Source camera identification based on content-adaptive fusion residual networks

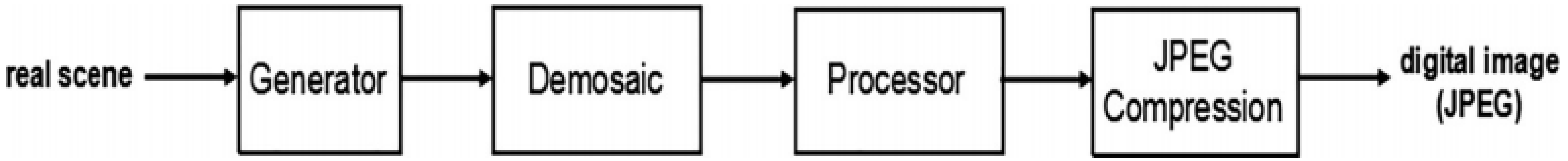
CONTENTS

01 Introduction

Need to considered for these methods based on SPN

1. Quality of SPN extracted from image depends on the image contents
2. Detection performance could be decreased with the reduction of image size

Process of generating the digital image



Generator(include SPN), Demosaic, JPEG → related with image contents

- Fingerprint left should be not same for the different image contents
- Separate the database into three subsets

Small-size images provide an effective reference for the splicing forgery

→ Propose a solution to identify the source camera of the small-size images

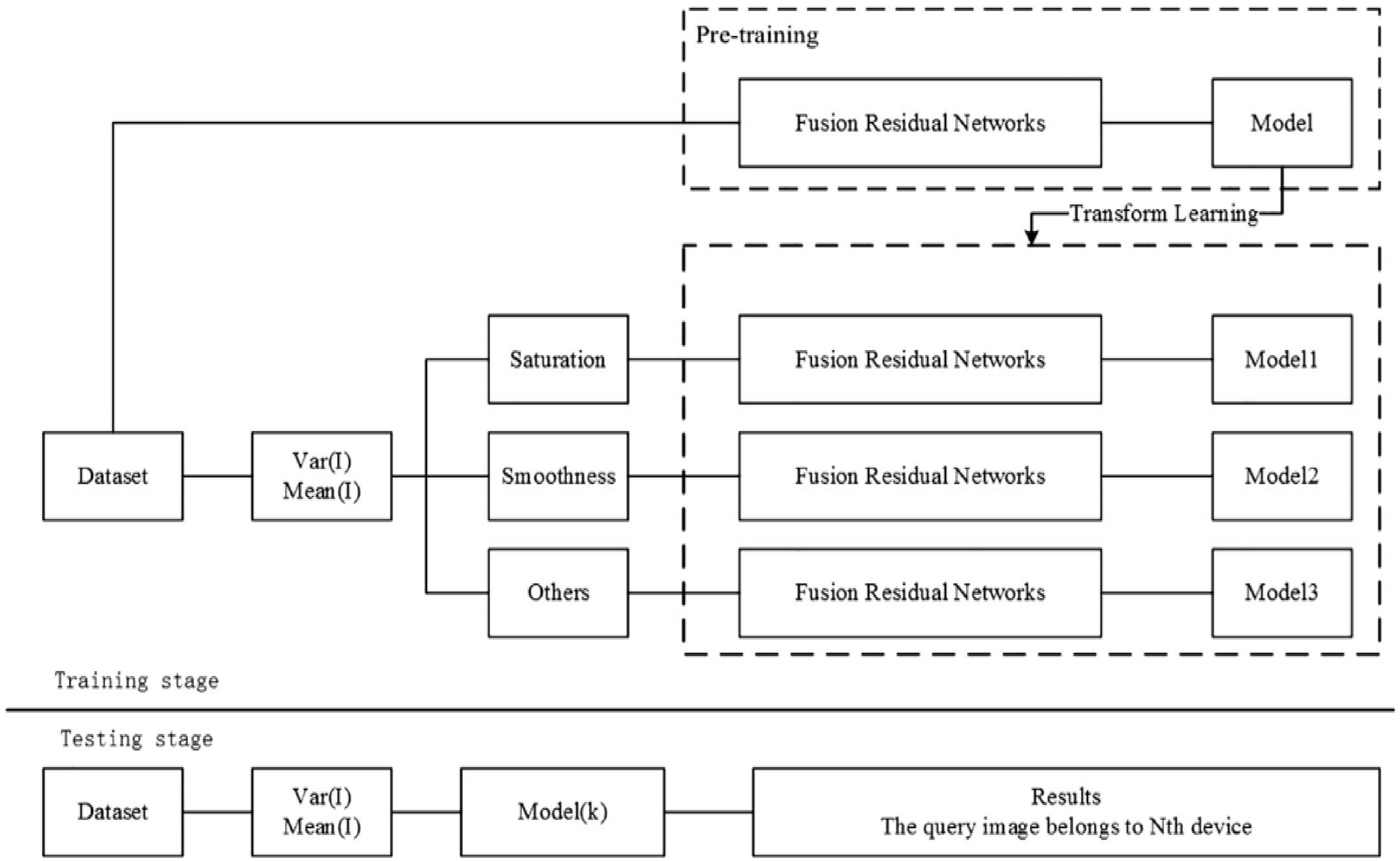
01 Introduction

Content-adaptive fusion residual networks

- Divide the images into three subsets
- Self learned in preprocessing
- FRN
- Transform learning
  (deal with limited training data)

