An Inception-Based Data-Driven Ensemble Approach to Camera Model Identification

Anselmo Ferreira, Han Chen, Bin Li and Jiwu Huang

December 11, 2018
1 Motivation

2 Proposed Method

3 Experimental Setup

4 Results

5 Conclusion
Motivation

Proposed Method

Experimental Setup

Results

Conclusion
Sensitive Image Source Linking

Figure 1: Which camera took this photo?
Related Work

- Several approaches have been proposed to tackle such problem with feature-based [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28] and Data-Driven solutions [1, 25, 26, 27, 28, 29]

- Branches of research are focused in two tasks:
  - Exact camera identification
  - Model identification
Motivation

Related Work

- Several approaches have been proposed to tackle such problem with feature-based [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28] and Data-Driven solutions [1, 25, 26, 27, 28, 29]
- Branches of research are focused in two tasks:
  - Exact camera identification
  - **Model identification**
Related Work-Limitations

- DL approaches solutions are focused on finding better pre-processing modules and using CNNs that are not too wide nor too deep.
- Public available datasets built so far such as DRESDEN [30] and VISION [31] consider only a very small set of devices with the same model and brand.
Related Work—Limitations

- DL approaches solutions are focused on finding better pre-processing modules and using CNNs that are not too wide nor too deep.

- Public available datasets built so far such as DRESDEN [30] and VISION [31] consider only a very small set of devices with the same model and brand.
1 Motivation

2 Proposed Method

3 Experimental Setup

4 Results

5 Conclusion
An Inception-Based Data-Driven Ensemble Approach to Camera Model Identification

Proposed Method

Our Solution

Figure 2: Proposed method for camera model identification. It is composed of a simple architecture CNN applied on CNNs pre-processed data on images regions of interest.
STEP #1 REGIONS OF INTEREST EXTRACTION
STEP #1 REGIONS OF INTEREST EXTRACTION

Image blocks are extracted using a metric [1] considering the mean and standard deviation of pixel values in each $k$ channel:

$$u_{score_k} = -4 \times u_k^2 + 4 \times u_k$$  \hspace{1cm} (1)$$

$$\sigma_{score_k} = 1 - e^{-2 \times \log_e(10) \times \sigma_k}$$  \hspace{1cm} (2)$$

$$patch_{score} = 0.7 \times \overline{u_{score}} + 0.3 \times \overline{\sigma_{score}},$$  \hspace{1cm} (3)$$
STEP #2: TWO-DIMENSIONAL SIGNALS CNNs
PRE-PROCESSING
INCEPTION-RESNET-v2 [32]
An Inception-Based Data-Driven Ensemble Approach to Camera Model Identification

Proposed Method

Figure 3: Inception-ResNet-v2 architecture using residual and parallel feature maps merging.
XCEPTION-NETWORK [33]
An Inception-Based Data-Driven Ensemble Approach to Camera Model Identification

Proposed Method

Middle Flow

Conv 32, 32*3, stride=2*2
Relu
Relu
Conv 64, 3*3, stride=2*2
SeparableConv 128, 3*3
Relu
SeparableConv 128, 3*3
Relu
MaxPooling 3*3, stride=2*2
SeparableConv 256, 3*3
Relu
SeparableConv 256, 3*3
Relu
MaxPooling 3*3, stride=2*2
SeparableConv 728, 3*3
Relu
SeparableConv 728, 3*3
Relu
SeparableConv 728, 3*3
Repeated 8 times

Entry Flow

Conv 1*1 stride=2*2
Conv 1*1 stride=2*2
Conv 1*1 stride=2*2
Input (299*299*3)

Fully-connected (softmax) 10
Fully-connected (relu) 64
GlobalAveragePooling
Relu
SeparableConv 2048, 3*3
Relu
SeparableConv 1536, 3*3
Relu
SeparableConv 1024, 3*3
Relu
SeparableConv 728, 3*3
Relu
MaxPooling 3*3, stride=2*2

Exit Flow

Relu
SeparableConv 728, 3*3
Relu
SeparableConv 1024, 3*3
Relu
SeparableConv 728, 3*3
Relu
Conv 1*1 stride=2*2
fully-connected (relu) 64
fully-connected (softmax) 10

