Camera Model Identification With The Use of Deep Convolutional Neural Networks

Amel TUAMA [2,3], Frédéric COMBY [2,3], and Marc CHAUMONT [1,2,3]
(1) University of Nîmes, France
(2) University Montpellier, France
(3) CNRS, Montpellier, France

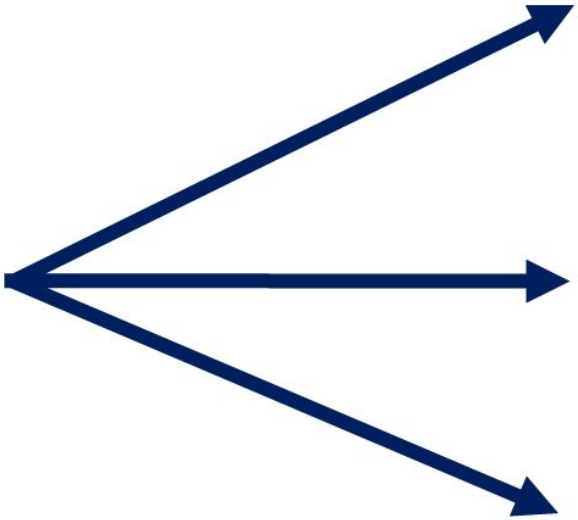
November 30, 2016

IEEE International Workshop on Information Forensics and Security,
NYU, Abu Dhabi, December 4-7, 2016.

