Augmented Convolutional Feature Maps For Robust CNN-Based Camera Model Identification

CONTENTS

01 Introduction

Initial camera model identification

Initial CNNs designed to perform camera model identification operate by first suppressing an image's contents and extracting low-level using a fixed high-pass filter

→ Low-level features are then passed to a CNN

Constrained Convolutional layer

→low-level forensic features can be adaptively learned
→Initially proposed to learn image manipulation detection features
→Suppress image's content
→Learn diverse set of linear prediction residuals

Weakness

Often degrades significantly if the image is subjected to post-processing

01 Introduction

Manipulation detection area

Have encountered a similar problems with robustness to post-processing
(solved) utilize nonlinear residuals → potentially increase an algorithm's robustness to post-processing

Can the addition of low-level nonlinear residuals such as the median filter residual (MFR) increase the robustness of camera model identification?

Proposed

Robust CNN-based camera model identification in resampling, JPEG recompression
→"Augmented convolutional feature maps(ACFM)"
low level forensic feature extraction

02 Augmented Convolutional Feature Maps

Recently, CNN-based forensic approaches have been proposed to individually make use of adaptive linear prediction-error feature extractors as well as fixed nonlinear residual extractors

Advantage

Adaptive linear feature extractor

Ability to learn a diverse set of prediction residual features that outperform the fixed linear residuals

Nonlinear residual feature

Have been experimentally shown to improve the robustness of manipulation detection CNNs to JPEG compression

Integrate these low-level nonlinear residuals into our CNN

02 Augmented Convolutional Feature Maps

Augmented convolutional feature maps(ACFM)

1. A fixed nonlinear residual feature extractor is placed in parallel with a set of constrained convolutional filters
2. Feature maps produced by the constrained convolutional layer Concatenated with the nonlinear feature residuals
3. Passed to a regular convolutional layer

Deeper convolutional layers in CNN

learn higher-level and associations features between these linear and nonlinear residuals

In training

nonlinear feature extractors are held constant

filters in the constrained convolutional layer are updated through SGD

→ linear prediction-error feature extractors are learned that compliment nonlinear residual features

→ expect to increase robustness of CNNs to post-processing

02 Augmented Convolutional Feature Maps

MFR (nonlinear feature extractor)

$$d(i,j) = x(i,j) - med_3(i,j),$$

Constrained convolutional layer (prediction-error filter constraints)

$$\begin{cases} \boldsymbol{w}_k^{(1)}(0,0) = -1, \\ \sum_{m,n\neq 0} \boldsymbol{w}_k^{(1)}(m,n) = 1, \end{cases}$$

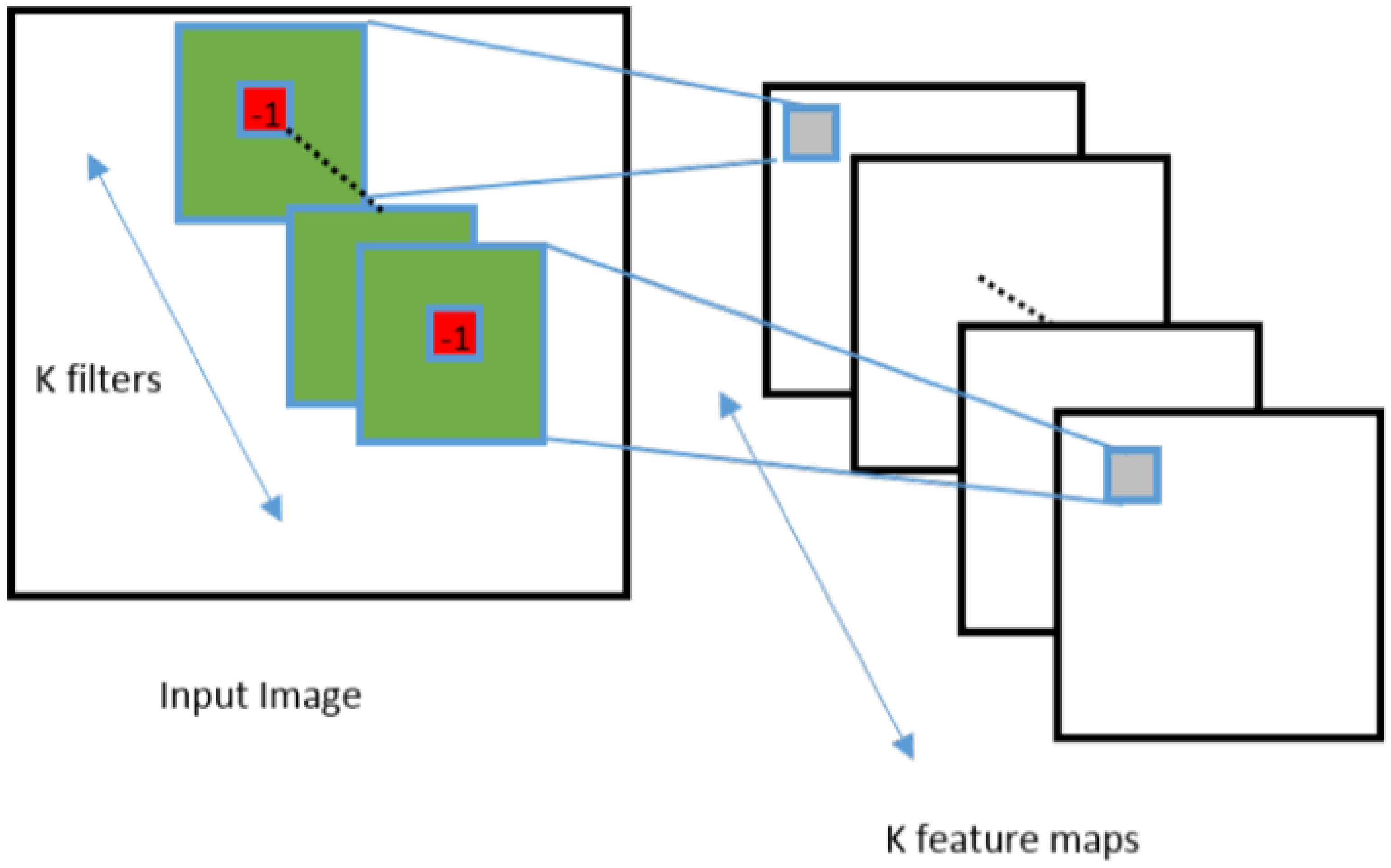
Algorithm 1 Training algorithm for constrained convolutional layer