Validate

- Camera brand identification
- Camera model identification
- Camera device identification

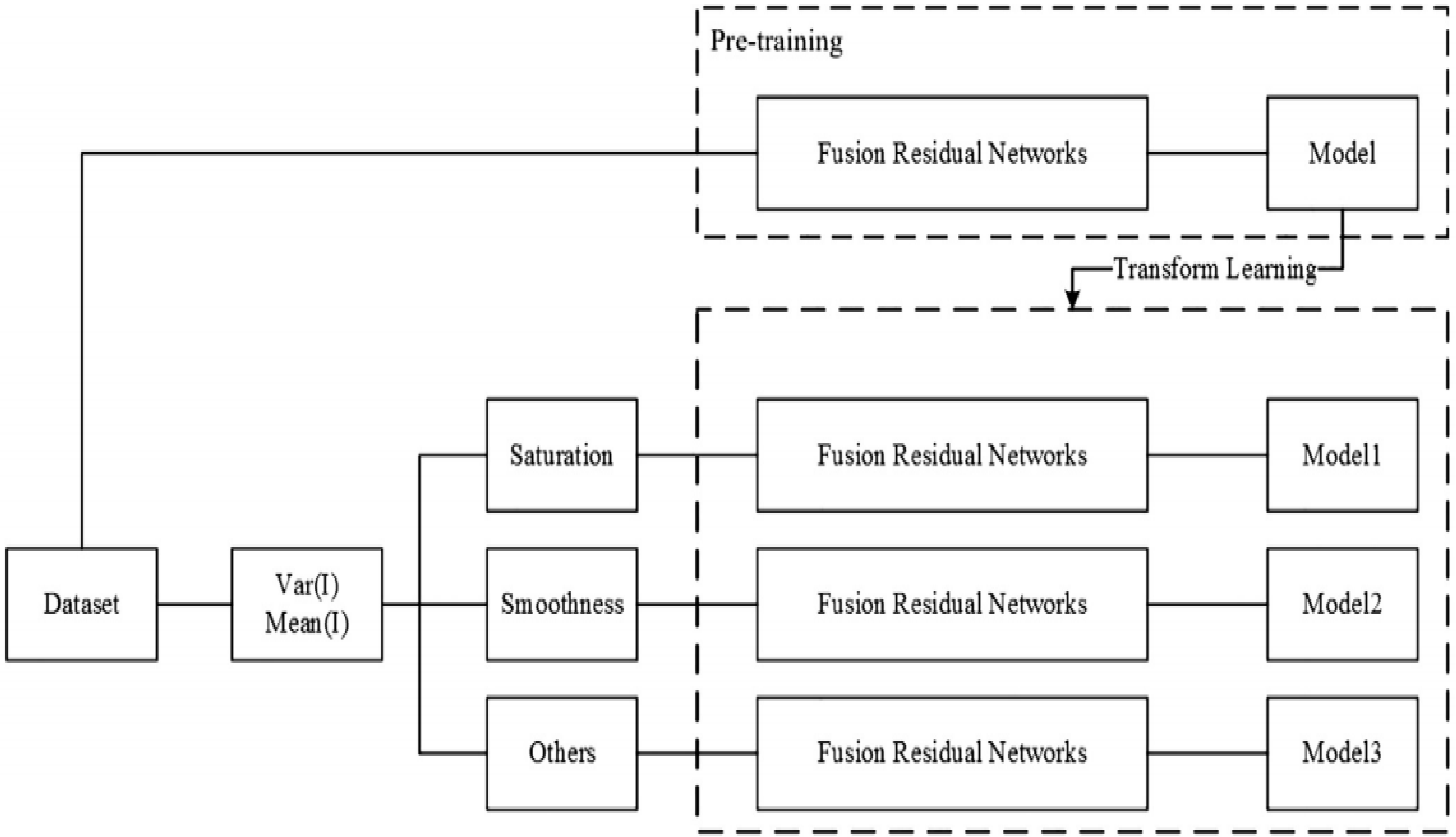


02 Proposed algorithm

In order to capture effective features for the different image contents
→ Fusion residual networks is designed

