Reliable Camera Model Identification Using Sparse Gaussian Processes

Benedikt Lorch, Franziska Schirrmacher, Anatol Maier, and Christian Riess, *Member, IEEE*

***Abstract*—Identifying the model of a camera that has captured an image can be an important task in criminal investigations. Many methods assume that the image under analysis originates from a given set of known camera models. In practice, however, a photo can come from an unknown camera model, or its appearance could have been altered by unknown post-processing. In such a case, forensic detectors are prone to fail silently.**

One way to mitigate silent failures is to use a rejection mechanism for unknown examples. In this work, we propose Gaussian processes (GPs), which intrinsically provide such a rejection mechanism. This makes GPs a potentially powerful tool in multimedia forensics, where forensic analysts regularly work on images from unknown origins. We demonstrate that GPs scale well to the task of camera model identification. Probabilistic predictions from a GP classifier achieve high classification accuracy for known camera models while providing reliable uncertainty estimates. The built-in uncertainty estimates effectively tackle open-set camera model identification, outperforming two state-of-the-art methods.

***Index Terms*—camera model identification, Gaussian processes, open-set classification**

I. Introduction

LINKING a photo to a particular camera device or model is an important task in criminal investigations. Previous work has shown that camera devices leave characteristic traces during image acquisition. Some of these traces such as PRNU are specific to the source device [16], while others are shared among devices of the same make and model. Source device identification requires access to images from the specific candidate device, which can be expensive to obtain or may not be available at all. As a complementary technique, camera model identification only needs images from another instance of the same camera model and facilitates narrowing down the set of candidate devices [15].

Previous work on camera model identification can be divided into model-based and learning-based approaches. Some model-based approaches aim at characterizing a particular component in the imaging pipeline [15], while others derive a fingerprint from DCT [23] or noise statistics [24]. In both cases, the camera model is identified by hypothesis testing.

Conversely, learning-based methods derive model-specific traces from general-purpose feature descriptors, which are designed to capture complex interactions between the components of the camera processing pipeline. Early work devised global feature descriptors based on color formation, image quality metrics, wavelet features, and binary similarity measures [4, 11, 14, 26]. More recent work has focused on local interactions between neighboring pixels. One of these feature descriptors is spatial rich models [7]. Marra *et al*. evaluated different variants of the spatial rich models and showed that SPAM features outperformed previously proposed feature descriptors on the task of camera model identification [18]. Furthermore, SPAM features have also been used for forgery detection and localization in images [5, 6] and videos [20].

Many studies have shown that feature-based camera model identification works in a closed-set scenario. However, only few studies consider the practically relevant open-set scenario, where an investigator may work with an image from an unknown camera model. Additionally, machine learning classifiers are sensitive to post-processing that is not covered by the training data [17]. In both cases, false accusations can be prevented with the ability to recognize that an image does not belong to any of the known cameras, or that it differs from the training data.

For the open-set scenario, related work builds on classifiers that can reject images from unknown cameras. Gloe trained a one-class SVM for each camera model, but reported only moderate performance [9]. Wang *et al*. combined a set of one-class SVMs with a multi-class SVM [27]. Costa *et al*. moved the decision boundary of an SVM towards the positive class to avoid false positive camera device identifications [21]. Bayar and Stamm partitioned the known camera models into *knowns* and *known unknowns* and trained a binary classifier to recognize unknown camera models [1]. However, a recent evaluation reported that this binary classifier is sensitive to the choice of *known unknowns* [19]. In that study, the *Probability of Inclusion*-SVM (PI-SVM) achieved the best performance [19].

An underexplored solution for rejecting unknown inputs are classifiers based on Bayesian modeling. In particular, Gaussian processes provide a suitable framework for multi-class classification and yield an uncertainty range with each prediction. This makes GPs a potentially powerful tool for camera model identification in the open-set scenario. As our contribution, we demonstrate that a sparse Gaussian process classifier (GPC) [12] based on SPAM features is effective at camera model identification. The uncertainty in the GPC's predictions enables rejecting unknown test images, *e.g*., from unknown cameras or unseen post-processing. We show that the proposed GPC outperforms the combined classification framework from [27] and the PI-SVM from [19] and study the trade-off between sparsity and performance and the impact of the image resolution.