1: *Initilize* $\boldsymbol{w}_k$*'s using randomly drawn weights*
2: i=1
3: **while** $i \leq max\_iter$ **do**
4: *Do feedforward pass*
5: *Update filter weights through stochastic gradient descent and backpropagate errors*
6: *Set* $\boldsymbol{w}_k(0,0)^{(1)} = 0$ *for all K filters*
7: *Normalize* $\boldsymbol{w}_k^{(1)}$*'s such that* $\sum_{\ell,m\neq 0} \boldsymbol{w}_k^{(1)}(\ell,m) = 1$
8: *Set* $\boldsymbol{w}_k(0,0)^{(1)} = -1$ *for all K filters*
9: i = i+1
10: **if** training accuracy converges **then**
11: exit
12: **end**

02 Augmented Convolutional Feature Maps

Constrained convolutional layer (prediction-error filter constraints)

$$\begin{cases} \boldsymbol{w}_k^{(1)}(0,0) = -1, \\ \sum_{m,n\neq 0} \boldsymbol{w}_k^{(1)}(m,n) = 1, \end{cases}$$



Prediction-error filter

→ Pixel value at center of filter window, then subtract this central value to produce the prediction error

03 Network Architecture

Propose: ACFM-based CNN

Ability

- ✓ Jointly suppress an image’s content and adaptively learn low-level linear residual features while training the network
- ✓ Perform convolutional feature maps augmentation using linear and nonlinear residuals
- ✓ Extract higher-level features through deep layers
- ✓ Learn new association between higher-level augmented feature

03 Network Architecture

Only Constrained Convolutional layer

→May not capture all camera model identification features.

Using "CFMA"

→1st channel (green channel image)

