Camera Model Identification with a Data-Driven Model

Juan Calderón
juan2220049@correo.uis.edu.co

Dana Villamizar
dana2220081@correo.uis.edu.co

Brayan Sanchez
brayan2220083@correo.uis.edu.co

Universidad Industrial de Santander

Abstract

Determining which camera captured a particular photograph has a significant impact in the field of forensic analysis, especially given the ease with which digital images can now be copied, modified, and redistributed. To address this challenge, the forensics community has developed a variety of techniques to analyze multimedia provenance, often by exploiting distinctive traces left by the camera during the image acquisition pipeline. Through reverse engineering of these traces, traditional approaches aim to attribute an image to its source device. In this paper, we propose a data-driven alternative based on deep convolutional neural networks, which automatically learn discriminative features directly from image data to perform source camera identification. A repository of this work is available at: https://github.com/JJCG25/An-Image-Processing-Project-Camera-Model-Identification-with-a-Data-Driven-Model .

1. Introduction

As a consequence of the increasing use of smartphones equipped with high-resolution cameras, along with the growing popularity of social media platforms, digital photography has become an essential part of everyday life. However, the ease with which digital images can be copied, modified, and redistributed has raised significant concerns regarding their authenticity and origin. The need for effective tools to attribute an image to its original capture device has become increasingly relevant, particularly in applications such as intellectual property protection, digital forensics, legal proceedings and online content moderation. In this context, the forensics community has developed a series of tools to analyze the history of multimedia objects [1].

Among the challenges addressed by the multimedia forensics community, source camera identification stands out as a key problem [2]. Determining which device captured an image can play a critical role in legal investigations or in attributing authorship of sensitive or illicit content. Traditionally, this problem has been approached using model-based methods that aim to detect specific traces introduced by the image acquisition pipeline, such as color filter array (CFA) interpolation patterns [3, 4], gain histogram features [5], lens distortion [6], or traces of the white balance algorithm [7]. In particular, Photo-Response Non-Uniformity (PRNU) noise has been widely studied, as it enables differentiation even between different instances of the same camera model due to the unique imperfections of each sensor [8–10].

Despite their accuracy, these model-driven methods require careful feature engineering and may underperform in uncontrolled settings or with previously unseen devices. Deep learning has established itself as a powerful approach in image analysis due to its ability to learn discriminative features directly from data. Convolutional Neural Networks (CNNs), in particular, have demonstrated remarkable results in tasks such as image classification, tampering detection, and device attribution [11–13], which is why we adopt them in this work. By stacking multiple convolutional layers, CNNs build a hierarchical representation of the input, capturing both low-level textures and high-level patterns—characteristics especially important for identifying subtle device-specific traces in images.

In this work, we propose to adapt and fine-tune a CNN-based architecture—originally designed for general digital camera identification—to the specific context of mobile phones. To achieve this, we employ a transfer learning strategy: starting from a model pretrained on the Dresden Image Database [14], we partially retrain it on a new dataset named Floreview [15], composed exclusively of images captured with smartphones and organized by device model.

The main contributions of this work are: (i) evaluating the generalization capability of the pretrained model when adapted to the mobile domain, and (ii) building a closed-set

classification system capable of assigning images to known smartphone models and their individual instances.

2. Method

The proposed methodology follows a two-stage approach. Initially, we aimed to replicate the work presented by Baroffio *et al.* [16], which introduces a convolutional neural network (CNN) architecture specifically designed for camera model identification. Their network receives as input a small RGB patch of size 48×48 pixels and comprises a series of layers configured to extract sensor-specific artifacts introduced during the image acquisition pipeline. The architecture begins with a convolutional layer consisting of 128 filters of size $7 \times 7 \times 3$ and stride 1, followed by a ReLU activation and a max-pooling layer with 3×3 windows and stride 2. This pattern is repeated with a second convolutional layer ($7 \times 7 \times 128$ filters, 512 channels), ReLU activation, and max-pooling. A third convolutional block with 2048 filters of size 6×6 is followed by a fully connected layer (2048 neurons) with a dropout rate of 0.5, another ReLU activation, and a final fully connected layer with a number of neurons equal to the number of classes. A SoftMax layer outputs the class probabilities, and the model is trained to minimize the multinomial logistic loss. The full structure of this architecture is detailed in Table 1.

