Application of DenseNet in Camera Model Identification and Post-processing Detection

Sangmyung Univ.
201910976 Sunjung AN

CONTENTS

01 | Introduction

01 Introduction

Propose

- Present a DenseNet pipeline to solve the problem of identifying the source camera model of an image
- Be very robust for identifying the source camera model, even when the original image is post-processed
- **In the absence of metadata and the presence of extensive post-processing in images.**
- **Focus has towards detecting intrinsic camera features. Ex) CFA pattern, IQM , interpolation algorithm..**
- Use a number of data-augmentation schemes
  - ✓ gamma correction
  - ✓ JPEG compression
  - ✓ re-scaling extracting patches
  - ✓ randomly cropping
  - ✓ flipping the training image

02 Materials

2.1 Description of Datasets
2.2 Data Augmentation

02 Materials

2.1 Description of Datasets

Dataset-1

Camera Model Identification Datasets provided for the IEEE Signal Processing Cup 2018

- ✓ 10 different camera models having 275 images for each → all of them 2750
- ✓ Without any labels
- ✓ Collected from Flickr during the open competition phase of SP Cup 2018

Using 2640 images of size 512 x 512, among which 1320 are unaltered and the rest are manipulated extremally

Dataset-2

Dresden Image Database

- ✓ 27 different camera models

Camera Model	SP Cup Data (No. of Images)	Flickr Data (No. of Images)
HTC-1-M7	275 × 10	746
iPhone-4s		499
iPhone-6		548
LG-Nexus-5x		405
Motorola-Droid-Maxx		549
Motorola-Nexus-6		650
Motorola-X		344
Samsung-Galaxy-Note3		274
Samsung-Galaxy-S4		1137
Sony-NEX-7		557
Sub-Total	2750	5709
Grand-Total	8459	

Table 1. SP Cup data & Flickr data (Dataset I)

02 Materials

2.2 Data Augmentation

Post-Processing

- JPEG-Compression with quality factor 90% and 70%
- Resizing by a factor of 0.5, 0.8, 1.5 and 2.0
- Gamma-Correction using r as 0.8 and 1.2
- **EMD**

Camera Model	SP Cup Data (No. of Images)	Flickr Data (No. of Images)
HTC-1-M7	275 × 10	746
iPhone-4s		499
iPhone-6		548
LG-Nexus-5x		405
Motorola-Droid-Maxx		549
Motorola-Nexus-6		650
Motorola-X		344
Samsung-Galaxy-Note3		274
Samsung-Galaxy-S4		1137
Sony-NEX-7		557
Sub-Total	2750	5709
Grand-Total	8459	

Table 1. SP Cup data & Flickr data (Dataset I)

03 Methods

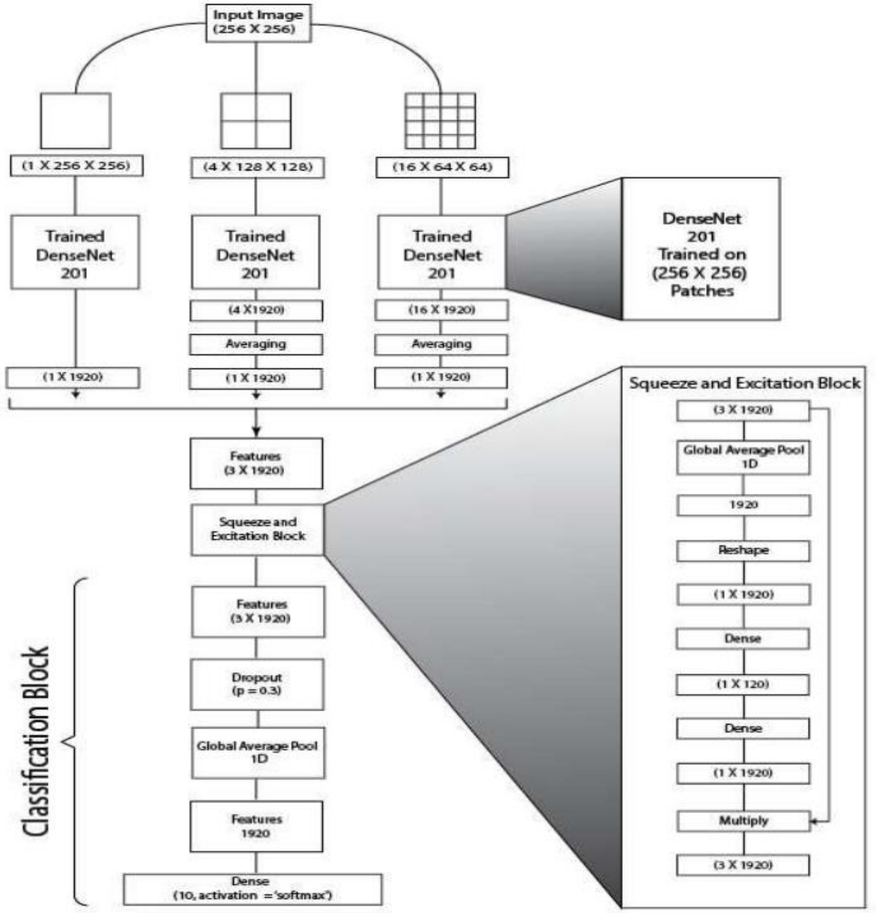
3.1 Model Proposal
3.2 Training Data Generation
3.3 Architecture