Camera Model Identification

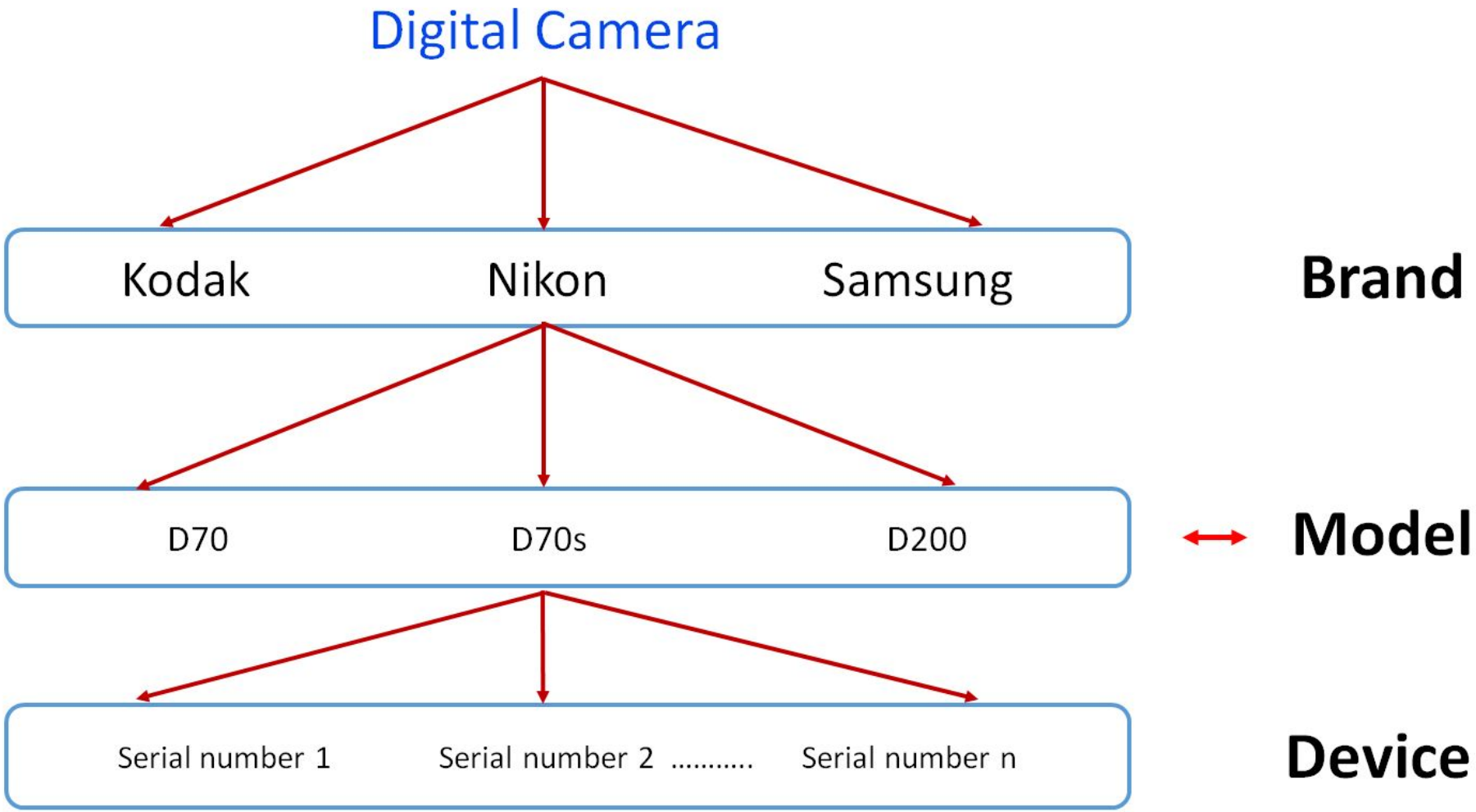
Image to test

Which model?



...

Brand / Model / Device



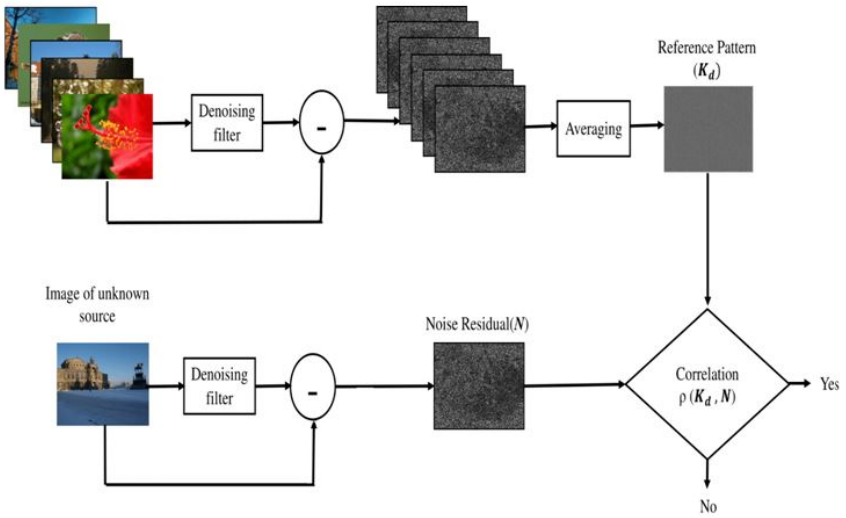
Deadlocks; most of them are still not addressed...

- Scalability in efficiency when nb of camera $\nearrow$,
- Treat the "unknown" class,
- Treat the mismatch phenomenon (generalization / overfitting),
- Take into account the variations in cameras setting,
- Identify even after image manipulations,

The different families

Methods based on a formal model:

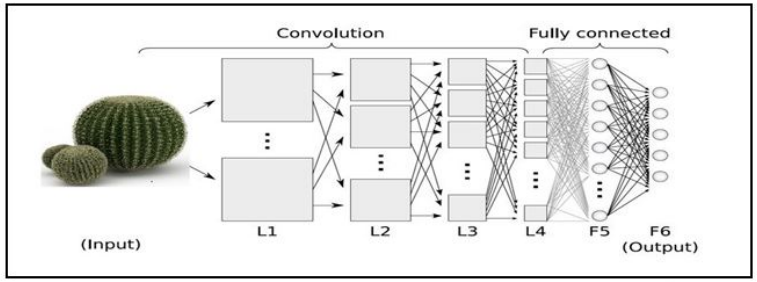
Example: based on PRNU computation:



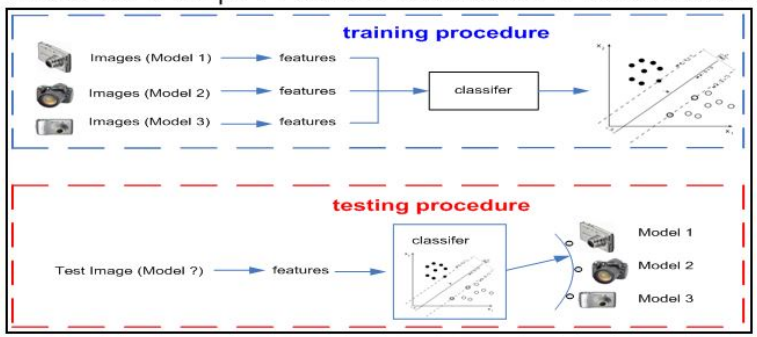
but also: based on radial distortion, sensor dust ...

