Master in Artificial Intelligence and Robotics
Information and Automation Engineering, University Sapienza of Rome
Academic Year 2020/2021

Implementation and Comparison of Camera Model Identification Methods

Paper References:
First Steps Toward Camera Model Identification with Convolutional Neural Networks (L. Bondi, et al.)

The Forchheim Image Database for Camera Identification in the Wild (B. Hadwiger, C. Riess)

Riccardo Riglietti, Yannick Moell, Ascanio Santomarco

SAPIENZA
UNIVERSITÀ DI ROMA

The Task – Camera Model Identification

- Identify camera model using given shot of a picture or even a single patch of the picture.
- Intuition: Each camera with **individual acquisition process** (light focus, color channel interpolation, different brightness)
- Important for **image forensics** (forgery detection, copyright tracking etc.)

Sapienza
Università di Roma

Previous Approaches

- Search for **specific traces** in images
→ e.g. **noise** characteristics [1], **lens distortion** [2], **gain** histograms (= electronic signal amplification) [3]
- Capture statistical image properties like local binary patterns [4]

→ Methods **require a priori knowledge** about the models, need of **manually defined procedures**

References for previous approaches:
[1] T. Filler et al.: *Using sensor pattern noise for camera model identification*
[2] K. Choi et al.: ***Automatic source camera identification using the intrinsic lens radial distortion***
[3] S.-H. Chen and C.-T. Hsu: ***Source camera identification based on camera gain histogram***
[4] G. Xu and Y. Q. Shi: ***Camera model identification using local binary patterns***

Pros and Cons of the Proposed Methods

- CNN+SVM approach (aka “*BondiNet*”)
  - **Two-step fashion** (feature extractor, classifier)
  - **Data-driven** (no analytical modeling needed, **low-dimensional feature vectors**), big dataset needed
  - **Generalisation ability** to new camera models
- EfficientNet
  - **Pretrained network** on ImageNet, mainly for image recognition
  - **Finetuning** and transfer to our task, i.e. adding dense layers for our final classification
  - Good way of weights initialization, **fast training**

SAPIENZA
UNIVERSITÀ DI ROMA

Datasets – Forchheim Image Database

- **3851 images** from **27 smartphones** (**143 scenes** for each)
- Original images, compressions of social media platforms
- Each image taken from (almost) the same position with all models