03 Methods

3.1 Model Proposal

OUTLINE

1. Select patches of size 256 x 256
2. Train DenseNet-201 architecture with patches of size 256x256
3. Extract feature from second to the last layer
4. Extract all non-overlapping patches of size 128 x 128 and 64 x 64 from each training image
5. Averaging (4 x 1920) → (1 x 1920), (16 x 1920) → (1 x 1920)
6. Concatenate the feature vectors(3x1920)
7. SE block
8. Classification block



03 Methods

3.2 Training Data Generation

Reason to extract patches

1. It results in more data to train our network, thus making the training process more generalized
2. To generate multiple predictions for a given test image
3. Training patches of small size → prevents our network from learning domain special features of the image

Select patch algorithm

- ✓ Compute the quality value of a patch
- ✓ Select 20 patches of 256 x 256 with highest Q value

$$Q(\mathcal{P}) = \frac{1}{3} \sum_{c \in [R,G,B]} \left[\alpha \cdot \beta \cdot (\mu_c - \mu_c^2) + (1 - \alpha) \cdot (1 - e^{\gamma \sigma_c}) \right]$$

- ✓ Quality measure tends to be lower for overly saturated or flat patches
- ✓ It is higher for textured patches showing some statistical variance

parameter

a = 0.7, B = 4, r = ln(0.01), P = patch
μc = mean and σc = standard deviation

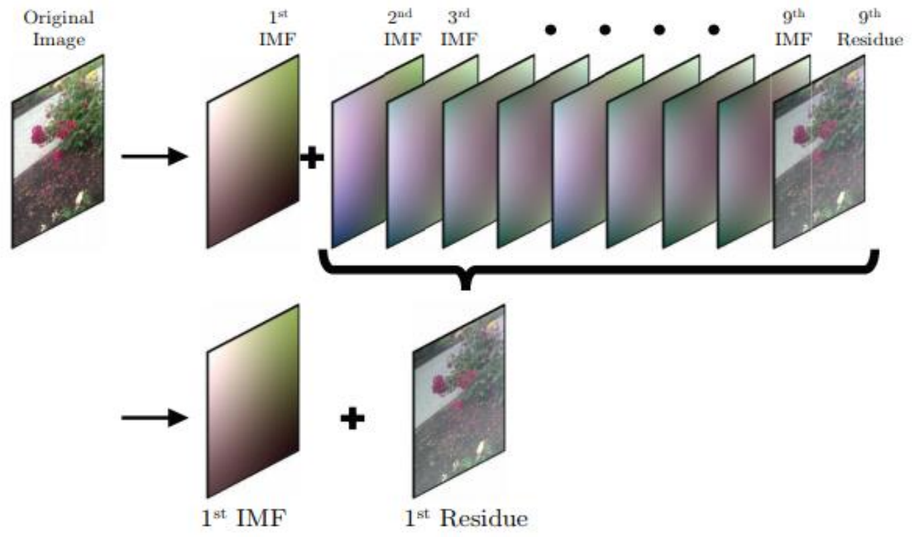
03 Methods

3.2 Training Data Generation

Empirical Mode Decomposition

- ✓ Input signal is decomposed into IMF's and Residue
- ✓ Used FastRBF to interpolate upper and lower envelopes of scattered local maxima and minima from I(m,n).
- ✓ The mean of envelopes is then subtracted from the image to get the IMF

$$I(m,n) = \sum_{j=1}^{L} \mathrm{IMF}_j(m,n) + \mathrm{Res}_L(m,n)$$



$$s(\mathbf{x}) = \mathrm{p}_m(\mathbf{x}) + \sum_{i=1}^{N} \lambda_i \phi(\|\mathbf{x} - \mathbf{x_i}\|)$$

EMD essentially works as a denoising scheme by removing random high-frequency noise components from the image data → **using EMD more than once may prove to be detrimental**

03 Methods

3.3 Architecture

DenseNet-201

to preserve image information throughout the network, the output of each layer is propagated to all of the layers in front of it.