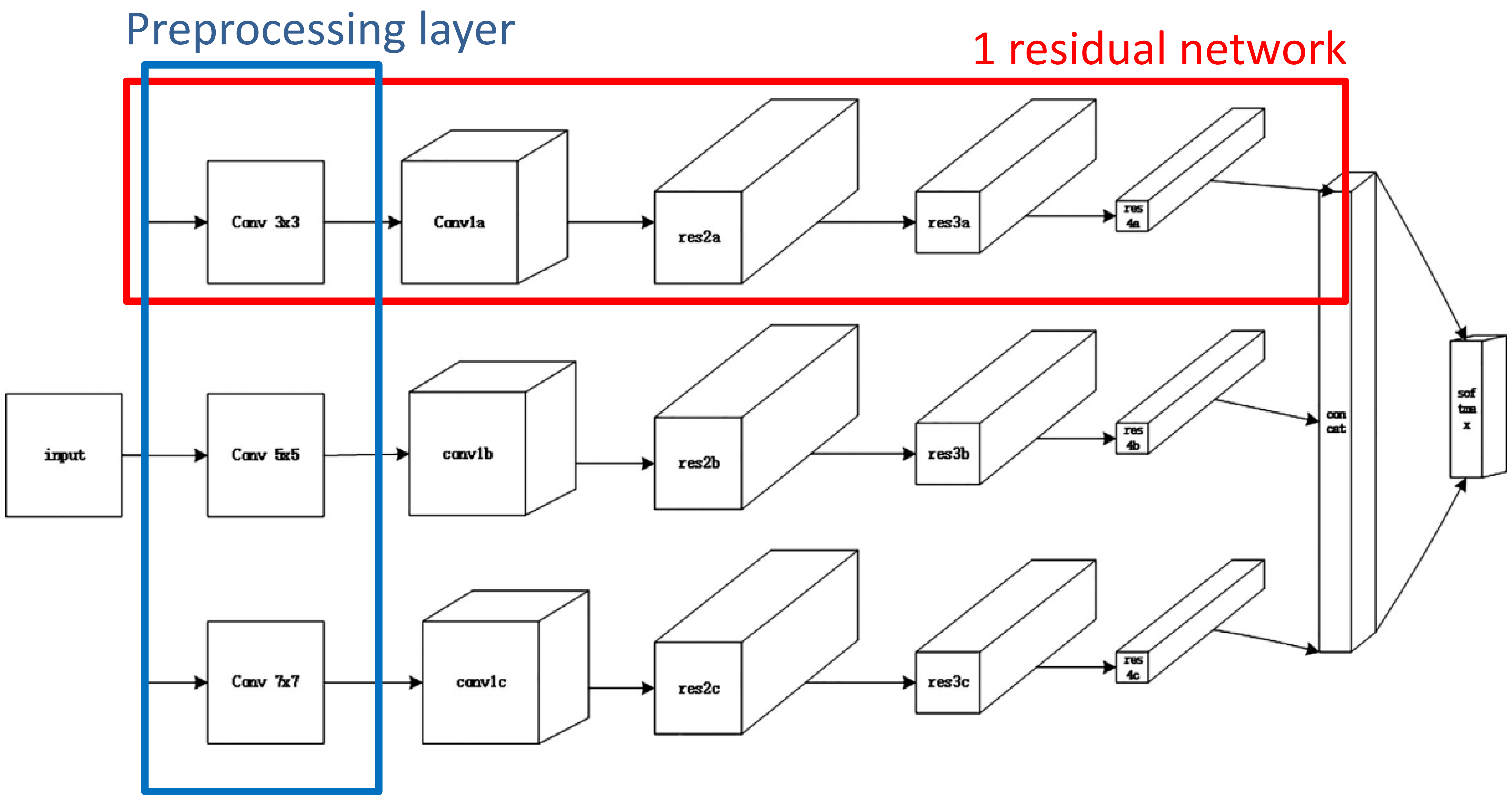
<Train>
1. pre-training
2. Divided into three subsets
3. Train by transfer learning

<Test>
1. Calculate mean, variance
2. Feed into the trained model



02 Proposed algorithm

In order to capture effective features for the different image contents
→ Fusion residual networks is designed



Fusion residual network

3 residual network(parallel)