Methods based on a machine learning:

Based on a CNN:



Based on 2 steps : Feature Extraction + Classification :



Recent works on **forgery** detection with CNNs:

- B. Bayar and M. C. Stamm, "A deep learning approach to universal image manipulation detection using a new convolutional layer," in ACM IH&MMSec'16. Vigo, Galicia, Spain, 2016.
- J. Chen, X. Kang, Y. Liu, and Z. Wang, "Median filtering forensics based on convolutional neural networks," IEEE Signal Processing Letters, vol. 22, no. 11, pp. 1849-1853, Nov 2015.

Work on camera model identification (... next year!):

- L. Bondi, L. Baroffio, P. Bestagini, E. Delp, and S. Tubaro, "A preliminary study on convolutional neural networks for camera model identification", in IS&T Electronic Imaging, Media Watermarking, Security, and Forensics, Jan **2017**.

Outline

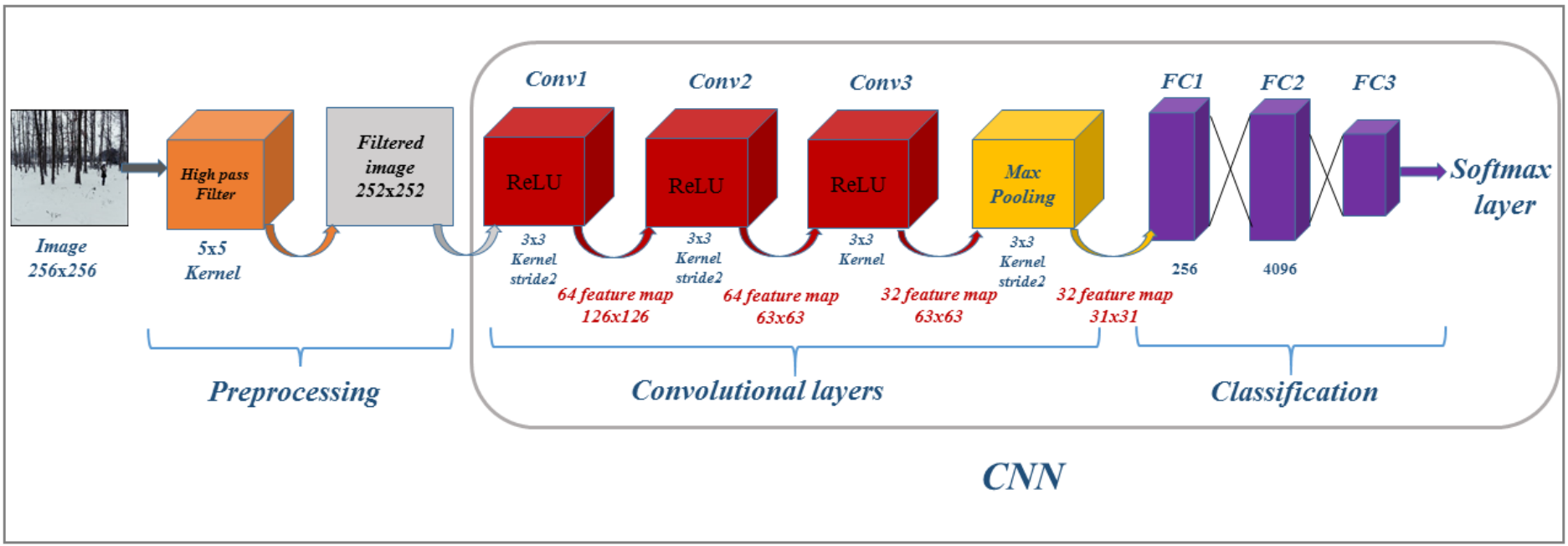
An experimental paper

What do we have look at in this paper?