The camera-model features inherent in an image are very subtle and minute features of the image

→Invariably becomes dependent on the image content of the device specific noise, as all of the minute statistical information is lost when the image is propagated through consecutive layers

→This problem is alleviated in the DenseNet through the use of dense connections

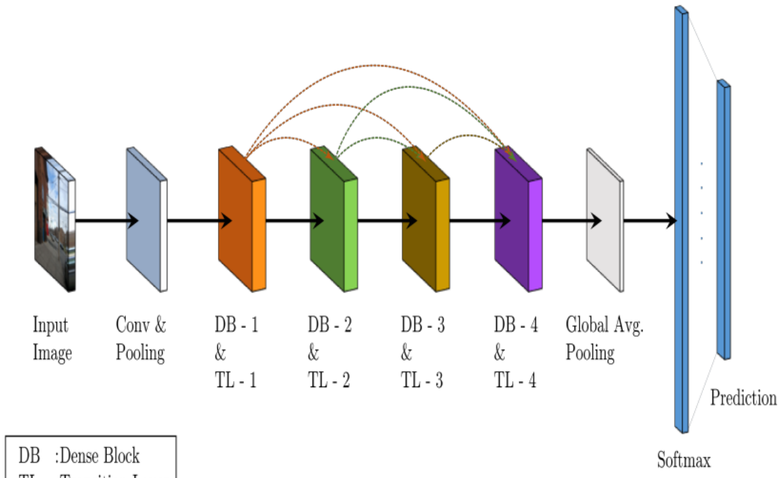


Table 2. Architecture of DenseNet-201

Layers	DenseNet-201
Convolution	7×7 conv, stride 2
Pooling	3×3 max pool, stride 2
Dense Block (1)	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 6$
Transition Layer (1)	1×1 conv
	3×3 max pool, stride 2
Dense Block (2)	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 12$
Transition Layer (2)	1×1 conv
	2×2 average pool, stride 2
Dense Block (3)	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 48$
Transition Layer (3)	1×1 conv
	2×2 average pool, stride 2
Dense Block (4)	$\begin{bmatrix} 1 \times 1 \text{ conv} \\ 3 \times 3 \text{ conv} \end{bmatrix} \times 32$
Layer	Global Average Pooling
Classification	Softmax

Consist of 4 dense blocks, each with a growth-rate of 32

03 Methods

3.3 Architecture

SE block (Squeeze and Excitation Block)

The SE block performs feature recalibration, through which it can learn to use global information to selectively emphasize informative features and suppress less useful ones, without changing the dimensions of the feature vector

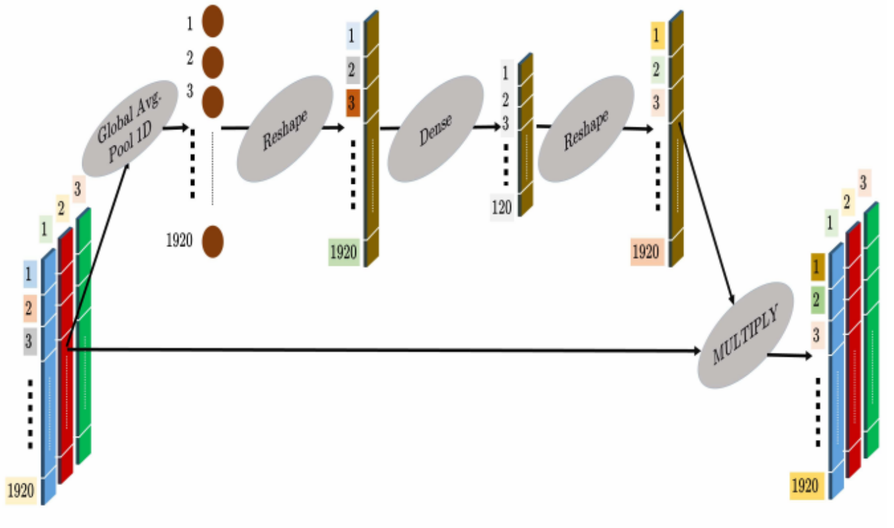


Figure 4. Illustration of a Squeeze-and-Excitation Block.