Work was supported by Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) as part of the Research and Training Group 2475 "Cybercrime and Forensic Computing" (grant #393541319/GRK2475/1-2019).

The authors are with the IT Security Infrastructures Lab at the Friedrich-Alexander University Erlangen-Nuremberg, 91058 Erlangen, Germany (e-mail: *firstname.lastname*@fau.de). The source code is available at https://www.cs1.tf.fau.de/research/multimedia-security.

II. Gaussian Process Classification

In this work, we use a GP as black-box function approximator to predict the camera model, and we use the GP's predictive variance to reject unknown camera models as well as unseen post-processing. A GP is a non-parametric Bayesian method to approximate arbitrary functions via probabilistic predictions. This section gives a short introduction to sparse Gaussian process classification as found in [12, 22].

A GP defines a probability distribution over functions where the function values $f(\boldsymbol{x})$ at any position $\boldsymbol{x} \in \mathbb{R}^D$ are random variables with a joint Gaussian distribution [2]. Following standard practice, we assume that this joint distribution has zero mean and the covariance is defined by a kernel function, *i.e.*, $p(\boldsymbol{f}|\boldsymbol{X}) = \mathcal{N}(\boldsymbol{f}|\boldsymbol{0}, k(\cdot,\cdot))$ where $\boldsymbol{f} = [f(\boldsymbol{x}_1), \ldots, f(\boldsymbol{x}_n)]$ and $\boldsymbol{X} = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]$ for n training samples. We choose an isotropic radial basis functions (RBF) kernel

$$k(\boldsymbol{x}_1, \boldsymbol{x}_2) = \sigma_f^2 \exp\left(-\frac{\|\boldsymbol{x}_1 - \boldsymbol{x}_2\|_2^2}{2\,l^2}\right) \quad , \tag{1}$$

where the output scale $\sigma_f^2 \in \mathbb{R}$ and the length scale $l \in \mathbb{R}$ are hyper-parameters.

GPs for classification are derived from the noise-free regression formulation. In noise-free regression, the goal is to obtain the prediction f_* for a new data point $\boldsymbol{x}_* \in \mathbb{R}^D$ by conditioning the joint distribution $p(\boldsymbol{y}, f_*|\boldsymbol{X}, \boldsymbol{x}_*)$ on the observations $\boldsymbol{y} \in \mathbb{R}^n$. The resulting predictive distribution is denoted as $p(f_*|\boldsymbol{x}_*, \boldsymbol{X}, \boldsymbol{y})$.

For multi-class classification, we assume one latent function $\boldsymbol{f}^c$ for each of the C classes and define an independent GP prior $p(\boldsymbol{f}^c|\boldsymbol{X})$ over each latent function. Each GP prior introduces its own kernel hyper-parameters, *i.e.*, $\boldsymbol{l}, \boldsymbol{\sigma_f^2} \in \mathbb{R}^C$. The latent variables at all training points are denoted as $\mathfrak{f} \coloneqq \left[\boldsymbol{f}^1, ..., \boldsymbol{f}^C\right] \in \mathbb{R}^{n \times C}$. The one-hot encoded training labels are $\boldsymbol{\gamma} \coloneqq \left[\boldsymbol{y}^1, ..., \boldsymbol{y}^C\right] \in \mathbb{R}^{n \times C}$. The posterior over the latent variables $p(\mathfrak{f}|\boldsymbol{X}, \boldsymbol{\gamma})$ follows from Bayes' theorem, but in our case needs to be approximated as described below.

Predictions are computed in two steps. The first step obtains the latent variables $\boldsymbol{f}_* = \left[f_*^1, \ldots, f_*^C\right] \in \mathbb{R}^C$ at test point $\boldsymbol{x}_*$:

$$p(\boldsymbol{f}_*|\boldsymbol{x}_*, \boldsymbol{X}, \boldsymbol{\gamma}) = \int p(\boldsymbol{f}_*|\boldsymbol{x}_*, \boldsymbol{X}, \mathfrak{f})\, p(\mathfrak{f}|\boldsymbol{X}, \boldsymbol{\gamma})\, d\mathfrak{f} \quad . \tag{2}$$