The original authors do not report the training hyperparameters, such as optimizer type or learning rate. Consequently, in our implementation, we employed stochastic gradient descent (SGD) with a learning rate of 0.001 and momentum set to 0.9, which are commonly used settings in comparable tasks. The network was trained and evaluated using the Dresden Image Database [14], which contains 16,982 natural images captured by 74 devices from 27 distinct camera models. Following the experimental setup in [16], we extracted 48×48 patches from the images to train the model.

After successfully training and evaluating the CNN on the Dresden dataset, we selected the best-performing model for fine-tuning on a different dataset, named *FloreView*. This dataset, introduced by Baracchi *et al.* [15], comprises over 9,000 media samples—specifically, 6,637 natural and flat images and 1,831 videos—captured by 46 smartphones from 11 major brands under tightly controlled acquisition conditions. For each smartphone, at least 100 natural images and 30 videos were collected in consistent lighting conditions and from the same urban locations, aiming to minimize bias induced by content variation across devices. Despite FloreView including video samples, we restricted our analysis to still images due to the image-based nature of the classification task addressed in this work.

To improve classification performance, we transitioned to a more robust transfer learning approach using EfficientNetV2-M [17], a modern convolutional neural network architecture known for its balance between accuracy and training efficiency. We used the implementation available in TorchVision, initializing the network with pretrained weights on ImageNet [18].

Table 1. Characteristics and hiperparameters of the ConvNet

#	**Layer type**	**Parameter**	**Value**
1	Convolution	Kernel size # of filters Stride	7×7×3 128 1
2	ReLU	-	-
3	Max pooling	Kernel size Stride	3×3 2
4	Convolution	Kernel size # of filters Stride	7×7×128 512 1
5	ReLU	-	-
6	Max pooling	Kernel size Stride	3×3 2
7	Convolution	Kernel size # of filters Stride	6×6×512 2048 1
8	ReLU	-	-
9	Fully connected	Kernel size # of combinations Dropout rate	1×1×2048 2048 0.5
10	ReLU	-	-
11	Fully connected	Kernel size # of combinations	1×1×2048 # output classes
12	SoftMax	-	-
13	LogLoss	-	-

In our design, we removed the original classification head of EfficientNetV2-M and appended a custom classifier composed of three fully connected layers: $1280 \rightarrow 512$, $512 \rightarrow 256$, and $256 \rightarrow 39$, where 39 represents the number of smartphone classes in the FloreView dataset. Each linear layer was followed by a ReLU activation function. Importantly, in contrast to common transfer learning strategies, we did not freeze any layers of the EfficientNet backbone—allowing the entire model to be fine-tuned on our domain-specific task. The full architecture used in our work is depicted in Fig. 1.

The model was trained using the Adam optimizer with a learning rate of 1×10^{-3} and the cross-entropy loss function. Training was performed for up to 200 epochs with batch size 24, using the wandb library for experiment tracking. Metrics such as accuracy and gradient norms were logged at every step. The final model achieved a classification accuracy of 82% on the FloreView dataset, demonstrating strong generalization capabilities to smartphone-captured images across multiple brands.