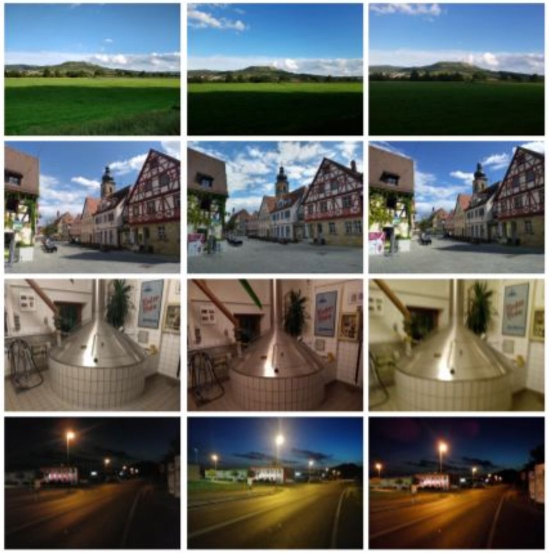


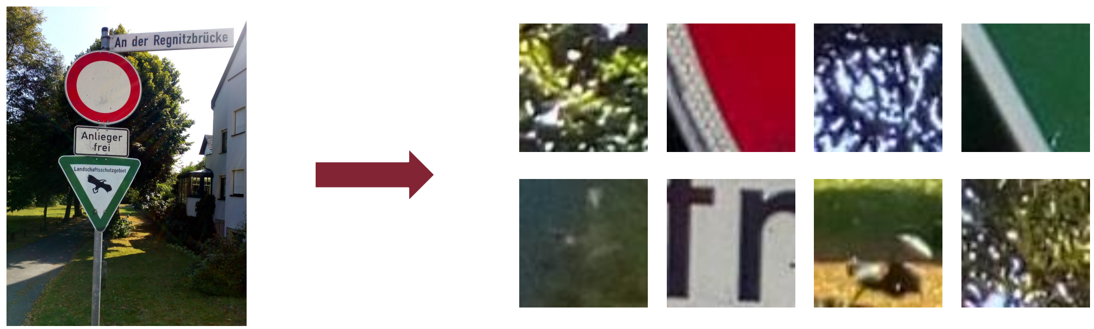
TABLE I
MAIN FEATURES OF SMARTPHONES IN FODB

ID	Brand	Model	OS	Version	Date
01	Motorola	E3	Android	6.0	09/2016
02	LG	Optimus L50	Android	4.4.2	06/2010
03	Wiko	Lenny 2	Android	5.1	09/2014
04	LG	G3	Android	5.0	07/2014
05	Apple	iPhone 6s	iOS	13.6	09/2015
06	LG	G6	Android	9	05/2017
07	Motorola	Z2 Play	Android	8.0.0	08/2017
08	Motorola	G8 Plus	Android	9	10/2019
09	Samsung	Galaxy S4 mini	Android	4.4.4	05/2013
10	Samsung	Galaxy J1	Android	4.4.4	01/2015
11	Samsung	Galaxy J3	Android	5.1.1	01/2016
12	Samsung	Galaxy Star 5280	Android	4.1.2	05/2013
13	Sony	Xperia E5	Android	6.0	11/2016
14	Apple	iPhone 3	iOS	7.1.2	06/2008
15	Samsung	Galaxy A6	Android	10	05/2018
16	Samsung	Galaxy A6	Android	10	05/2018
17	Apple	iPhone 7	iOS	12.3.1	09/2016
18	Samsung	Galaxy S4	Android	6.0.1	04/2013
19	Apple	iPhone 8 Plus	iOS	13.2	09/2017
20	Google	Pixel 3	Android	9	11/2018
21	Google	Nexus 5	Android	8.1.0	10/2015
22	BQ	Aquaris X	Android	8.1.0	05/2017
23	Huawei	P9 lite	Android	6.0	05/2016
24	Huawei	P8 lite	Android	5.0	04/2015
25	Huawei	P9 lite	Android	7.0	05/2016
26	Huawei	P20 lite	Android	8.0.0	04/2018
27	Google	Pixel XL	Android	10	10/2016

Images: B. Hadwiger, C. Riess: ***The Forchheim Image Database for Camera Identification in the Wild*** (https://arxiv.org/abs/2011.02241)

Data Preprocessing (CNN+SVM)

- **Patch creation** (64x64, gridwise), inherit same label from source
- **80 patches** per image



- **Compare patch dynamics** with total image dynamics (mean intensity) → exclude too dark/saturated patches (for CNN+SVM)
- Dataloader, **normalizing technique** (subtract mean image, normalize via factor 0.0125)
- **Train/val/test split** into 0.7/0.1/0.2
- Only 18 classes used due to **quadratic behaviour** of OneVsOne SVM

Structure of *BondiNet*

- The attribution of a picture to a specific camera model is done in a **blind fashion**, by using camera **noiseprints**

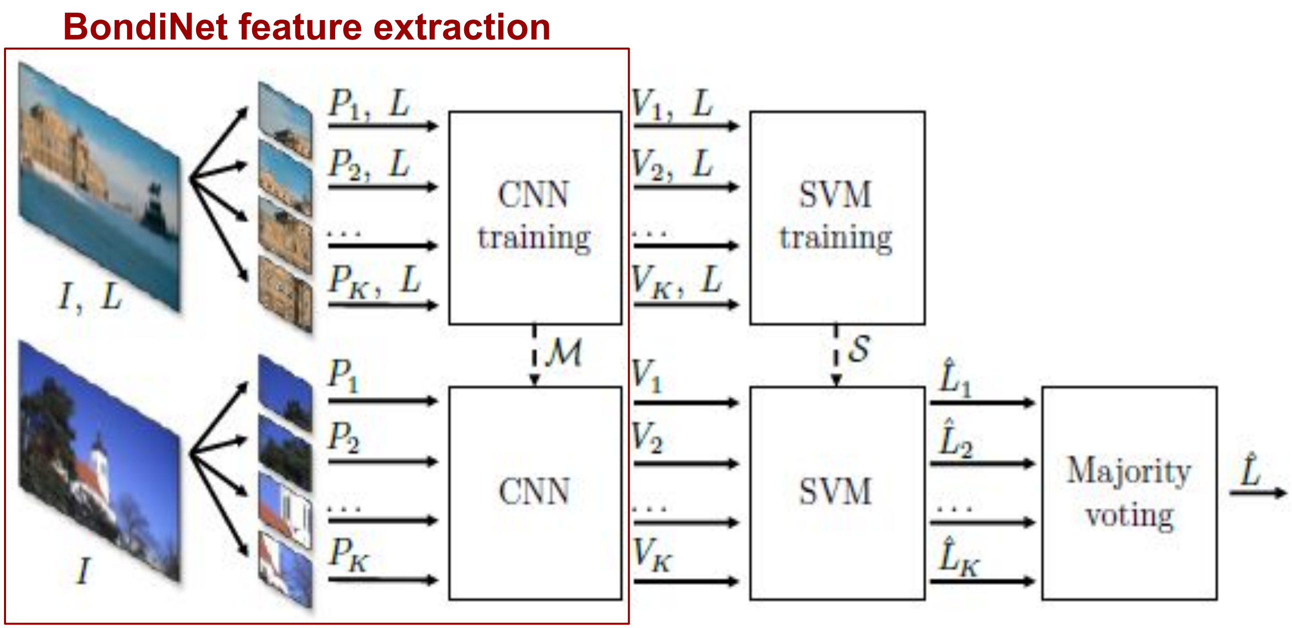
- The overall architecture of the **CNN** is characterized by **255,000** parameters.
- Reduced network size due to smaller dataset (avoid overfitting): the orginal number of parameters was **340,462** in the paper