In the second step, the C latent variables are rescaled by a softmax function to obtain a posterior distribution over the categorical classes

$$p(\boldsymbol{y}_*|\boldsymbol{x}_*, \boldsymbol{X}, \boldsymbol{\gamma}) = \int p(\boldsymbol{y}_*|\boldsymbol{f}_*)\, p(\boldsymbol{f}_*|\boldsymbol{x}_*, \boldsymbol{X}, \boldsymbol{\gamma})\, d\boldsymbol{f}_* \quad , \tag{3}$$

where the likelihood is defined as

$$p(y_*^c|\boldsymbol{f}_*) = \frac{\exp(f_*^c)}{\sum\limits_{c' \in C} \exp(f_*^{c'})} \quad . \tag{4}$$

Exact analytical treatment is not possible as the Gaussian prior and the softmax likelihood are non-conjugate. Following [12], we approximate the posterior distribution over the latent variables $p(\mathfrak{f}|\boldsymbol{X}, \boldsymbol{\gamma})$ by a variational distribution $q(\boldsymbol{u}) = \mathcal{N}(\boldsymbol{u}|\boldsymbol{m}, \boldsymbol{S})$ with learnable parameters $\boldsymbol{m}$ and $\boldsymbol{S}$. Here, $\boldsymbol{u} \in \mathbb{R}^m$ are the latent variables at inducing points $\boldsymbol{Z} \in \mathbb{R}^{m \times D}$. These inducing points aim to compress the training data into a subset of $m \ll n$ auxiliary points learned during training. Besides providing a tractable optimization problem, this sparse variational approximation reduces the computational complexity of inference from $\mathcal{O}(n^3)$ to $\mathcal{O}(nm^2)$ [25].

The parameters of the variational posterior and the covariance hyper-parameters are obtained by maximizing the evidence lower bound (ELBO)

$$\mathcal{L}_{\text{ELBO}} = \sum_{i=1}^{n} \mathbb{E}_{q(\boldsymbol{f}_i)} \left[\log p(\boldsymbol{y}_i \,|\, \boldsymbol{f}_i)\right] - \text{KL}\left[q(\mathbf{u}) \| p(\mathbf{u})\right] \quad , \tag{5}$$

where KL denotes the Kullback-Leibler divergence. Maximizing the ELBO is equivalent to minimizing the KL divergence between the variational posterior and the intractable true posterior.

After obtaining the variational posterior, the posterior over the latent variables for the test example $\boldsymbol{x}_*$ is computed analogously to Eq. 2 by marginalizing over the inducing variables, *i.e.*,

$$q(\boldsymbol{f}_*|\boldsymbol{x}_*) = \int p(\boldsymbol{f}_*|\boldsymbol{x}_*, \boldsymbol{Z}, \boldsymbol{u}) q(\boldsymbol{u})\, d\boldsymbol{u} \quad . \tag{6}$$

The predictive distribution from Eq. 3 is approximated as

$$q(\boldsymbol{y}_*|\boldsymbol{x}_*) = \int p(\boldsymbol{y}_*|\boldsymbol{f}_*)\, q(\boldsymbol{f}_*|\boldsymbol{x}_*)\, d\boldsymbol{f}_* \quad . \tag{7}$$

Because Eq. 7 is intractable, we approximate the integration using T Monte Carlo samples. The mean is approximated as

$$\hat{q}(\boldsymbol{y}_*|\boldsymbol{x}_*) = \frac{1}{T} \sum_{i=1}^{T} p(\boldsymbol{y}_*|\boldsymbol{f}_*^{(i)}) \quad , \tag{8}$$

where $\boldsymbol{f}_*^{(i)}$ denotes the i-th draw from the latent posterior $q(\boldsymbol{f}_*|\boldsymbol{x}_*)$. The unbiased variance is computed as

$$\text{Var}\left[q(\boldsymbol{y}_*|\boldsymbol{x}_*)\right] = \frac{1}{T-1} \sum_{i=1}^{T} \left(p(\boldsymbol{y}_*|\boldsymbol{f}_*^{(i)}) - \hat{q}(\boldsymbol{y}_*|\boldsymbol{f}_*)\right)^2 \tag{9}$$

and we take the mean over the classes as uncertainty estimate.

III. Experimental Setup

In this section, we describe the data splitting, the feature extraction, and the training details for the evaluated methods.