1 self-learning filter

3 residual blocks

02 Proposed algorithm

Residual network

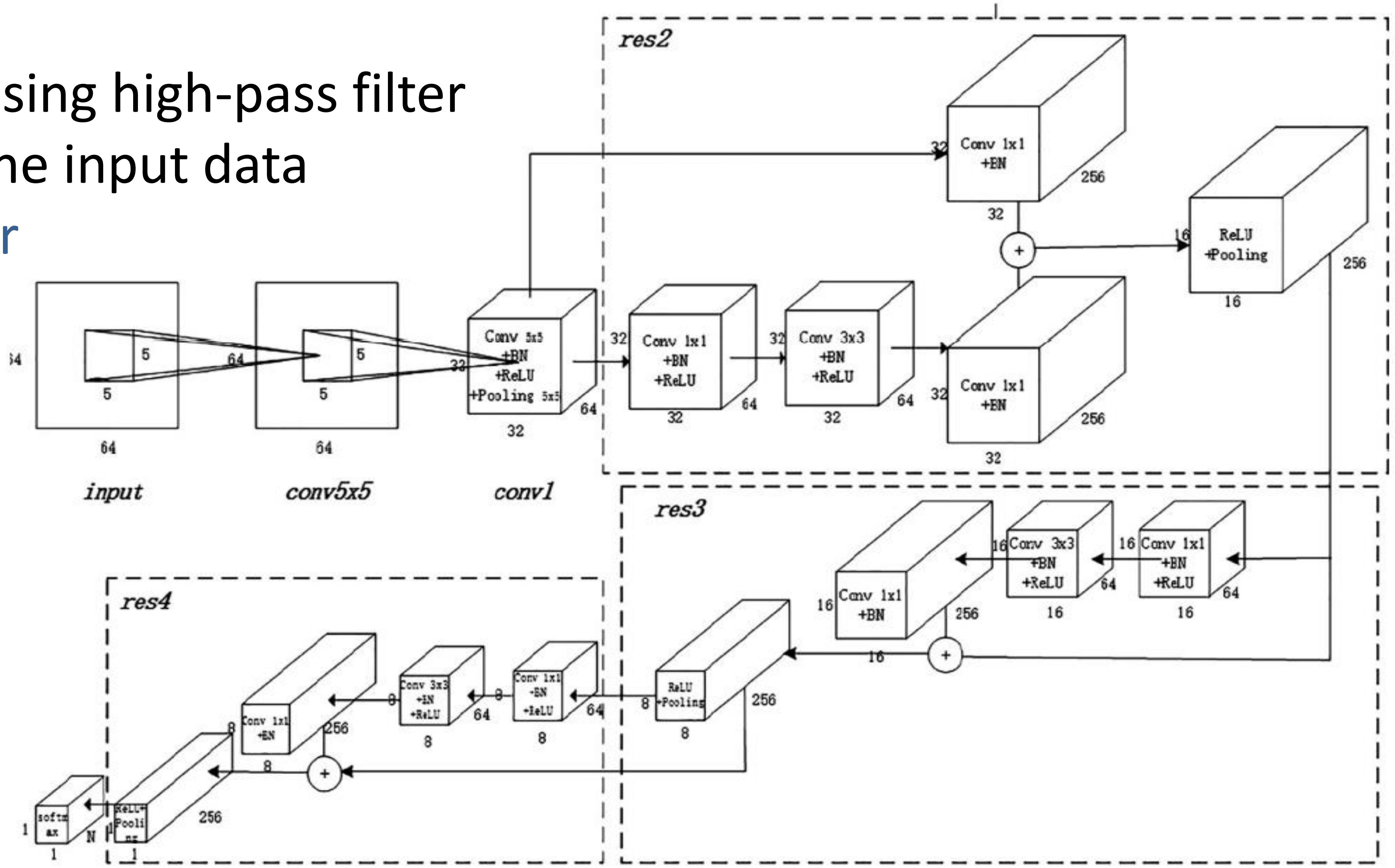
SPN is related to the image contents

→ In order to amplify the inter-class difference and reduce the impact of the image contents, using Preprocessing

→ Not the best way to preprocessing the input data using high-pass filter

→ Self-learning better feature representations from the input data

Replace the special filter(HPF) with convolutional layer



02 Proposed algorithm

Table 1
The parameters of the fusion residual networks.

Layername	Parameters		
input		64 × 64 × 3	
conv_	3 × 3 × 1 stride:1	5 × 5 × 1 stride:1	7 × 7 × 1 stride:1
conv1_	5 × 5 × 64 stride:1	5 × 5 × 64 stride:1	5 × 5 × 64 stride:1
ave_pooling	5 × 5 stride:2	5 × 5 stride:2	5 × 5 stride:2
res2_	1 × 1 × 64 stride:1	1 × 1 × 64 stride:1	1 × 1 × 64 stride:1
	3 × 3 × 64 stride:1	3 × 3 × 64 stride:1	3 × 3 × 64 stride:1
	1 × 1 × 256 stride:1	1 × 1 × 256 stride:1	1 × 1 × 256 stride:1
ave_pooling	5 × 5 stride:2	5 × 5 stride:2	5 × 5 stride:2
res3_	1 × 1 × 64 stride:1	1 × 1 × 64 stride:1	1 × 1 × 64 stride:1
	3 × 3 × 64 stride:1	3 × 3 × 64 stride:1	3 × 3 × 64 stride:1
	1 × 1 × 256 stride:1	1 × 1 × 256 stride:1	1 × 1 × 256 stride:1
ave_pooling	5 × 5 stride:2	5 × 5 stride:2	5 × 5 stride:2
res4_	1 × 1 × 64 stride:1	1 × 1 × 64 stride:1	1 × 1 × 64 stride:1
	3 × 3 × 64 stride:1	3 × 3 × 64 stride:1	3 × 3 × 64 stride:1
	1 × 1 × 256 stride:1	1 × 1 × 256 stride:1	1 × 1 × 256 stride:1
global_ave_pooling	8 × 8 stride:1	8 × 8 stride:1	8 × 8 stride:1
softmax		1 × n	