- 2 classical CNNs + a designed CNNs,
- Efficiency in function of the number of models (scalability),
- Learning time (complexity),
- Potential of CNNs (possible improvement and comparison with another approach),

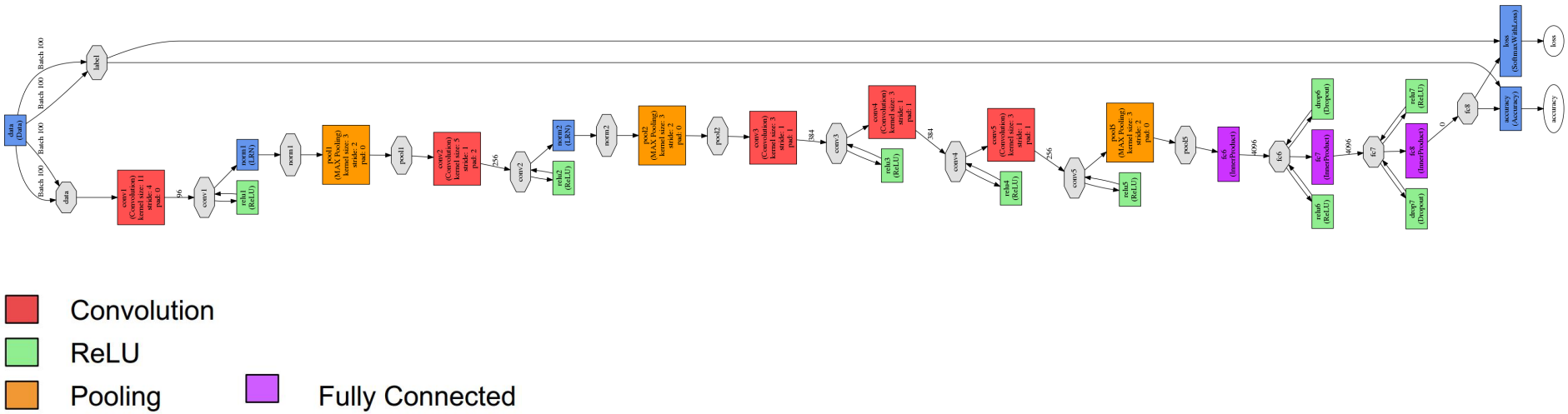
SmallNet

- A SmallNet, a small Net designed,
- 3 convolutional layers.



AlexNet

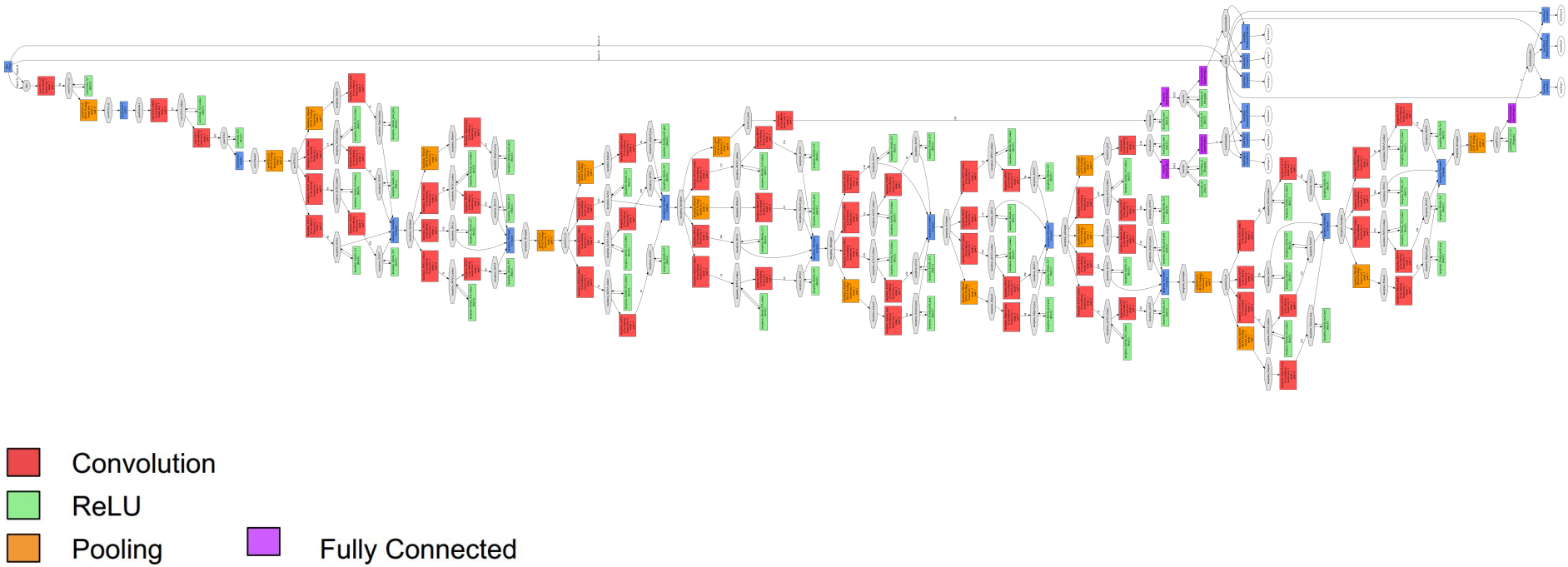
- Winner of ImageNet competition in 2012,
- 5 convolutions.