1) Dataset: The Dresden database provides 18 camera models with images from at least two devices [10]. As in previous work [3], we consider the *Nikon D70* and *Nikon D70s* the same model. We randomly select $C = 10$ out of these 18 as known camera models. From each of the known camera models, we randomly pick one device for the training set. The images from the other devices of known camera models are assigned to the test set. To ensure that all known camera models are represented equally, we sample for each device the same number of images. This number is determined by the device with the smallest number of images. To avoid

side effects from scene content, we perform a random, non-overlapping split by scene using 80% of the scenes for training and 20% for testing. The remaining 16 camera models are used to evaluate the rejection ability of unknown cameras.

2) Feature representation: While the GPC can operate on any feature representation, we use in this work the popular SPAM features with identical configuration as previous work on camera model identification [18]. This yields a 338-dimensional statistical feature vector per image.

3) GPC Training: The variational posterior and kernel hyper-parameters are optimized by maximizing the ELBO. We use the Adam optimizer with a learning rate of 0.001, and coefficients $\beta_1 = 0.9$, and $\beta_2 = 0.999$. The learning rate is divided by 10 whenever the loss plateaus for 1k iterations. Training terminates when the loss has not improved for 20k iterations. The kernel length scales $\boldsymbol{l} \in \mathbb{R}^C$ and output scales $\boldsymbol{\sigma_f} \in \mathbb{R}^C$ are initialized to 0 and are transformed by a softplus function to ensure positivity. If not mentioned otherwise, we use $m = 512$ inducing points, which are initialized by random selection from the training samples. The integral in Eq. 7 is approximated using $T = 100$ Monte Carlo samples at training time and $T = 1000$ samples at test time for increased precision. All experiments are implemented using *GPyTorch 1.2.1* [8] and executed on a NVIDIA GTX 1080 Ti GPU.

4) Competing methods: We compare the proposed method to two previous approaches for open-set camera model identification. Wang *et al*. proposed a combined classification framework (CCF) consisting of a one-class RBF-SVM per camera model [27]. A test sample is rejected if none of the SVMs recognizes it. If exactly one SVM recognizes the test sample, it is assigned the corresponding class. If a test sample is recognized by more than one SVM, a multi-class RBF-SVM is trained on-the-fly to distinguish the possible classes.

As one of the most successful methods used for open-set camera model identification [19], the PI-SVM is an SVM extension that re-calibrates the decision scores to (unnormalized) posterior probabilities by modeling the most extreme positive training data [13]. For multi-class classification, the PI-SVM fits one SVM with an RBF kernel for each class using a 1-vs-rest approach. After training, the PI-SVM returns the maximum posterior probability and the corresponding class label, which also allows rejecting uncertain test examples.

The regularization hyper-parameters of CCF and PI-SVM are selected via the grid search described in [19], which simulates an open-set setup by randomly splitting the C training classes into $\frac{C}{2}$ known and $\frac{C}{2}$ known-unknown camera models. 80% of the images from the known set are used for training, 20% for validation. From the known-unknown set, we randomly pick the same number of images to obtain an equal number of images from known and known-unknown models in the validation set. This random splitting procedure is repeated 100 times for each combination of hyper-parameters. For each split, we use the validation set to evaluate the classification accuracy of known samples (AKS) and the accuracy of identifying unknown samples (AUS). For computing the AUS with the PI-SVM, we binarize the continuous posterior estimate using the threshold closest to the top-left corner of the receiver operating characteristic (ROC) curve. Finally, we pick the set of hyper-parameters that maximizes the sum of AKS and AUS on average over the 100 splits. The final classifier is trained on all C camera models with the best hyper-parameters.

TABLE I
CLASSIFICATION ACCURACY OF KNOWN AND DETECTION OF UNKNOWN CAMERA MODELS WITH FULL-RESOLUTION IMAGES

Method	Known model accuracy	Detection of unknown models AUC
CCF	0.85 $\pm$ 0.04	0.67 $\pm$ 0.01
PI-SVM	0.97 $\pm$ 0.01	0.87 $\pm$ 0.02
GPC	**0.98** $\pm$ **0.01**	**0.94** $\pm$ **0.02**

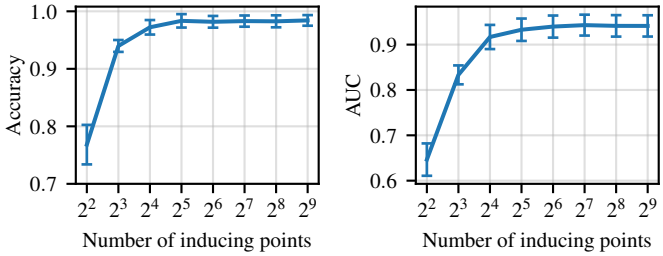

Fig. 1. Classification accuracy of known camera models (left) and detection of unknown models (right). The error bars denote the standard deviation over five training-test splits. The GPC's performance increases with the number of inducing points, but saturates between 128 and 512 inducing points.