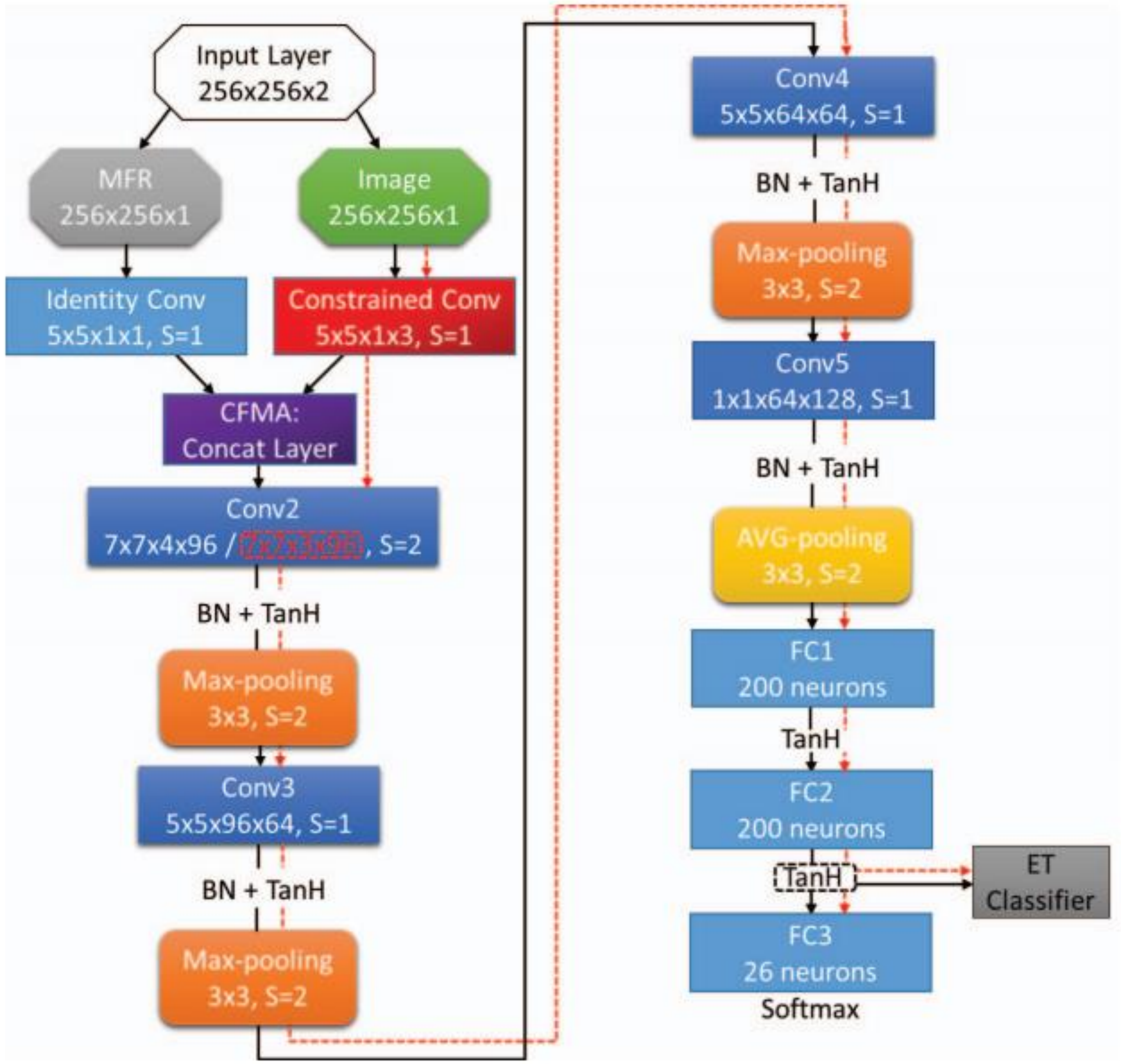
→2nd channel (MFR channel)

→Concatenate

2nd FC layer

→Use activation levels of each neuron as a set of deep features

→Passed to an extremely randomized trees(EZ)



04 Experiment

Dataset

Dresden Image Database

→ 15000 images (26 camera models)

→ Train: randomly select 12000 images

→ Test: 3000 images

→ Divided images into 256x256 pixel patches

→ Retained all the 36 central patches form the green layer of each image

→ 432000 patches (train)

↓

Image editing operation

Parameter

Batch size: 64

SGD: momentum = 0.9, decay = 0.0005

Learning rate = 10e-3(decrease every 4epochs, factor = 0.5)

Epochs = 44

04 Experiment

Table 1: CNN's identification rate on processed images using Softmax layer (top) and Extremely Randomized Trees (ET) classifier (bottom).

	Resampling			Resampling + JPEG (QF=90)			JPEG	Original
Methods	120%	90%	50%	120%	90%	50%	QF = 90	–
ACFM-based CNN	**97.14%**	**95.93%**	**90.75%**	93.86%	91.42%	**79.31%**	**97.26%**	**98.26%**
NonACFM-based CNN	96.75%	95.76%	87.70%	**94.94%**	**91.89%**	75.68%	97.23%	98.24%
HPF-based CNN	95.94%	95.68%	87.54%	90.45%	83.96%	67.16%	96.00%	97.52%
Top: Softmax-based CNN; Bottom: ET-based CNN								
ACFM-based CNN	**97.61%**	**96.44%**	**91.47%**	94.71%	92.10%	**79.74%**	**97.63%**	**98.58%**
NonACFM-based CNN	97.28%	96.38%	88.88%	**95.50%**	**92.50%**	76.05%	97.60%	98.52%
HPF-based CNN	96.47%	96.14%	88.67%	91.31%	84.71%	67.42%	96.36%	97.83%

ACFM-based CNN: complete proposed method
Non ACFM-based CNN: not use MFR layer, only use constrained convolutional layer
HFF-based CNN: using fixed high pass filter