Middle Flow

Anselmo Ferreira, Han Chen, Bin Li and Jiwu Huang
STEP #3 MERGED CHARACTERIZATIONS CNN-BASED PROCESSING
1-D CNN (Inception Frankenstein)

- We used the outputs from the 2CNNs fully connected layers as an input for a very simple 1-D CNN with the following characteristics:
  1. 512-D inputs (256-D from Inception-Resnet and 256-D from Xception).
  2. The network is trained using the RMSPROP [34] algorithm for updating weights, with an early stopping criterion on 100 epochs with a batch size of 32 samples.
  3. The image source identification is done by majority voting of classified blocks.
1-D CNN (Inception Frankenstein)

- We used the outputs from the 2CNNs fully connected layers as an input for a very simple 1-D CNN with the following characteristics

1. 512-D inputs (256-D from Inception-Resnet and 256-D from Xception).
2. The network is trained using the RMSPROP [34] algorithm for updating weights, with an early stopping criterion on 100 epochs with a batch size of 32 samples.
3. The image source identification is done by majority voting of classified blocks.
1-D CNN (Inception Frankenstein)

- We used the outputs from the 2CNNs fully connected layers as an input for a very simple 1-D CNN with the following characteristics:
  1. 512-D inputs (256-D from Inception-Resnet and 256-D from Xception).
  2. The network is trained using the RMSPROP [34] algorithm for updating weights, with an early stopping criterion on 100 epochs with a batch size of 32 samples.
  3. The image source identification is done by majority voting of classified blocks.
1-D CNN (Inception Frankenstein)

We used the outputs from the 2CNNs fully connected layers as an input for a very simple 1-D CNN with the following characteristics:

1. 512-D inputs (256-D from Inception-Resnet and 256-D from Xception).
2. The network is trained using the RMSPROP [34] algorithm for updating weights, with an early stopping criterion on 100 epochs with a batch size of 32 samples.
3. The image source identification is done by majority voting of classified blocks.
1-D CNN (Inception Frankenstein)

Figure 4: 1D-CNN applied on merged 2-D CNN Outputs
1 Motivation

2 Proposed Method

3 Experimental Setup

4 Results

5 Conclusion
Methodology and Datasets

- **Experiment #1:** Cross Dataset on camera model identification considering two datasets:

  1. **DATASET 1:** IEEE Signal Processing Cup: Forensic Camera Model Identification Challenge [35], containing 2740 JPEG images from 10 cameras. One individual camera per model.
  2. **DATASET 2:** Flickr images from the same camera models in **DATASET 1**. More than one individual camera per model.
Dataset and methodology

- **Experiment #2**: Applying trained models on $DATASET_1$ at kaggle benchmark [36].
- **Experiment #3**: 2-fold cross validation on specific camera identification considering : the Dresden Dataset [30].
1 Motivation