02 Proposed algorithm

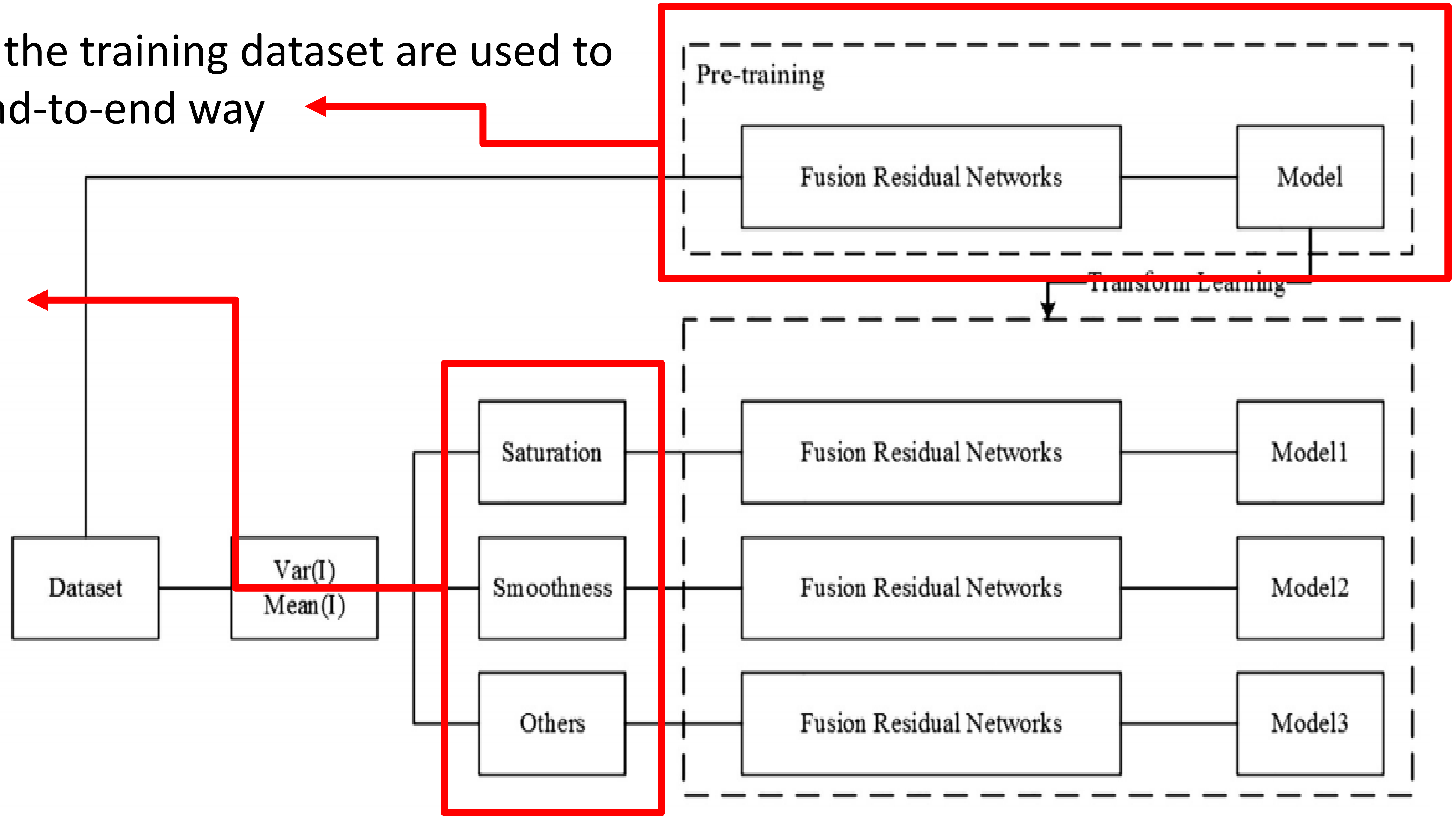
In pre-training state, all images form the training dataset are used to train a fusion residual networks in end-to-end way