Alex Krizhevsky, Ilya Sutskever, and Georey E. Hinton. "Imagenet classication with deep convolutional neural networks," in Advances in Neural Information Processing Systems 25, pages 1097-1105. Curran Associates Inc., 2012.

GoogleNet

- Winner of ImageNet competition in 2015,
- 22 convolutions.



Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott E.Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. "Going deeper with convolutions," in IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Boston, USA, June 7-12, 2015.

Preliminaries observations:

- Two classical CNNs + a small one,
- Requires GPU(s):
  - In our case a Nvidia GeForce GTX Titan X ($\approx$ 1000-1500 dollars).
- Lots of libraries for CNNs on GPU ;
  - In our case DIGITS.
- Learning $\approx$ optimization of a function of million of parameters ;
  - In our case less than 2 days.

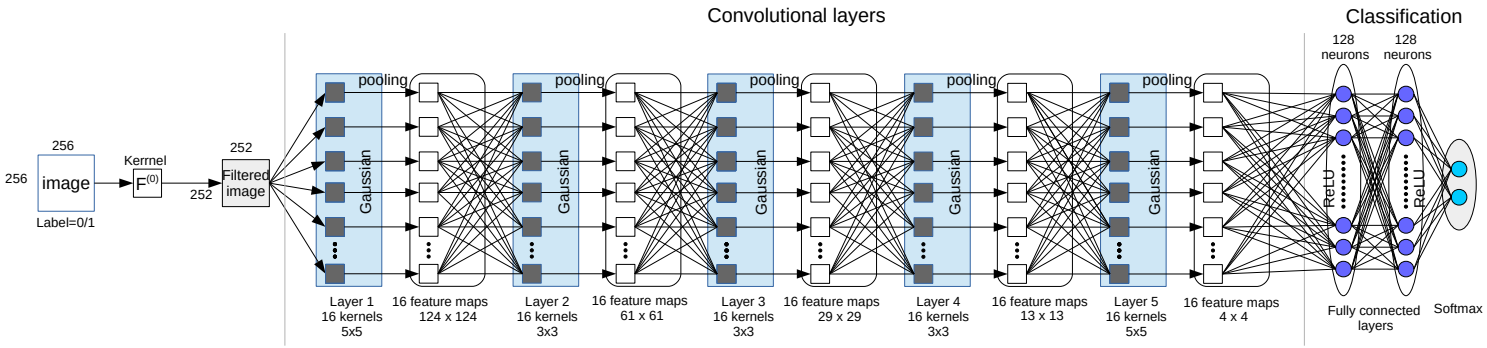
Basic question:

- Is it working?
- Complexity (time)?
- Efficiency compared to other approaches?

and then...

- Is it scalable?
- Can we do better?