- These parameters are learned through **SGD** on batches of 128 patches

Model Summary of *BondiNet*

```
Model: "model_9"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
inputs (InputLayer)          [(None, 64, 64, 3)]       0
_________________________________________________________________
conv2d_32 (Conv2D)           (None, 61, 61, 32)        1568
_________________________________________________________________
batch_normalization_32 (Batc (None, 61, 61, 32)        128
_________________________________________________________________
max_pooling2d_24 (MaxPooling (None, 30, 30, 32)        0
_________________________________________________________________
conv2d_33 (Conv2D)           (None, 26, 26, 48)        38448
_________________________________________________________________
batch_normalization_33 (Batc (None, 26, 26, 48)        192
_________________________________________________________________
max_pooling2d_25 (MaxPooling (None, 13, 13, 48)        0
_________________________________________________________________
conv2d_34 (Conv2D)           (None, 13, 13, 64)        76864
_________________________________________________________________
batch_normalization_34 (Batc (None, 13, 13, 64)        256
_________________________________________________________________
max_pooling2d_26 (MaxPooling (None, 6, 6, 64)          0
_________________________________________________________________
conv2d_35 (Conv2D)           (None, 2, 2, 64)          102464
_________________________________________________________________
batch_normalization_35 (Batc (None, 2, 2, 64)          256
_________________________________________________________________
flatten_8 (Flatten)          (None, 256)               0
_________________________________________________________________
dense_18 (Dense)             (None, 128)               32896
_________________________________________________________________
dense_19 (Dense)             (None, 18)                2322
=================================================================
Total params: 255,394
Trainable params: 254,978
Non-trainable params: 416
_________________________________________________________________
```

Structure of *BondiNet*



- The CNN extracts a 128 dimensional feature vector for each patch
- The feature vectors are fed to N*(N - 1) / 2 linear SVM classifiers
- Validation dataset for the prediction
- The model with smallest loss on validation patches is chosen

Structure of EfficientNet

- **Pretrained convolutional network**, ImageNet for image recognition
- Added EfficientNetB1, one the “smaller” built-in EfficientNets with **7.8M** parameters to reduce overfitting.
- Adding two dense layers and dropout for camera model classification
  - Dense layer: **64** neurons
  - **Dropout** with rate of **0.05**
  - Dense layer: **18** neurons (number of classes)

Image: Mingxing Tan, Quoc V. Le: ***EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks*** (https://arxiv.org/abs/1905.11946)

