

Output dimension (fixed): $8 \times 8 \times C$
Input dimension: $64 \times 64 \times C$

STEP 1 ----> $32 \times 32 \times C$
STEP 2 ----> $16 \times 16 \times C$
STEP 3 ----> $8 \times 8 \times C$
Iterative Pooling Strategy - A Blind Technique w/o priors

Output dimension (fixed): \(8 \times 8 \times C\)

Input dimension: \(256 \times 256 \times C\)

STEP 1 -----> \(128 \times 128 \times C\), STEP 2 -----> \(64 \times 64 \times C\),
STEP 3 -----> \(32 \times 32 \times C\), STEP 4 -----> \(16 \times 16 \times C\),
STEP 5 -----> \(8 \times 8 \times C\)
Branched Network - A Non Blind Technique with priors
Branched Network - A Non Blind Technique with priors

Category I
64x64 patch

Low-Resolution
< 512 x 512

Category II
128x128 patch

VGA-Resolution
~ 1024 x 1024

Category III
256x256 patch

High Definition
eg: 2000 x 4000
Branched Network - A Non Blind Technique with priors

**Category I**
64x64 patch

**Category II**
128x128 patch

**Category III**
256x256 patch

**Low-Resolution**
< 512 x 512

**VGA-Resolution**
~ 1024 x 1024

**High Definition**
eg: 2000 x 4000
Branched Network - A Non Blind Technique with priors

**Category I**
64x64 patch

**Category II**
128x128 patch

**Category III**
256x256 patch

**Low-Resolution**
< 512 x 512

**VGA-Resolution**
~ 1024 x 1024

**High Definition**
eg: 2000 x 4000
<table>
<thead>
<tr>
<th>Method</th>
<th>Resampling Factors</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.6</td>
</tr>
<tr>
<td>MISLnet (Bayar et al. TIFS 2018)</td>
<td>84.7</td>
</tr>
<tr>
<td>Quan et al. TIFS(2018)</td>
<td>80.4</td>
</tr>
<tr>
<td>Chen et al. JRTIP(2019)</td>
<td>73.4</td>
</tr>
<tr>
<td>Iterative Pooling Strategy (ours)</td>
<td>98.6</td>
</tr>
<tr>
<td>Branched Network (ours)</td>
<td>98.4</td>
</tr>
</tbody>
</table>

Image Resolutions: 512×512, 1024×1024, 3008×2000, 4288×2848 and 4928×3264
<table>
<thead>
<tr>
<th>Method</th>
<th>Resampling Factors</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.6</td>
</tr>
<tr>
<td>MISLnet (Bayar et al. TIFS 2018)</td>
<td>84.7</td>
</tr>
<tr>
<td>Quan et al. TIFS(2018)</td>
<td>80.4</td>
</tr>
<tr>
<td>Chen et al. JRTIP(2019)</td>
<td>73.4</td>
</tr>
<tr>
<td>Iterative Pooling Strategy (ours)</td>
<td>98.6</td>
</tr>
<tr>
<td>Branched Network (ours)</td>
<td>98.4</td>
</tr>
</tbody>
</table>

JPEG+Resampling+JPEG
<table>
<thead>
<tr>
<th>Method</th>
<th>Resampling Factors</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.6</td>
</tr>
<tr>
<td>MISLnet (Bayar et al. TIFS 2018)</td>
<td>84.7</td>
</tr>
<tr>
<td>Quan et al. TIFS(2018)</td>
<td>80.4</td>
</tr>
<tr>
<td>Chen et al. JRTIP(2019)</td>
<td>73.4</td>
</tr>
<tr>
<td>Iterative Pooling Strategy (ours)</td>
<td>98.6</td>
</tr>
<tr>
<td>Branched Network (ours)</td>
<td>98.4</td>
</tr>
</tbody>
</table>

**MISLnet, Quan et al. & Chen et al.**  
Give ~99% for fixed resolution images
<table>
<thead>
<tr>
<th>Method</th>
<th>Resampling Factors</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.6</td>
</tr>
<tr>
<td>MISLnet (Bayar et al. TIFS 2018)</td>
<td>84.7</td>
</tr>
<tr>
<td>Quan et al. TIFS(2018)</td>
<td>80.4</td>
</tr>
<tr>
<td>Chen et al. JRTIP(2019)</td>
<td>73.4</td>
</tr>
<tr>
<td>Iterative Pooling Strategy (ours)</td>
<td>98.6</td>
</tr>
<tr>
<td>Branched Network (ours)</td>
<td>98.4</td>
</tr>
</tbody>
</table>

**MISLnet, Quan et al. & Chen et al.**

Give ~99% for fixed resolution images
But lose ~10% accuracy with images of variable resolution
Conclusion

- Flexible patch size at inference.
- Scales easily with variable resolution.
- Iterative Pooling Strategy w/o priors.
- Branched Network with priors.

Code: github.com/MohitLamba94/Iterative-Pooling