Divide the image into three subsets

- saturation, smoothness, others

Train

- End-to-end way
- Transfer learning



02 Proposed algorithm

Divide the image into three subsets

- saturation, smoothness, others

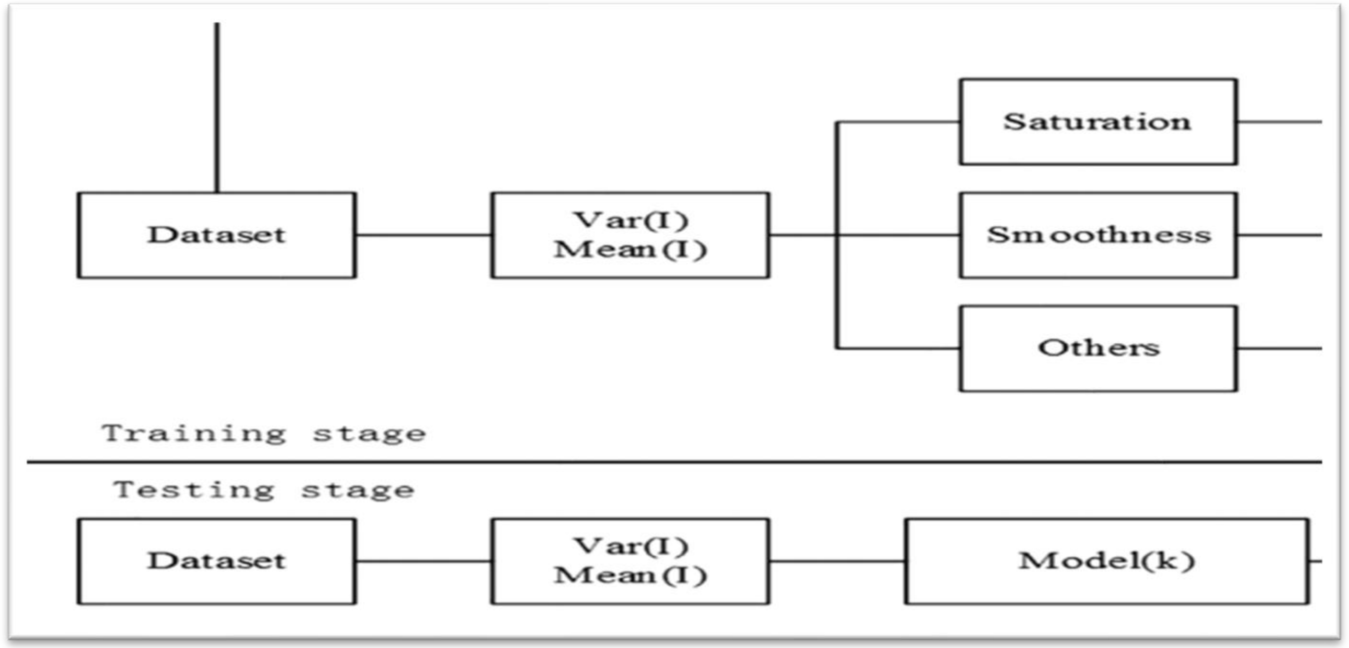
In grayscale

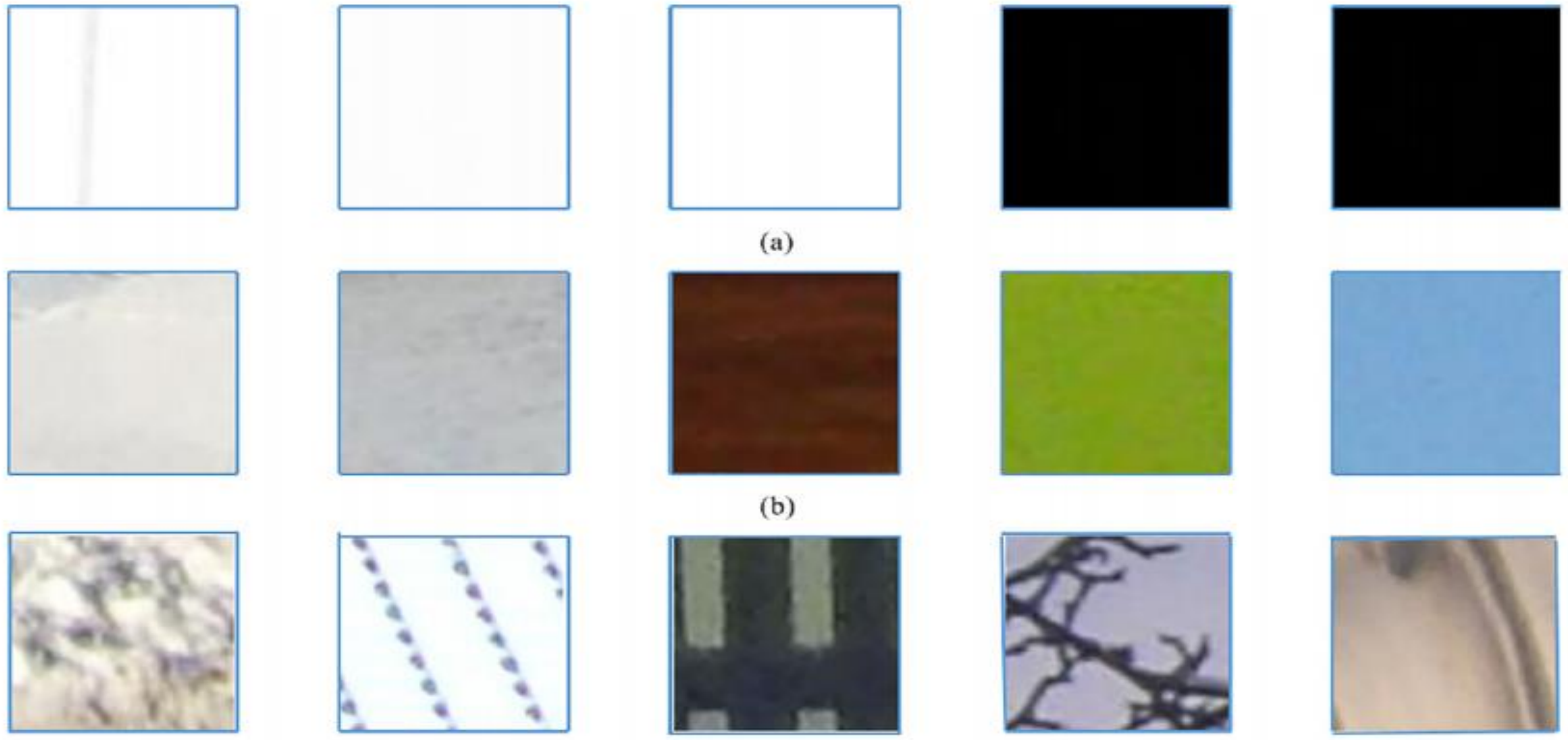
$$I_i \in \begin{cases} Subset1 = T_1(m, v) \\ Subset2 = T_2(m, v) \\ Subset3 = T_3(m, v) \end{cases} \quad (4)$$