5) Evaluation metrics: The competing methods are compared in terms of their classification accuracy on known camera models, denoted as *in-distribution* data, and in their ability to detect *out-of-distribution* images from unknown camera models or unseen post-processing. For the latter performance metrics, we compare the predictive variances of in-distribution and out-of-distribution images. The larger of these two sets is randomly subsampled such that the sets contain the same number of images. We then report the area under the ROC curve (AUC) as a threshold-independent metric.

IV. RESULTS

We first compare the proposed GPC to the CCF and the PI-SVM, followed by an ablation study on the number of inducing points. Subsequently, we evaluate the GPC's performance w.r.t. the image resolution, the number of known camera models, and with images distorted by unseen post-processing.

1) Comparison to related methods: Table I reports the classification accuracy with full-resolution test images from known camera models and the ability to reject images from unseen camera models. The results are averaged over five training-test splits where different instances of the same camera models were randomly selected for the training set. The GPC exhibits the highest accuracy, and it clearly outperforms the PI-SVM in the AUC for detecting unknown models As a sidenote (not reported in Tab. I), the linear SVM by Marra *et al*. [18] yields an even higher accuracy of 99.8%, but it is unable to reject unknown models.

2) Number of inducing points: Figure 1 provides an ablation on the number of the GPC's inducing points. The error bars show the standard deviation over five training runs with

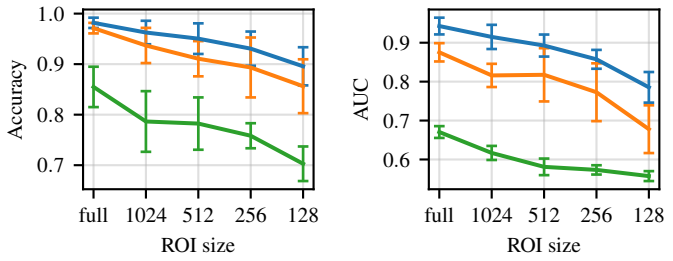

Fig. 2. Classification accuracy of known camera models (left) and detection of unknown camera models (right). Performance of GPC (blue), PI-SVM (orange), and CCF (green) with SPAM features extracted from smaller ROI sizes (*full* denotes full image resolution).

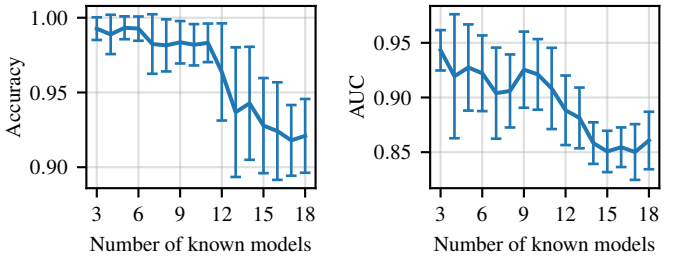

Fig. 3. Classification accuracy of known (left) and detection of unknown camera models (right) as a function of the number of known camera models.

varying training-test splits. During training, the GPC seeks to compress the training points into a lower number of inducing points so as to reduce computational complexity. As expected, the performance increases with the number of inducing points. With only 16 inducing points, the classification accuracy of the GPC with known camera models reaches 0.97. The detection of unseen camera models benefits from more inducing points but saturates between 128 and 512 inducing points. In comparison, a non-sparse GP ($m = n$ where n ranges between 1030 and 1350 training images depending on the random selection of known devices) achieves the same classification accuracy of 0.98 and AUC of 0.94. On our hardware, training the non-sparse GP requires 24.3 hours while the sparse GP with $m = 512$ only needs 3.8 hours. Given that the non-sparse GP does not yield any performance benefit, we use in the remaining experiments the GPC with $m = 512$ due to its significantly lower computational cost.