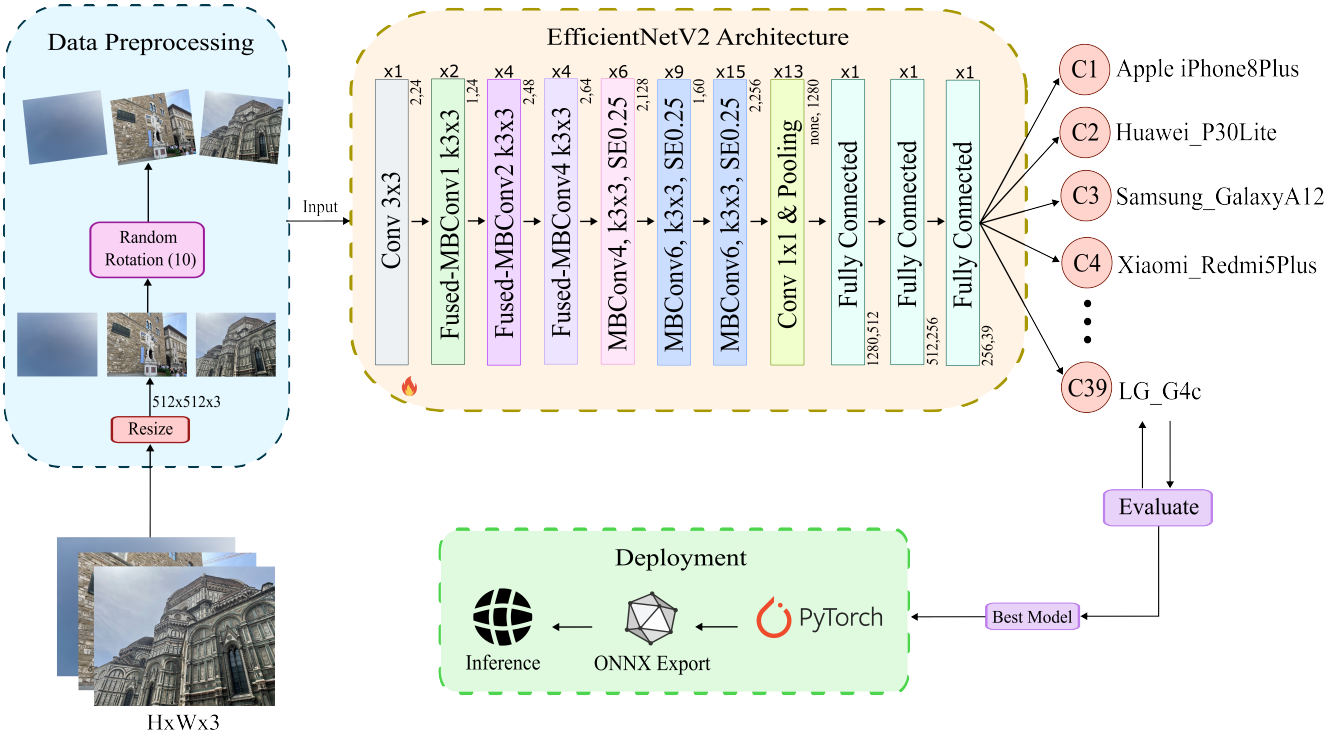

Figure 1. Modified EfficientNetV2-M architecture used in our approach. The original classification head was replaced by three fully connected layers (1280, 512, 256), ending in a 39-class output layer for smartphone brand classification. All layers, including the convolutional backbone, were fine-tuned.

To facilitate practical use of the trained model, we deployed it as a web application that allows users to upload an image and receive an estimated smartphone brand prediction. A web deployment of the model is available on: https://jjcg25.github.io/

3. Results

This section presents the results obtained from the approaches evaluated in this study: (i) training the original CameraConvNet architecture from scratch on the Dresden dataset; (ii) fine-tuning the pretrained CameraConvNet model on the smartphone-focused FloreView dataset; and (iii) full fine-tuning of an EfficientNetV2-M model on FloreView.

The first experiment aimed to reproduce the results reported in [16], where an accuracy of 0.941 was originally achieved using a CNN designed specifically for camera model identification. Following the same architectural specifications, the model was trained from scratch on the Dresden dataset using SGD with a learning rate of 0.001, momentum 0.9, and batch size 128. The final accuracy obtained was **81.89%**, as shown in Table 2. Although lower than the originally reported value, this result confirms that the architecture captures model-specific features effectively within the Dresden setting.

In the second experiment, a fine-tuning procedure was applied to the previously trained CameraConvNet model using the FloreView dataset. The aim was to adapt the model to the smartphone domain. However, this approach yielded unsatisfactory results. As illustrated in Figure 2, the validation accuracy peaked at only **30.2%**, with significant instability across training epochs. This suggests that the pretrained filters from the Dresden dataset were not sufficiently transferable to the domain of smartphone-captured images.

makecell

graphicx

Table 2. Validation accuracy and loss for each experimental setting.