M: mean
V: variance

$$\begin{cases} T_1(m, v) & m \in [0, 5] \bigcup [250, 255], v \in [0, 25] \\ T_2(m, v) & m \in [0, 5] \bigcup [250, 255], v \in [25, 50] \ || \\ & m \in (5, 250), v \in [0, 50] \\ T_3(m, v) & others \end{cases} \quad (5)$$

Subset1 → saturation
Subset2 → smoothness
Subset3 → Others




(a)
(b)
(c)

03 Experiments

Dataset

Dresden database

- Choose 13 devices
- Cut into non-overlapping 64x64
- 4 Train, 1 Test, 1 Validation
- Experiment 1: 2,757,888 patches
- Experiment 2 and 3: 818,748 patches

Table 2
The list of camera devices used.

ID	Camera devices	Original resolution
1	Kodak_M1063_0	3664 × 2748
2	Pentax_OptioA40_0	4000 × 3000
3	Nikon_CoolPixS710_1	4352 × 3264
4	Sony_DSC-H50_0	3456 × 2592
5	Olympus_mju_1050SW_2	3648 × 2736
6	Panasonic_DMC-FZ50_1	3648 × 2736
7	Agfa_Sensor530s_0	2560 × 1920
8	Ricoh_GX100_0	3648 × 2736
9	Samsung_NV15_0	3648 × 2736
10	Sony_DSC-W170_0	3648 × 2736
11	Sony_DSC-T77_0	3648 × 2736
12	Sony_DSC-T77_1	3648 × 2736
13	Sony_DSC-T77_2	3648 × 2736

Parameter

- (pre-training) Learning rate: 0.01 (decrease 10% for every 10000 iter)
- (transfer learning) learning rate: 0.001, max_iter: 500000, momentum: 0.9
- (Convolutional layers) weights initialization with Gaussian filter[expected value= 0, standard deviation = 0.01]
- (Convolutional layers) learned from the input data using mini-batch gradient descent
- Xavier filler is applied into Softmax layer

03 Experiments

Experiment 1 – camera brand identification

Select 9 camera devices

- CA-CNN: preprocessing layer → high pass filter, self learn convolutional kernels
- RN: residual networks
- FRN: paralleled residual networks
- CAF-CNN: three paralleled CA-CNNs
- CA-FRN: proposed
- Googlenet: high-pass filter in preprocessing

Table 3
The detection accuracy for camera brand identification.

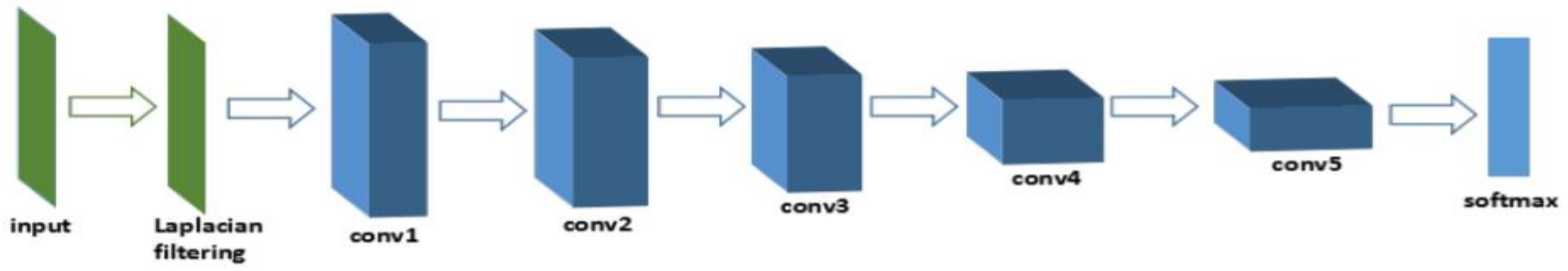
Type	Preprocessing	Ave_acc
CA-CNN	HP	81.62%
	Conv 3 × 3	87.72%
	Conv 5 × 5	90.11%
	Conv 7 × 7	90.68%
GoogleNet [21]	HP	91.60%
ResNet [27]	None	96.20%
RN	Conv 3 × 3	95.58%
	Conv 5 × 5	96.21%
	Conv 7 × 7	96.03%
CAF-CNN [22]	Conv3 5 7	94.17%
FRN	Conv3 5 7	96.26%
CA-FRN	Conv3 5 7(Res)	97.03%