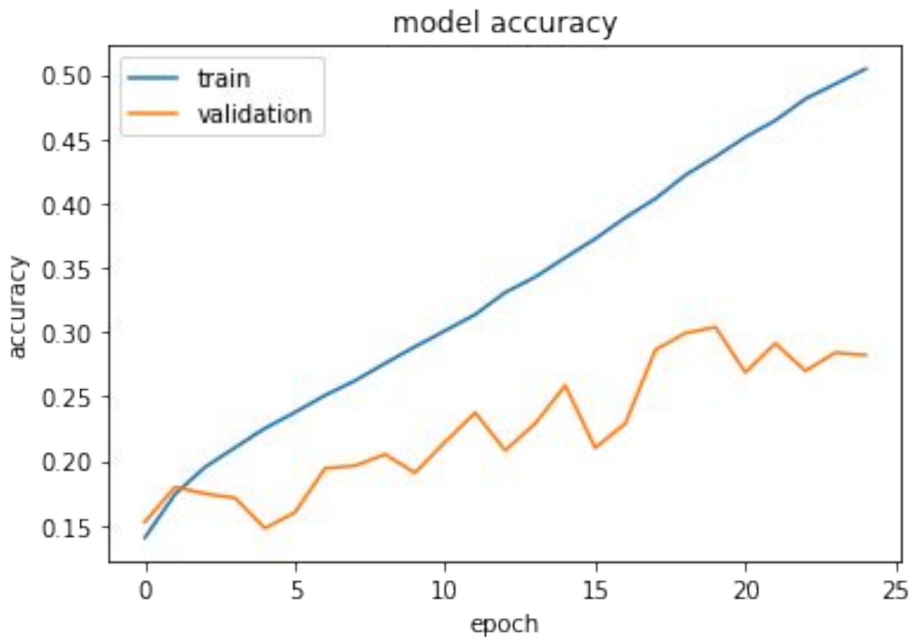
Comparison of Validation Accuracies

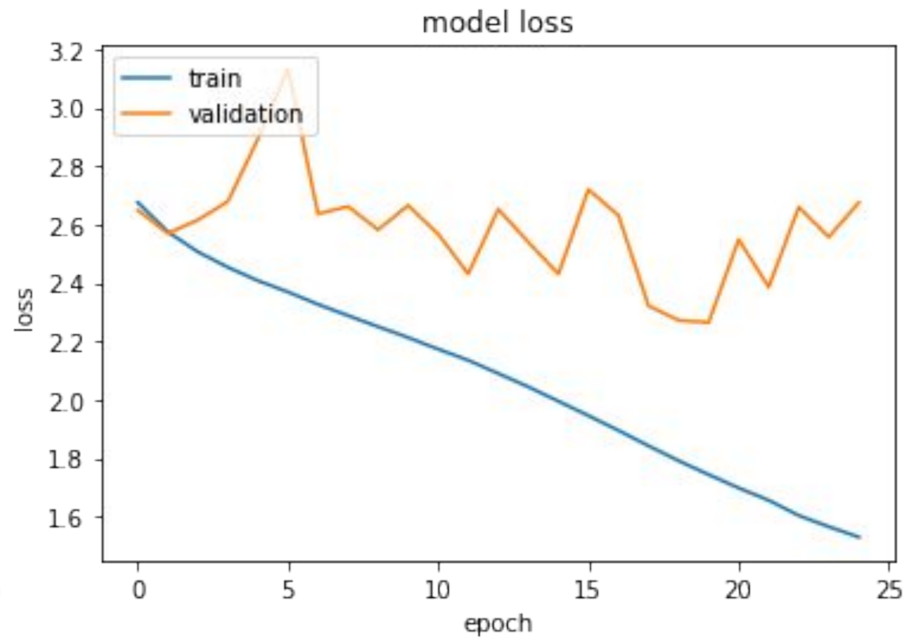
- Dataset itself is totally balanced, focus on accuracy (F1 not needed in our case)

- **BondiNet** as stand-alone: **30.4%** (with 43.6% train acc)
- **BondiNet** + **OneVsRest**: **35.1%**
- **BondiNet** + **OneVsOne**: **36.7%**
- **EfficientNet**: **67.9%** (with 80.6% train acc)
  (last epoch: 96.6% in train with 70.8% for val)