Classification block

1. (3x1920) produced at the output of the SE block is then passed through a Dropout layer with 30%
2. Global Average Pooling -> (1x1920)
3. Softmax -> 10 classes

Excitation operation learned for each channel by a self-gating mechanism based on channel dependence
Govern the excitation of each channel

→ Feature maps are then reweighted to generate the output of the SE block

04

Experiments & Results

4.1 Phase-1
4.2 Phase-2
4.3 Phase-3

04 Experiments & Results

4.1 Phase-1

Experiments

- Dataset-1 20 patches of size 256 x 256
- Train: 85%, validation: 15%
- Stochastic Gradient Descent: momentum of 0.9, learning rate = 10e-3
- The learning rate is decreased by a factor of 10e-1 if the validation loss have not decreased in 2 successive epochs
  (when learning rate is reduced to 10e-7, training is stopped)

Results

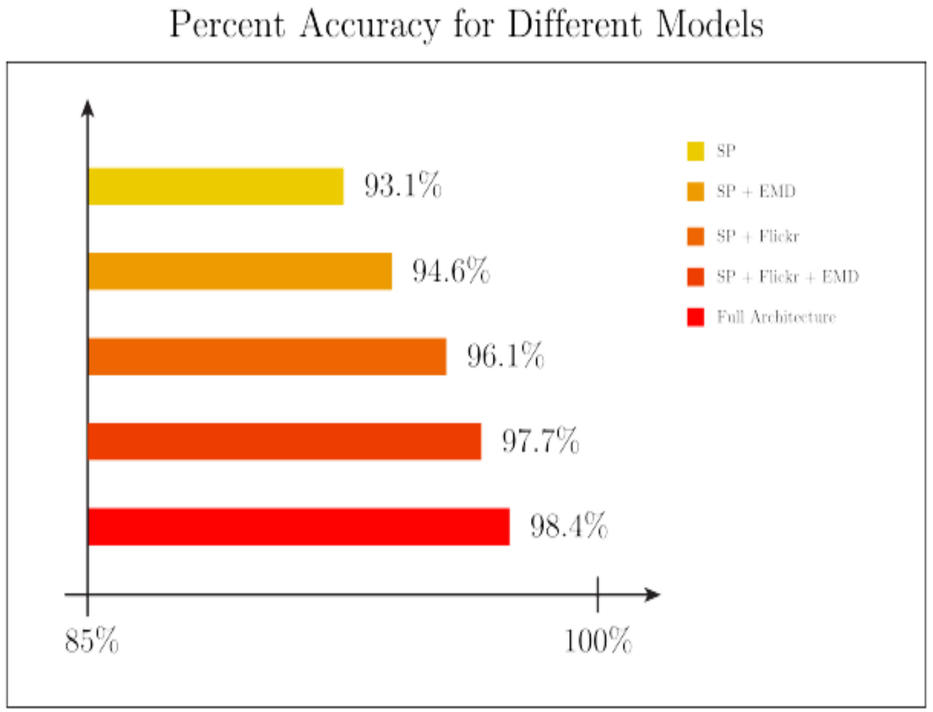
- Score = 0.7 * (Accruracy of Unaltered Images) + 0.3 * (Accuracy of Manipulated Images)
- Higher accuracy is produced for larger input sizes
- Residual camera model information left after cropping an image to this size are minimal
- EMD augmented images in increasing the accuracy
- This IMF likely to have some correlations to device-level features

04 Experiments & Results

4.1 Phase-1

Table 3. Detection Accuracy of Camera-Models for different Input Sizes

Network	Accuracy		
	Unaltered (70%)	Manipulated (30%)	Total (100%)
DenseNet-201 (64 × 64)	67.16%	27.43%	94.59%
DenseNet-201 (128 × 128)	68.33%	28.61%	96.94%
DenseNet-201 (256 × 256)	68.75%	28.82%	97.57%
DenseNet-201 (Final Layer Prediction Average)	69.12%	28.84%	97.96%
Full Pipeline	69.33%	29.04%	98.37%



04 Experiments & Results

4.2 Phase-2

Experiment

- Dataset-2 20 patches of size 256 x 256
- Transfer learning on Phase-1 (initial weights of the network from Phase-1)
- Not use our full pipeline for the dataset (not SE layer)
- Trained for the 27 classes

Result

- Overall accuracy of over 99% is achieved for 19 camera models by the 1st network of Phase-1
- Though the training dataset is very small compared to the dataset of Phase-1, but still DenseNet-201 is able to detect the camera models very accurately because of the learnt features from Phase-1

04 Experiments & Results

4.3 Phase-3

Experiment

- Used all images from Phase-1
- EMD data has not been included
- Sub-divided these data into 4 classes (Unaltered, Resized, JPEG Compressed and Gamma Corrected)
- Used 128 x 128 sized patches for training due to much higher prediction accuracy compared to other sizes

Result

- Tried to identify the 4 types of image-manipulations used on the image of Dataset-1
- There result have been obtained by using only DenseNet-201 architecture and 128 x 128 patch size

Table 4. Predictions of detected manipulations

	Unaltered	JPEG-Compr.	Gamma-Corrected	Resized
Unaltered	**90.07%**	3.49%	6.06%	0.38%
JPEG-Compr.	0.15%	**99.85%**	0%	0%
Gamma-Corrected	3.18%	0%	**96.75%**	0.07%
Resized	0%	0%	0%	**100%**