3) Impact of ROI size: In Fig. 2, we compare the three methods with SPAM features extracted from a region-of-interest (ROI) of smaller size. The training and test images are center-cropped to squares of side lengths between 1024 down to 128 prior to feature extraction. As expected, both classification accuracy and AUC decrease for smaller ROI sizes. The GPC outperforms the PI-SVM in classification accuracy and AUC. Only on small patches of size 128 the AUC of both methods is on-par.

4) Number of known camera models: In Fig. 3, we study the GPC's performance w.r.t. the number of known camera models. More specifically, we randomly select the desired number of camera models, train on images from one device each, and test with the images from the other devices. To facilitate a fair comparison across different numbers of known models, the set of unknown cameras consists only of those

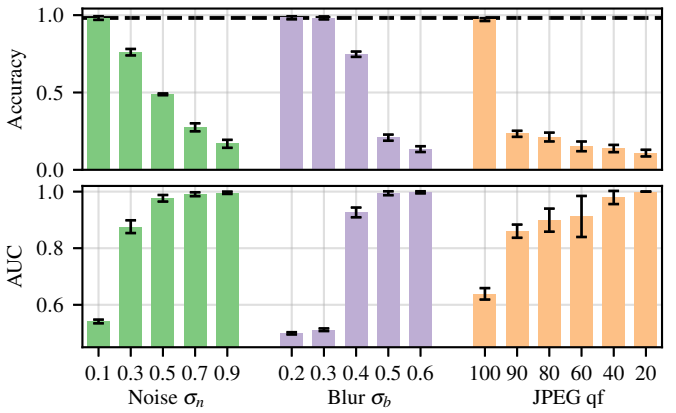

Fig. 4. Classification accuracy for post-processed images (top) and ability to reject unseen post-processing (bottom). The classification accuracy decreases with higher distortion but the predictive variance enables detecting many of the post-processed images. The black dashed line indicates the accuracy for test images without any post-processing.

8 camera models with one device in the Dresden database. Figure 3 reports the accuracy with test images from known camera models but unseen devices (left) and the detection of unseen camera models (right). The error bars show the standard deviation over five training-test splits with randomly selected known models. Overall, the classification accuracy remains at around 0.98 for up to 11 known camera models and decreases to 0.92 with 18 known camera models. The AUC ranges from 0.94 to 0.90 for the first 11 known camera models and achieves more than 0.85 with more camera models.

5) Unseen post-processing: In addition to images from unknown camera models, the forensic detector may be used on images with unseen post-processing. In Fig. 4, we study the effect of additive Gaussian noise with standard deviation σ_n, Gaussian blur with standard deviation σ_b, and JPEG compression with quality factor *qf*. With little distortion, the test accuracy remains close to 0.98, as in the case without any post-processing, and the predictive variances remain close to 0. As the amount of distortion increases, the classification accuracy drops. At the same time, the predictive variance increases, which allows detecting many of the post-processed images. We also observe that the classifier is sensitive to unseen JPEG compression. However, also here the predictive variance facilitates detecting such unseen JPEG compression. This hints an analyst at improving the distribution matching between training and test data.

V. Conclusion

We proposed Gaussian process classifiers for reliable camera model identification. The GPC achieves a classification accuracy of 0.98 on 10 camera models. At the same time, the probabilistic predictions enable detecting images from unknown camera models and unseen post-processing. The ability to recognize unknown types of images is important to prevent false accusations. Despite the probabilistic predictions, classifiers should be able to cope with smaller variations in the test data, *e.g.*, mild compression. In future work, we plan to study possible GPC resilience improvements through data augmentation while retaining computational efficiency.

REFERENCES

[1] B. Bayar and M. C. Stamm, "Towards open set camera model identification using a deep learning framework," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, Apr. 2018, pp. 2007–2011.

[2] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006.

[3] L. Bondi, L. Baroffio, D. Güera, P. Bestagini, E. J. Delp, and S. Tubaro, "First steps toward camera model identification with convolutional neural networks," *IEEE Signal Processing Letters*, vol. 24, no. 3, pp. 259–263, 2017.

[4] O. Celiktutan, B. Sankur, and I. Avcibas, "Blind identification of source cell-phone model," *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 3, pp. 553–566, Sep. 2008.

[5] D. Cozzolino, D. Gragnaniello, and L. Verdoliva, "Image forgery detection through residual-based local descriptors and block-matching," in *IEEE International Conference on Image Processing*, Oct. 2014, pp. 5297–5301.