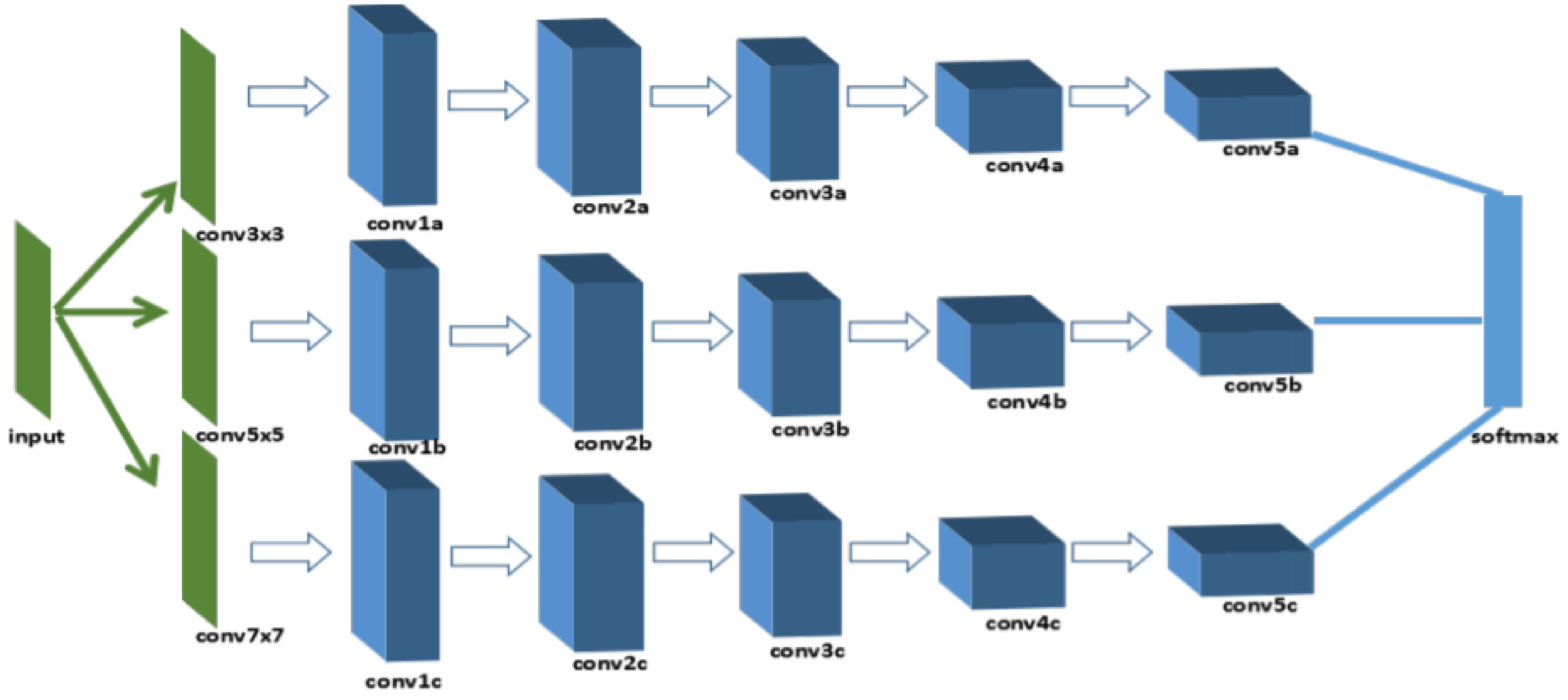
Table 4
The detection accuracy of content-adaptive fusion residual networks and fusion residual networks. The best results are highlighted in bold.

	Smoothness		Saturation		Others	
	FRN	CA-FRN	FRN	CA-FRN	FRN	CA-FRN
1	99.27%	99.09%	84.16%	80.83%	99.13%	99.57%
2	99.03%	99.76%	27.33%	42.55%	97.90%	99.46%
3	93.99%	97.51%	97.66%	93.92%	96.16%	97.49%
4	96.07%	98.02%	51.44%	66.93%	96.82%	95.11%
5	91.14%	89.95%	61.05%	61.18%	96.38%	97.65%
6	94.25%	96.27%	78.45%	85.63%	96.02%	98.19%
7	97.33%	94.9%	20.42%	97.16%	96.38%	98.15%
8	96.96%	98.27%	11.50%	97.08%	97.11%	97.83%
9	96.66%	96.59%	31.40%	46.61%	98.3%	96.95%
AVE	96.14%	**96.73%**	68.5%	**76.89%**	97.12%	**97.8%**

03 Experiments

CA-CNN (with Laplacian filter)





CAF-CNN (with self learned)

03 Experiments

Experiment 2 – camera model identification

Select: Sony_DSC-H50, Sony_DSCW170, Sony_DSC-T77
Finetune the model trained in the first experiment

Accuracy: 87.55%

Experiment 2 – camera device identification

Select: Sony_DSC-T77_0, Sony_DSC-T77_1, Sony_T77_2
Finetune the model trained in the first experiment

Accuracy: 73.27%

Mixed

Different camera brands
Different camera models(same brands)
Different camera devices(same models)

Select: Sony_DSC-T77_0, Sony_DSC-T77_1, Sony_DSC-H50_0, Olympus_mju_1050W_2, Panasonic_DMC-FZ50-1, Agfa_sensor530s+0, Ricoh_GX100_0, Samsung_NV15_0, Kodak_M1063_0

Accuracy: 92%