Recalls on the main bricks: preliminary filter

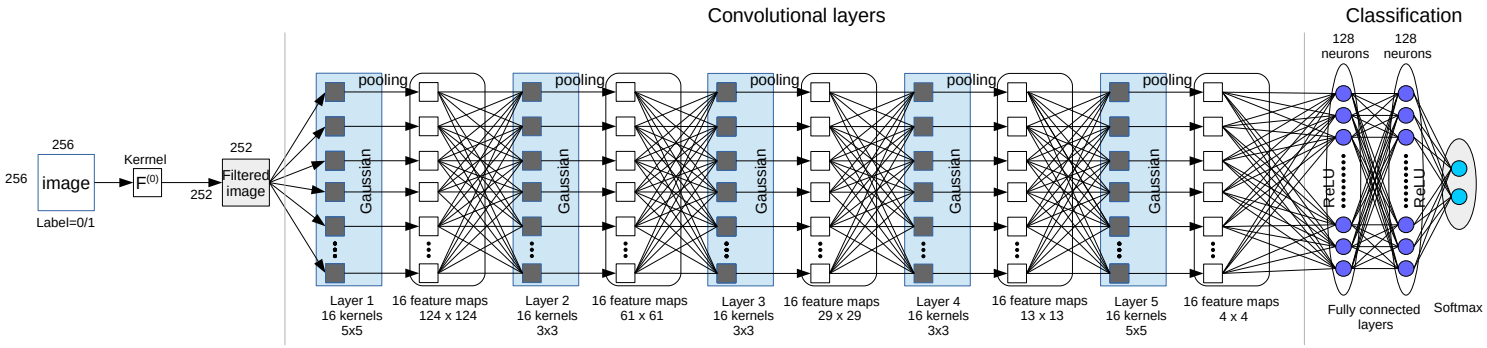


$$F^{(0)} = \frac{1}{12}\begin{pmatrix} -1 & 2 & -2 & 2 & -1 \\ 2 & -6 & 8 & -6 & 2 \\ -2 & 8 & -12 & 8 & -2 \\ 2 & -6 & 8 & -6 & 2 \\ -1 & 2 & -2 & 2 & -1 \end{pmatrix}$$

- Gives an orientation for the CNNs convergence, and suppress the interference caused by image content.
- We also tested a noise removing filter (used in the papers related to PRNU) but results were lower.

Yinlong Qian, Jing Dong, Wei Wang, and Tieniu Tan, "Deep Learning for Steganalysis via Convolutional Neural Networks," in Proceedings of SPIE Media Watermarking, Security, and Forensics 2015.

Recalls on the main bricks: the layers in a Convolution Neural Network



Inside one layer; successive steps:

- a convolution step,
- the application of an activation function,
- a pooling step,
- a normalization step.

Experimental protocol

Database

- 33 camera models:
  - 27 from Dresden database,
  - 6 from personal cameras,
- 1 model = only 1 device,
- Images are cropped to 256×256 based on a regular paving,
- 80% of images for the training ; 20% of images for the test,
- From 9 000 to 64 000 images (256×256) per model,

→ Results are averaged, after running the procedure 5 times, with 5 different split of the database.

Results of the 3 CNNs

	12 models	14 models	33 models
AlexNet (5 convolutions layers)	94.5 %	90.5%	83.5%
SmallNet (3 convolutions layers)	98.0 %	97.1%	91.9%
GoogleNet (27 convolutions layers)	99.0 %	98.0 %	94.5%

Table: Networks identification accuracy

- Best results for **GoogleNet**; SmallNet is not so bad,
- GoogleNet's results are **not so far from the state-of-the-art**;
  - Note: All the Networks only use a portion of the image (256×256) for the identification...
- GoogleNet **scales** well with the increase of cameras number,

Additional conclusions

- Bigger networks or networks better tuned $\rightarrow$ better results,
- Scalability is not so bad,
- Filter **pre-processing** allows better results (see paper) and probably a [easier/faster/better] convergence,
- GoogleNet is **3 times longer** 'to learn / to test' compared to SmallNet, but it takes only 16 hours for 12 cameras ($\approx$ 420 000 images),

What about other approaches?

As an example, for the 14 first camera models:

- PRNU accuracy = **97.5%** (images at **full resolution**),
- GoogleNet accuracy = **98%** (less than **1%** of the full resolution:
  - Portion of 256×256 from images $\in \{3000 \times 2000, ..., 4000 \times 3000\}$)
- 'Features + SVM [Amel *et al. 2016*]' accuracy = **98.75%** (images at **full resolution**),

So, why using CNNs?

A. Tuama , F. Comby, M. Chaumont, "Camera Model Identification Based Machine Learning Approach With High Order Statistics Features", in EUSIPCO'2016, 24th European Signal Processing Conference 2016, Budapest, Hungary, August 29 - September 2, 2016, pp 1183-1187.

Easy way to improve the efficiency

- Transfer learning (= pre-learning),
- Batch Normalization and better activation function,
- Virtual database augmentation,
- Use of an "unknown" class,
- Pool a set of vote (1 vote per portion of the image),
- Use a bigger Net (ResNet; 152 layers).

Conclusion

- Even a small Net (as SmallNet) can give good results,
- GoogleNet can give results close to the State-of-the-Art,
- CNNs is probably the best way to do camera model identification; just use a big Net and the new tricks.

Future: Continue exploring CNNs but look at deadlocks...:

- Scalability in efficiency when nb of cameras $\nearrow$,
- Treat the "unknown" class,
- Treat the mismatch phenomenon (generalization / overfitting),
- Take into account the variations in cameras setting,
- Identify even after image manipulations.