[6] ——, "Image forgery localization through the fusion of camera-based, feature-based and pixel-based techniques," in *IEEE International Conference on Image Processing*, Oct. 2014, pp. 5302–5306.

[7] J. Fridrich and J. Kodovsky, "Rich models for steganalysis of digital images," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 3, pp. 868–882, 2012.

[8] J. R. Gardner, G. Pleiss, D. Bindel, K. Q. Weinberger, and A. G. Wilson, "GPyTorch: Blackbox matrix-matrix Gaussian process inference with GPU acceleration," in *Advances in Neural Information Processing Systems*, Dec. 2018, pp. 7587–7597.

[9] T. Gloe, "Feature-based forensic camera model identification," in *Transactions on Data Hiding and Multimedia Security VIII*. Springer, 2012, pp. 42–62.

[10] T. Gloe and R. Böhme, "The Dresden image database for benchmarking digital image forensics," in *ACM Symposium on Applied Computing*, Mar. 2010, pp. 1584–1590.

[11] T. Gloe, K. Borowka, and A. Winkler, "Feature-based camera model identification works in practice," in *Information Hiding*, Jun. 2009, pp. 262–276.

[12] J. Hensman, A. G. de G. Matthews, and Z. Ghahramani, "Scalable variational gaussian process classification," in *International Conference on Artificial Intelligence and Statistics*, vol. 38, May 2015, pp. 351–360.

[13] L. P. Jain, W. J. Scheirer, and T. E. Boult, "Multi-class open set recognition using probability of inclusion," in *European Conference on Computer Vision*, Sep. 2014, pp. 393–409.

[14] M. Kharrazi, H. T. Sencar, and N. Memon, "Blind source camera identification," in *International Conference on Image Processing*, Oct. 2004, pp. 709–712.

[15] M. Kirchner and T. Gloe, "Forensic camera model identification," in *Handbook of Digital Forensics of Multimedia Data and Devices*. Wiley, 2015, ch. 9, pp. 329–374.

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

[17] S. Mandelli, N. Bonettini, P. Bestagini, and S. Tubaro, "Training CNNs in presence of JPEG compression: Multimedia forensics vs computer vision," in *IEEE International Workshop on Information Forensics and Security*, Dec. 2020.

[18] F. Marra, G. Poggi, C. Sansone, and L. Verdoliva, "Evaluation of residual-based local features for camera model identification," in *New Trends in Image Analysis and Processing – ICIAP 2015 Workshops*, Sep. 2015, pp. 11–18.

[19] P. R. Mendes Júnior, L. Bondi, P. Bestagini, S. Tubaro, and A. Rocha, "An in-depth study on open-set camera model identification," *IEEE Access*, vol. 7, pp. 180 713–180 726, Jun. 2019.

[20] P. Mullan, D. Cozzolino, L. Verdoliva, and C. Riess, "Residual-based forensic comparison of video sequences," in *IEEE International Conference on Image Processing*, Sep. 2017, pp. 1507–1511.

[21] F. d. O. Costa, M. Eckmann, W. J. Scheirer, and A. Rocha, "Open set source camera attribution," in *SIBGRAPI Conference on Graphics, Patterns and Images*, Aug. 2012, pp. 71–78.

[22] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*. The MIT Press, 2005.

[23] T. H. Thai, F. Retraint, and R. Cogranne, "Camera model identification based on DCT coefficient statistics," *Digital Signal Processing*, vol. 40, pp. 88–100, 2015.

[24] ——, "Camera model identification based on the generalized noise model in natural images," *Digital Signal Processing*, vol. 48, pp. 285–297, 2016.

[25] M. Titsias, "Variational learning of inducing variables in sparse gaussian processes," in *International Conference on Artificial Intelligence and Statistics*, ser. Proceedings of Machine Learning Research, vol. 5, Apr. 2009, pp. 567–574.

[26] A. Tuama, F. Comby, and M. Chaumont, "Camera model identification based machine learning approach with high order statistics features," in *European Signal Processing Conference*, Sep. 2016, pp. 1183–1187.

[27] B. Wang, X. Kong, and X. You, "Source camera identification using support vector machines," in *Advances in Digital Forensics V*, Sep. 2009, pp. 107–118.