Model	Dataset	Val. Accuracy	Val. Loss
CameraConvNet (from scratch)	Dresden	81.89%	0.59
CameraConvNet (fine-tuned)	FloreView	30.2%	1.73
EfficientNetV2-M (fine-tuned)	FloreView	**82.0%**	**1.03**

In light of these results, a third experiment was conducted using a more modern architecture. The EfficientNetV2-M model, pretrained on ImageNet, was adapted for classification on FloreView by replacing its final classification head with a custom block of three fully connected layers ($1280 \rightarrow 512 \rightarrow 256 \rightarrow 39$). Unlike con-

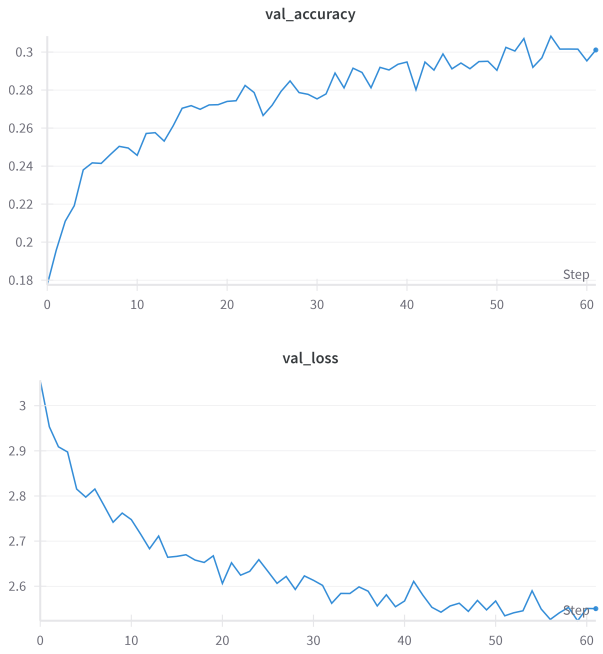

Figure 2. Validation accuracy (left) and precision (right) for the fine-tuned CameraConvNet on FloreView. The performance remains low and unstable across epochs.

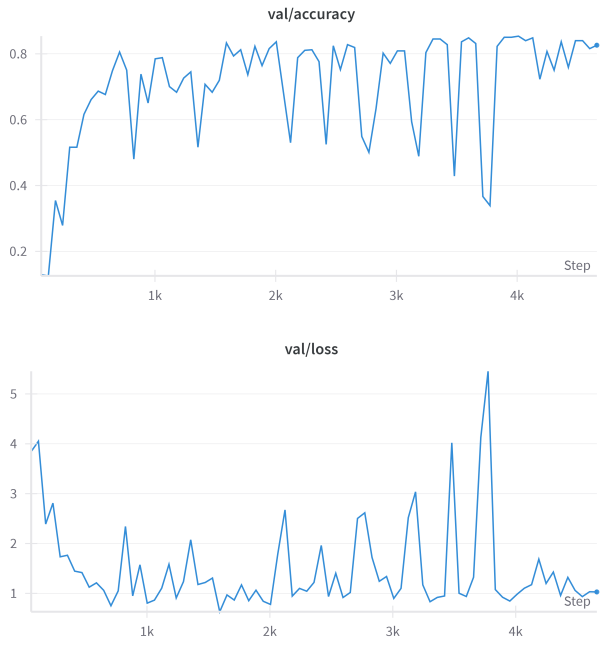

Figure 3. Validation accuracy (left) and precision (right) for the fine-tuned CameraConvNet on FloreView. The performance remains low and unstable across epochs.

ventional transfer learning strategies, no layers were frozen; instead, the entire network was fine-tuned end-to-end on the target dataset. Training was performed for 200 epochs using the AdamW optimizer and a batch size of 24. As illustrated in Figure 3, this approach achieved a final validation accuracy of **82%**, surpassing the previous fine-tuning attempt by a wide margin and matching the performance obtained by the original CameraConvNet on the Dresden dataset. The training process remained stable, with validation metrics showing consistent improvement across epochs.

These results demonstrate that while the CameraConvNet architecture remains effective in its original context, transfer learning with a modern architecture such as EfficientNetV2-M yields significantly better performance for smartphone-based camera identification tasks.