2 Proposed Method

3 Experimental Setup

4 Results

5 Conclusion
Experiment #1: mean results on pristine images

<table>
<thead>
<tr>
<th>Rank</th>
<th>Method</th>
<th>BLOCK % (1)</th>
<th>BLOCK % (2)</th>
<th>MEAN BLOCK %</th>
<th>IMAGE % (1)</th>
<th>IMAGE % (2)</th>
<th>MEAN IMAGE %</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>PROPOSED-METHOD</td>
<td>89.83%</td>
<td>99.37%</td>
<td>94.60%</td>
<td>92.51%</td>
<td>99.81%</td>
<td>96.16%</td>
</tr>
<tr>
<td>2</td>
<td>NASNET [31]</td>
<td>83.25%</td>
<td>98.11%</td>
<td>90.68%</td>
<td>86.05%</td>
<td>99.67%</td>
<td>92.86%</td>
</tr>
<tr>
<td>3</td>
<td>MOBILE-NET [12]</td>
<td>82.86%</td>
<td>97.91%</td>
<td>90.39%</td>
<td>86.42%</td>
<td>99.01%</td>
<td>92.72%</td>
</tr>
<tr>
<td>4</td>
<td>INCEPTION-V4 [23]</td>
<td>83.59%</td>
<td>98.26%</td>
<td>90.93%</td>
<td>85.21%</td>
<td>99.48%</td>
<td>92.35%</td>
</tr>
<tr>
<td>5</td>
<td>DENSENET [13]</td>
<td>81.22%</td>
<td>97.87%</td>
<td>89.55%</td>
<td>84.37%</td>
<td>99.05%</td>
<td>91.71%</td>
</tr>
<tr>
<td>6</td>
<td>RESNET [10]</td>
<td>79.82%</td>
<td>97.05%</td>
<td>88.44%</td>
<td>83.57%</td>
<td>98.97%</td>
<td>91.27%</td>
</tr>
<tr>
<td>7</td>
<td>SE_RESNET [10]</td>
<td>77.58%</td>
<td>95.02%</td>
<td>86.30%</td>
<td>84.45%</td>
<td>98.06%</td>
<td>91.26%</td>
</tr>
<tr>
<td>8</td>
<td>CHEN_NET [5]</td>
<td>78.88%</td>
<td>96.72%</td>
<td>87.80%</td>
<td>82.84%</td>
<td>98.79%</td>
<td>90.82%</td>
</tr>
<tr>
<td>9</td>
<td>RESNEXT [29]</td>
<td>78.99%</td>
<td>96.16%</td>
<td>87.58%</td>
<td>83.17%</td>
<td>97.91%</td>
<td>90.54%</td>
</tr>
<tr>
<td>10</td>
<td>DUAL_PATH_NETWORK [6]</td>
<td>75.54%</td>
<td>93.27%</td>
<td>84.41%</td>
<td>80.65%</td>
<td>97.15%</td>
<td>88.90%</td>
</tr>
<tr>
<td>11</td>
<td>BONDI_NET [4]</td>
<td>66.93%</td>
<td>81.31%</td>
<td>74.12%</td>
<td>80.54%</td>
<td>94.63%</td>
<td>87.59%</td>
</tr>
<tr>
<td>12</td>
<td>TUAMA_NET [28]</td>
<td>60.26%</td>
<td>75.39%</td>
<td>67.83%</td>
<td>64.85%</td>
<td>88.79%</td>
<td>76.82%</td>
</tr>
</tbody>
</table>

Table 1: Experiments results considering block and image classification after majority voting of predicted block labels.
Experiment #1: manipulated images scenario

Figure 5: Experiments results of the proposed approach, best individual proposed models and two best baseline solutions considering 2-fold cross validation on manipulated images.
Experiment #2: validation on kaggle benchmark

Results Considering the Classification of Pristine and Manipulated Images

IEEE VI SPS Challenge Testing Dataset

<table>
<thead>
<tr>
<th>Approach</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>MOBILENET [12]</td>
<td>81.83%</td>
</tr>
<tr>
<td>NASNET [31]</td>
<td>86.71%</td>
</tr>
<tr>
<td>INCEPTION-RESNET [24]</td>
<td>87.98%</td>
</tr>
<tr>
<td>XCEPTION [7]</td>
<td>91.99%</td>
</tr>
<tr>
<td>PROPOSED-METHOD</td>
<td>93.29%</td>
</tr>
</tbody>
</table>

Figure 6: Experiments results in the IEEE Signal Processing Society challenge on camera model identification held on kaggle public benchmark [36]
Experiment #3: mean results on Dresden Dataset (pristine images)