- Keep in mind:
  Random guess is 100%/18 classes = **5.56% per class**

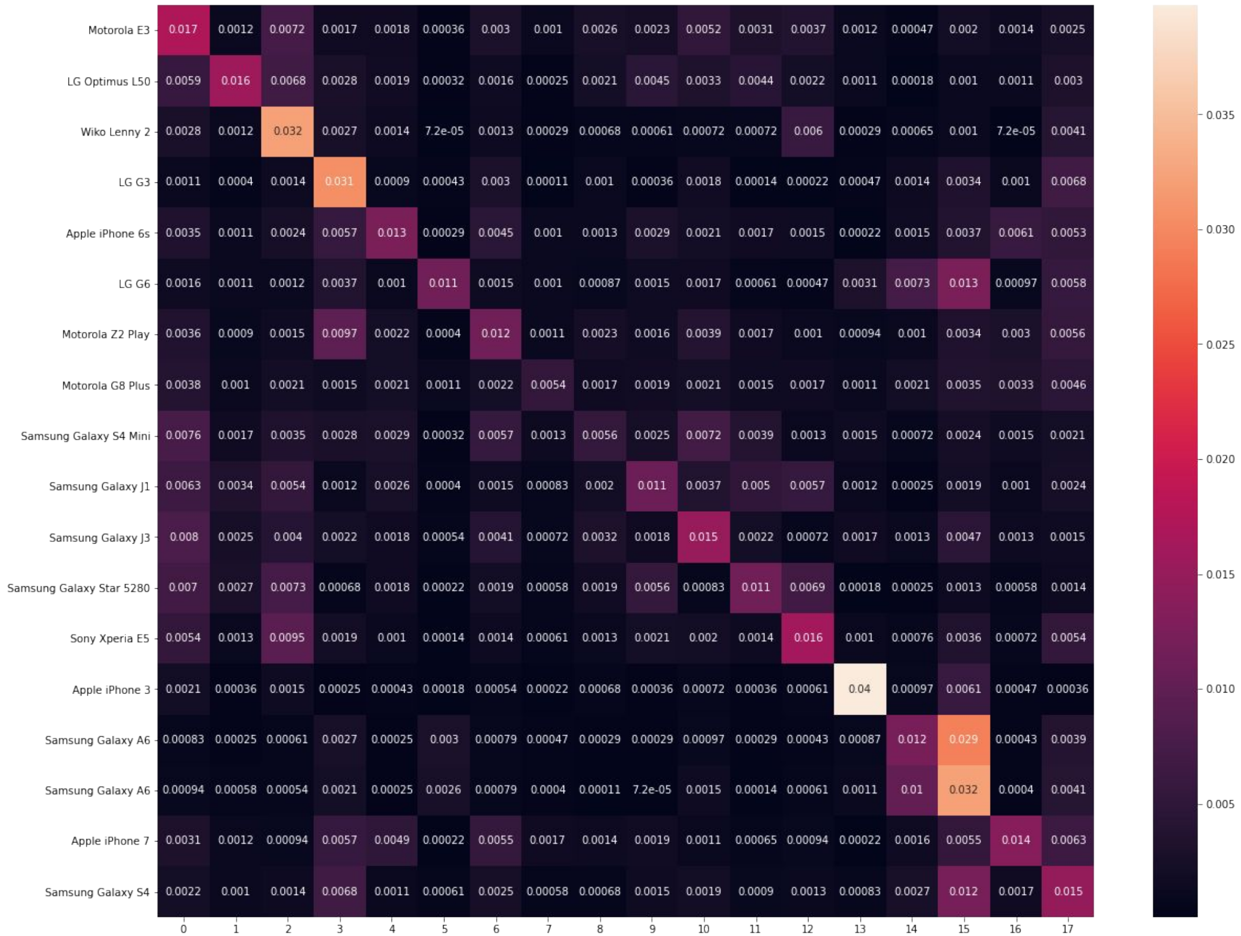
Results – BondiNet (1)