4. Conclusion

This work explored the problem of source camera identification using deep learning techniques. Initially, a reproduction of the CameraConvNet architecture, as proposed in [16], was conducted on the Dresden dataset, achieving a validation accuracy of 81.89%, which is close to the originally reported value of 94.1%. However, when this same architecture was fine-tuned on a smartphone-specific dataset (FloreView), performance significantly dropped to 30.2% accuracy, suggesting poor generalization to mobile-captured images.

To address this limitation, the study further investigated transfer learning using EfficientNetV2-M, a modern convolutional architecture pretrained on ImageNet. By replacing the classification head and fine-tuning the entire network on FloreView, the model achieved a validation accuracy of 82.0%, outperforming both the from-scratch and fine-tuned CameraConvNet models. These results demonstrate the superior representational capacity and adaptability of EfficientNetV2-M for smartphone camera identification.

In summary, while traditional CNN architectures remain viable for controlled datasets, modern transfer learning approaches prove significantly more effective when adapting to real-world data distributions, such as those found in mobile photography. Future work may focus on combining multiple data augmentation techniques or incorporating domain adaptation strategies to further enhance performance across heterogeneous image sources.

References

[1] A. Piva, "An overview on image forensics," *ISRN Signal Processing*, vol. 2013, pp. 1–22, 2013. 1

[2] J. F. Lukas and J. Lukáš, "Digital camera identification from sensor pattern noise," *IEEE Transactions on Information Forensics and Security*, vol. 1, no. 2, pp. 205–214, 2006. 1

[3] S. Bayram, H. Sencar, N. Memon, and I. Avcibas, "Source camera identification based on cfa interpolation," in *IEEE In-*

ternational Conference on Image Processing (ICIP), pp. 69–72, 2005. 1

[4] S. Milani, P. Bestagini, M. Tagliasacchi, and S. Tubaro, "Demosaicing strategy identification via eigenalgorithms," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 2659–2663, 2014. 1

[5] S.-H. Chen and C.-T. Hsu, "Source camera identification based on camera gain histogram," in *IEEE International Conference on Image Processing (ICIP)*, pp. IV–429–IV–432, 2007. 1

[6] K. Choi, E. Lam, and K. Wong, "Automatic source camera identification using the intrinsic lens radial distortion," *Optics Express*, vol. 14, no. 24, pp. 11551–11565, 2006. 1

[7] Z. Deng, A. Gijsenij, and J. Zhang, "Source camera identification using auto-white balance approximation," in *IEEE International Conference on Computer Vision (ICCV)*, pp. 57–64, 2011. 1

[8] J. F. Lukas and J. Lukáš, "Digital camera identification from sensor pattern noise," *IEEE Transactions on Information Forensics and Security*, vol. 1, no. 2, pp. 205–214, 2006. 1

[9] M. Chen, J. Fridrich, M. Goljan, and J. Lukas, "Determining image origin and integrity using sensor noise," *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 1, pp. 74–90, 2008. 1

[10] A. E. Dirik and N. Memon, "Source camera identification based on sensor dust patterns," *PR*, vol. 44, no. 10–11, pp. 2188–2198, 2011. 1

[11] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *NIPS*, pp. 1097–1105, 2012. 1

[12] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *ICLR*, 2015. 1

[13] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, pp. 770–778, 2016. 1

[14] T. Gloe and R. Böhme, "The dresden image database for benchmarking digital image forensics," *Journal of Digital Forensic Practice*, vol. 3, no. 2–4, pp. 150–159, 2010. 1, 2

[15] D. Baracchi, D. Shullani, M. Iuliani, and A. Piva, "Floreview: An image and video dataset for forensic analysis," *IEEE Access*, vol. 11, pp. 109267–109282, 2023. 1, 2

[16] L. Baroffio, L. Bondi, P. Bestagini, and S. Tubaro, "Camera identification with deep convolutional networks," *arXiv preprint arXiv:1603.01068*, 2016. 2, 3, 4

[17] M. Tan and Q. Le, "Efficientnetv2: Smaller models and faster training," in *International Conference on Machine Learning (ICML)*, pp. 10096–10106, 2021. 2

[18] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 248–255, IEEE, 2009. 2