<table>
<thead>
<tr>
<th>Rank</th>
<th>Method</th>
<th>BLOCK % (1)</th>
<th>BLOCK % (2)</th>
<th>MEAN BLOCK %</th>
<th>IMAGE % (1)</th>
<th>IMAGE % (2)</th>
<th>MEAN IMAGE %</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>CHEN_NET [5]</td>
<td>93.92%</td>
<td>93.90%</td>
<td>93.91%</td>
<td>96.43%</td>
<td>96.48%</td>
<td>96.46%</td>
</tr>
<tr>
<td>2</td>
<td>PROPOSED-METHOD</td>
<td>95.38%</td>
<td>95.14%</td>
<td>95.26%</td>
<td>96.71%</td>
<td>96.01%</td>
<td>96.36%</td>
</tr>
<tr>
<td>3</td>
<td>XCEPTION</td>
<td>95.38%</td>
<td>94.99%</td>
<td>95.19%</td>
<td>96.71%</td>
<td>95.69%</td>
<td>96.20%</td>
</tr>
<tr>
<td>4</td>
<td>DENSENET [13]</td>
<td>94.35%</td>
<td>94.37%</td>
<td>94.36%</td>
<td>96.20%</td>
<td>95.11%</td>
<td>96.16%</td>
</tr>
<tr>
<td>5</td>
<td>INCEPTION-RESNET</td>
<td>94.86%</td>
<td>94.60%</td>
<td>94.73%</td>
<td>96.52%</td>
<td>95.78%</td>
<td>96.15%</td>
</tr>
<tr>
<td>6</td>
<td>NASNET [31]</td>
<td>93.95%</td>
<td>93.83%</td>
<td>93.89%</td>
<td>95.92%</td>
<td>95.64%</td>
<td>95.78%</td>
</tr>
<tr>
<td>7</td>
<td>MOBILE-NET [12]</td>
<td>93.66%</td>
<td>93.60%</td>
<td>93.63%</td>
<td>95.64%</td>
<td>95.87%</td>
<td>95.76%</td>
</tr>
<tr>
<td>8</td>
<td>SE_RESNET [10]</td>
<td>90.66%</td>
<td>90.79%</td>
<td>90.73%</td>
<td>95.04%</td>
<td>95.23%</td>
<td>95.14%</td>
</tr>
<tr>
<td>9</td>
<td>RESNET [10]</td>
<td>84.09%</td>
<td>85.83%</td>
<td>84.96%</td>
<td>93.79%</td>
<td>93.79%</td>
<td>93.79%</td>
</tr>
<tr>
<td>10</td>
<td>BONDI_NET [4]</td>
<td>85.35%</td>
<td>85.32%</td>
<td>85.34%</td>
<td>93.37%</td>
<td>93.47%</td>
<td>93.42%</td>
</tr>
<tr>
<td>11</td>
<td>TUAMA_NET [28]</td>
<td>84.89%</td>
<td>84.43%</td>
<td>84.66%</td>
<td>91.29%</td>
<td>90.64%</td>
<td>90.97%</td>
</tr>
<tr>
<td>12</td>
<td>DUAL_PATH_NETWORK [6]</td>
<td>3.70%</td>
<td>88.85%</td>
<td>46.28%</td>
<td>3.70%</td>
<td>94.21%</td>
<td>48.96%</td>
</tr>
<tr>
<td>13</td>
<td>RESNEXT [29]</td>
<td>18.00%</td>
<td>3.70%</td>
<td>10.85%</td>
<td>19.81%</td>
<td>3.70%</td>
<td>11.76%</td>
</tr>
</tbody>
</table>

Table 2: Experiments results considering block and image classification after majority voting of predicted block labels.
1 Motivation

2 Proposed Method

3 Experimental Setup

4 Results

5 Conclusion
Conclusions

- We proposed what is the first CNN of CNNs for camera model identification.
- The proposed method improved the best individual model in all scenarios.
Conclusions

- We proposed what is the first CNN of CNNs for camera model identification.
- The proposed method improved the best individual model in all scenarios.
Future Work

1. Consider pre-processing CNNs to better deal with downscaled data;

2. Evolve the CNN who processes merged pre-processed data to consider convolutions and more complex modules.

3. Consider image filtering operations and feature map augmentation as ‘pre-pre-processing’ operations.
Future Work

1. Consider pre-processing CNNs to better deal with downscaled data;

2. Evolve the CNN who processes merged pre-processed data to consider convolutions and more complex modules.

3. Consider image filtering operations and feature map augmentation as ‘pre-pre-processing’ operations.
Future Work

1. Consider pre-processing CNNs to better deal with downscaled data;

2. Evolve the CNN who processes merged pre-processed data to consider convolutions and more complex modules.

3. Consider image filtering operations and feature map augmentation as ‘pre-pre-processing’ operations.
References


References II


References III


References IV


References


References VI


References VII


References VIII


References IX


References


References XI


References XII


References XIII


References XIV


References XV


References XVI


References XVII