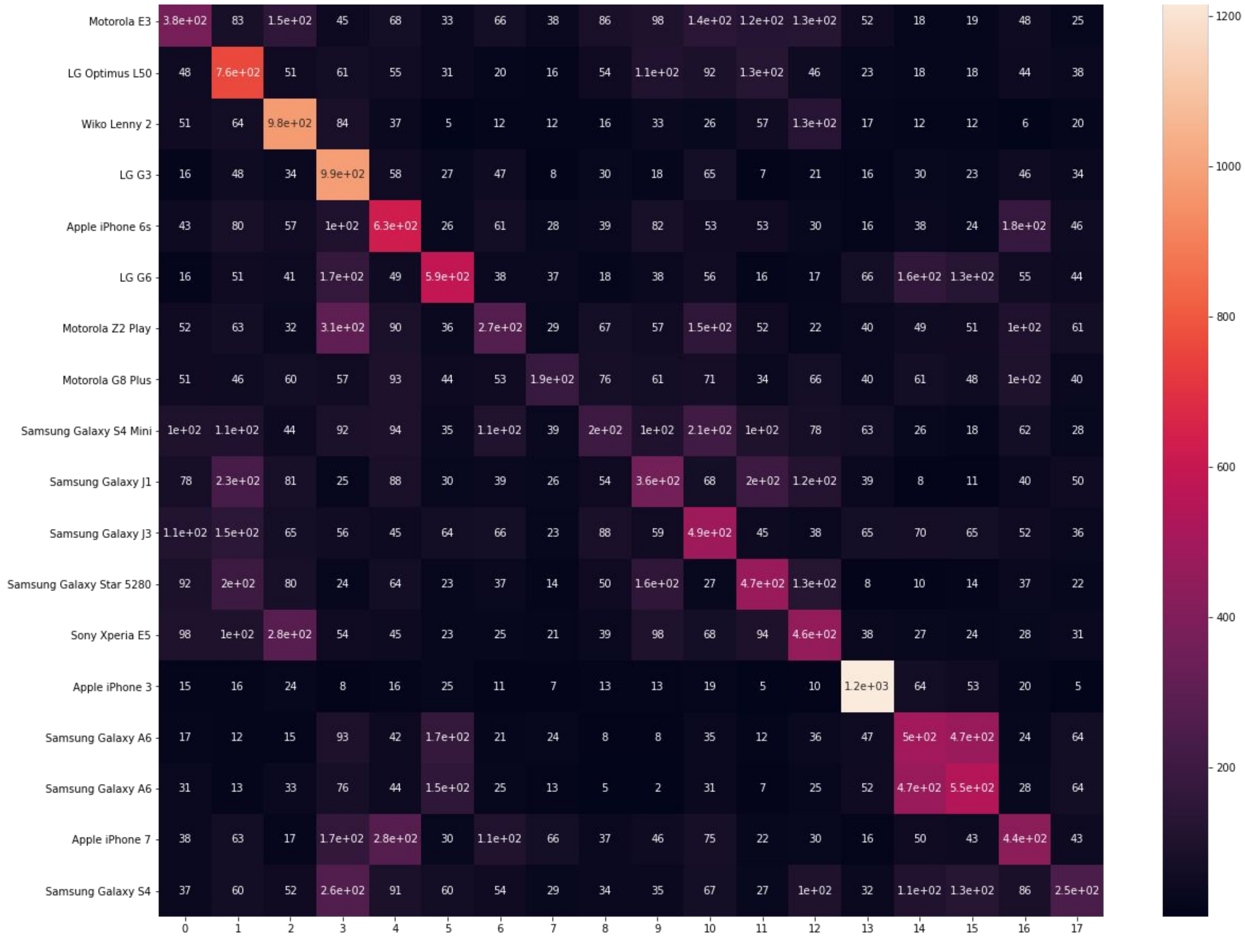


SAPIENZA
UNIVERSITÀ DI ROMA

Results – BondiNet (2)

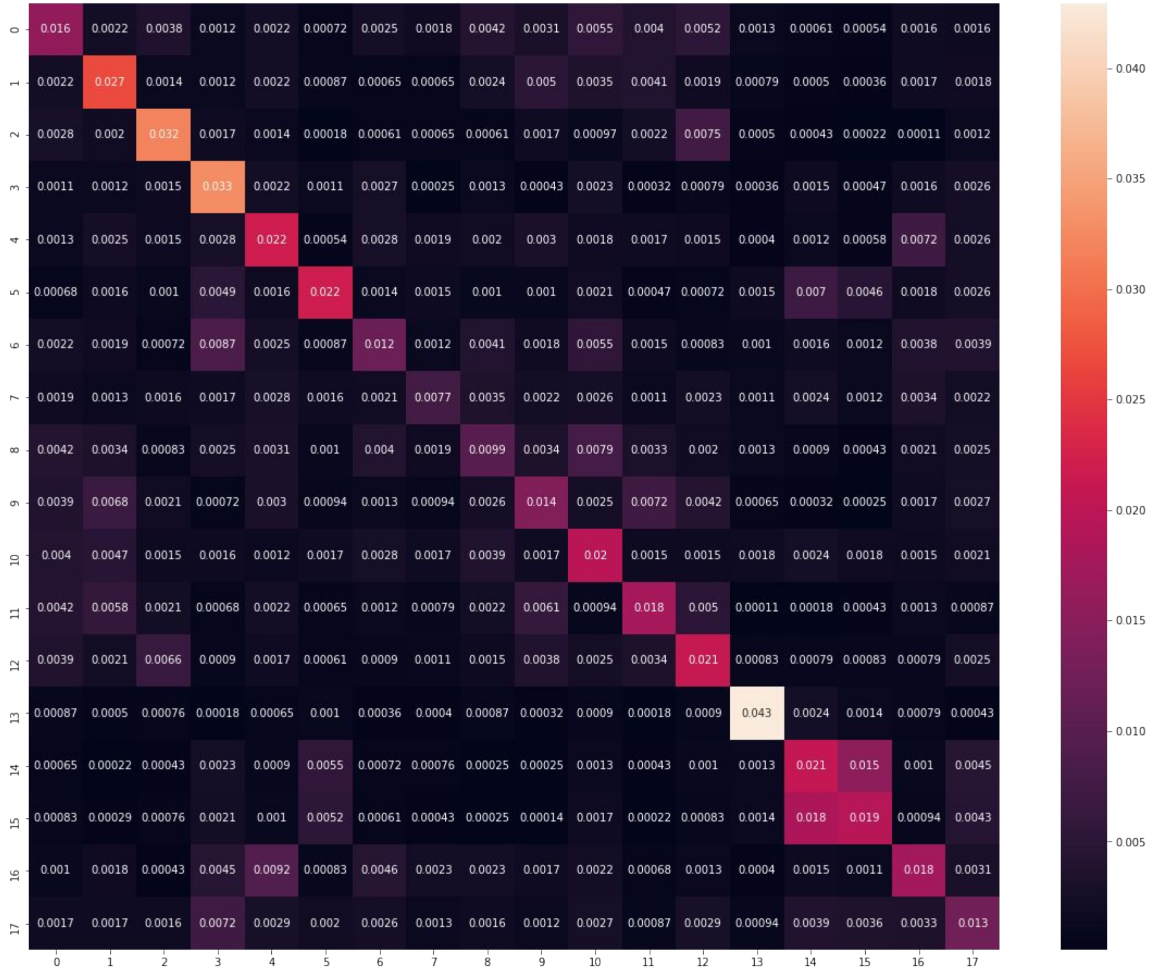


Results – BondiNet + SVM OneVsRest

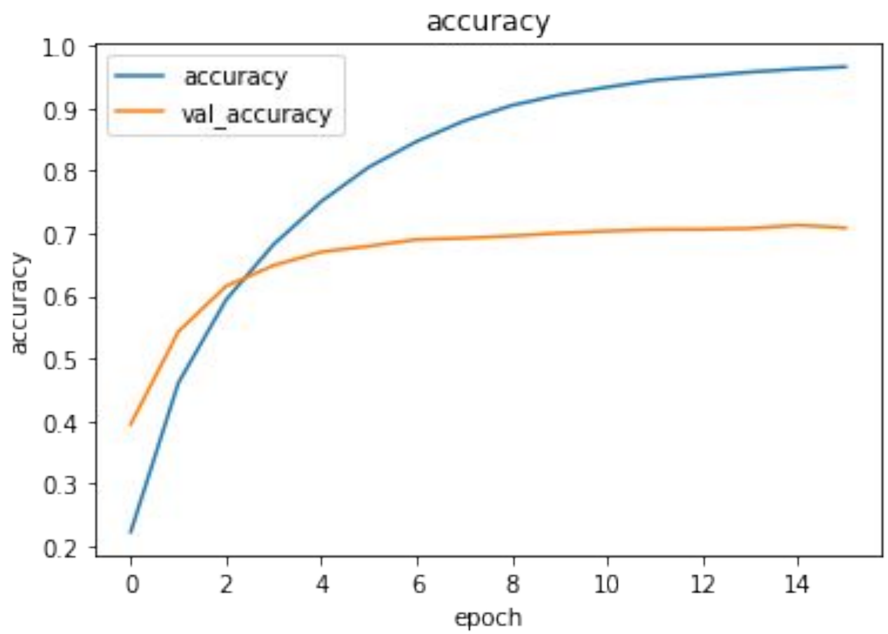


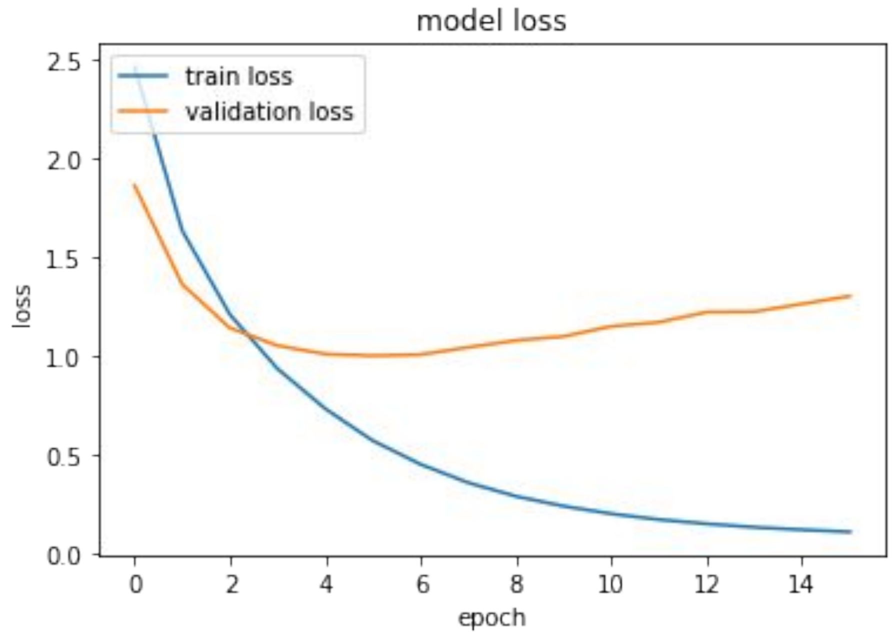
SAPIENZA
UNIVERSITÀ DI ROMA

Results – BondiNet + SVM OneVsOne



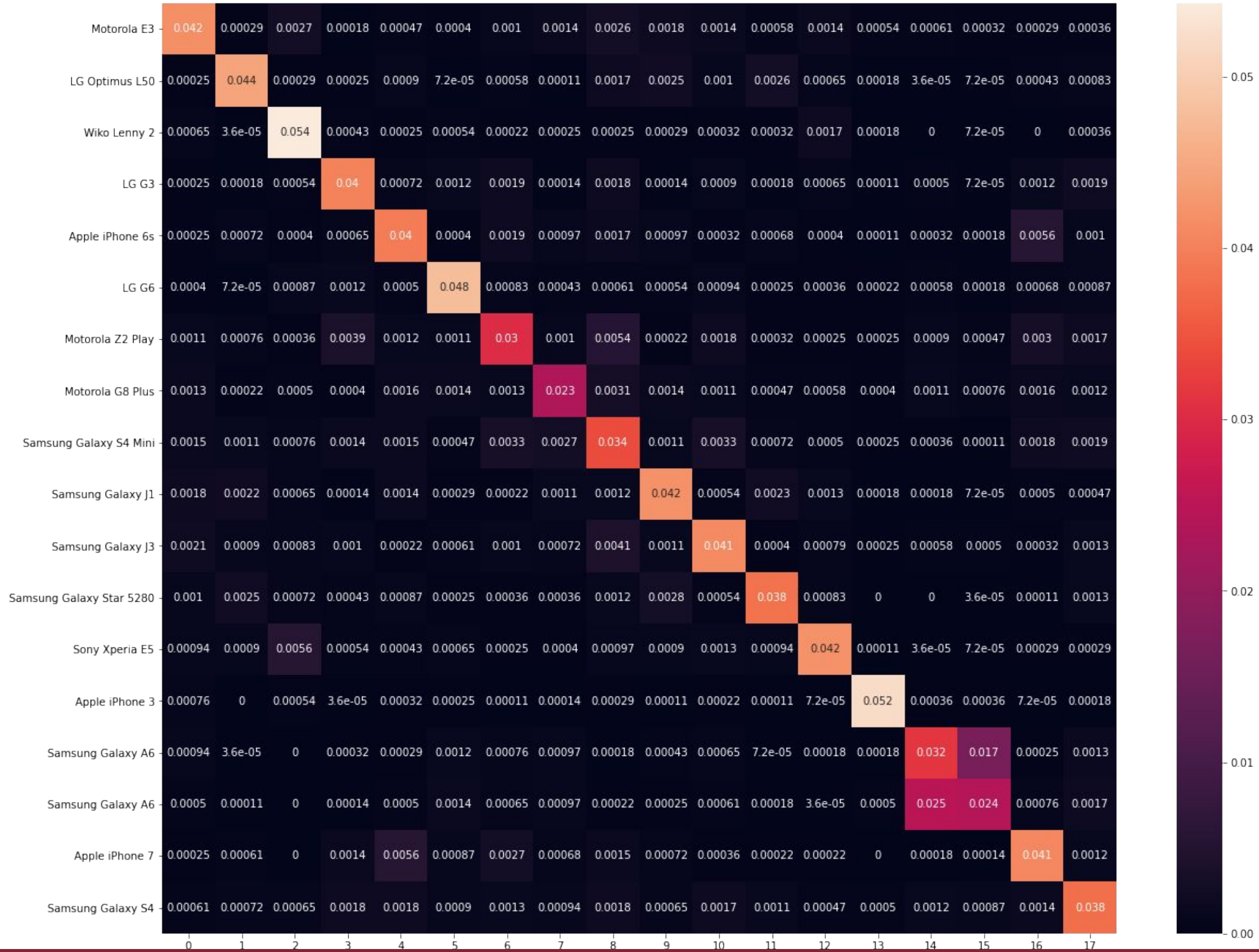
Results – EfficientNet (1)





Sapienza Università di Roma

Results – EfficientNet (2)



SAPIENZA
UNIVERSITÀ DI ROMA

Final Comments on Our Results

- Best result with **finetuned EfficientNet**, avoided overfitting via dropout layer and early stopping
- BondiNet itself poor, in combination with SVM variants **only slight improvement**

- **Important: task is hard**, differentiation only based on slight patterns and camera distortions, we only use one patch to classify
- **But remember (again):** Random chance is **5.56%**

- Dataset smaller and (probably) harder than original one: **Smartphone cameras more similar** to each other than digital cameras → *Possible further comparison in future*
- **One class doubled** in the dataset (Samsung Galaxy A6), leads to unwanted confusion

Master in Artificial Intelligence and Robotics
Information and Automation Engineering, University Sapienza of Rome
Academic Year 2020/2021

Implementation and Comparison of Camera Model Identification Methods

Riccardo Riglietti, Yannick Moell, Ascanio Santomarco

SAPIENZA
UNIVERSITÀ DI ROMA