Reliable JPEG Forensics via Model Uncertainty

Benedikt Lorch, Anatol Maier, Christian Riess
IT Security Infrastructure Lab
Friedrich-Alexander University Erlangen-Nürnberg
{benedikt.lorch, anatol.maier, christian.riess}@fau.de

***Abstract*—Many methods in image forensics are sensitive to varying amounts of JPEG compression. To mitigate this issue, it is either possible to a) build detectors that better generalize to unknown JPEG settings, or to b) train multiple detectors, where each is specialized to a narrow range of JPEG qualities. While the first approach is currently an open challenge, the second approach may silently fail, even for only slight mismatches in training and testing distributions. To alleviate this challenge, we propose a forensic detector that is able to express uncertainty in its predictions. This allows detecting test samples for which the training distribution is not representative. More specifically, we propose Bayesian logistic regression as an instance of an infinite ensemble of classifiers. The ensemble agrees in its predictions from test samples similar to the training data but its predictions diverge for unknown test samples. The applicability of the proposed method is evaluated on the task of detecting JPEG double compression. The detector achieves high performance on two goals simultaneously: It accurately detects double-JPEG compression, and it accurately indicates when the test data is not covered by the training data. We assert that the proposed method can assist a forensic analyst in assessing detector reliability and in anticipating failure cases for specific inputs.**

I. Introduction

Multimedia forensics aims to provide tools to verify the origin and authenticity of multimedia content. These tools target different user groups with varying requirements ranging from businesses and journalists to criminal investigators. While social media companies need tools that work at scale, criminal investigations require reliable and interpretable tools to produce evidence that is admissible in court. To this end, a broad range of analytical models has been developed where assumptions and error bounds can be stated explicitly.

While rigorous analytical models are certainly preferable, many traces are notoriously hard to describe and isolate, given the lack of knowledge about hardware manufacturing and the abundance of possible processing operations that an image may have undergone. When analytical derivations are infeasible, researchers resort to constructing statistical models from large sets of examples. While machine learning outperforms simple rule-based classifiers, learning-based classifiers tend to overfit to the type of data they were trained on, *i.e.*, the training distribution. When presented with an example of different characteristics, so called *out-of-distribution* examples, the output of the machine learning model becomes somewhat arbitrary. This problem is known as the training-test mismatch [1], [2]. Machine learning models can therefore only be applied to samples that fit to the training distribution.

Given an image to analyze, usually little is known about the origin and processing history of that example. Therefore, it is not trivial to ensure that the training data of a pre-trained detector is representative for the test example. While much research is being devoted to better generalization to unseen scenarios, for example by training on more diverse inputs, most methods are still challenged by out-of-distribution data where they silently fail.

In this work, we explore methods to safeguard forensic detectors from unfamiliar inputs. The proposed method informs the forensic analyst when the input does not match the training distribution, and provides an estimate for the model's degree of uncertainty. This uncertainty estimate can indicate whether (and to what extent) to trust the predictions of a detector. This work intents to facilitate creating trust in machine learning-based forensics and to develop more reliable detectors.

The specific subject of this work is JPEG forensics. Varying JPEG compression settings are a notorious challenge for statistics-based forensic algorithms. An often-proposed solution is to create a specific detector for each particular JPEG quality factor [3], [4], [5]. While this strategy may be able to reduce the training-test mismatch, we will demonstrate that this strategy is not foolproof. As a remedy, we propose using predictive uncertainty for explicitly indicating a training-test mismatch. To estimate predictive uncertainty, we use Bayesian logistic regression as a specific instance of an infinite ensemble of classifiers. Its applicability and usefulness is shown for the detection of double-JPEG compression using first-digit features [6]. These analytically tractable features allow us to thoroughly study the classifier and its predictive uncertainty. The contributions of this work are:

- We demonstrate that an unknown input may easily break state-of-the-art defenses against the training-test mismatch, namely selecting the best-matching detector and creating an input-specific training set.
- We show the benefits of predictive variance for measuring uncertainty in the detection of double JPEG compression.
- We show that Bayesian logistic regression with standard first-digit features reliably detects several pathologic failure cases in JPEG analysis, including unknown JPEG settings and unknown JPEG implementations.

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).

WIFS'2020, December, 6-11, 2020, New York, USA. 978-1-7281-9930-6/20/$31.00

II. Related Work

A. JPEG Double Compression

Traces of multiple JPEG compressions have always been a subject of interest in the forensics community. A common assumption is that original images are compressed once inside the camera. Traces from subsequent JPEG compressions indicate additional processing, which may be a first cue towards detecting modifications to the image content. Most work on detecting double JPEG compression can be categorized into whether the DCT grids of successive compressions are shifted or aligned. Notable approaches to detecting non-aligned double compression include blocking artifacts [7] and the periodicity of DCT coefficients [8]. Periodic artifacts and discontinuities in the DCT coefficients have also been exploited to detect aligned JPEG double compression [9], [10]. Another telltale sign indicating multiple compressions is the distribution of the first digits of the DCT coefficients. First digits from single-compressed images follow a distribution described by Benford's law. Multiple compressions change this distribution, which indicates double compression [11] and allows inferring the quantization step of the first compression [6]. First-digit features can also be used to estimate the number of consecutive JPEG compressions [12] and to localize image tampering [13]. In this work, we evaluate the proposed classifier on this standard set of first-digit features due to its simplicity but note that our approach is applicable to any feature set.

B. Novelty Detection

Many forensic algorithms are formulated as novelty detection tasks to avoid overly limiting assumptions on the type of image manipulation. These approaches model specific image properties either implicitly [14] or explicitly [15]. Genuine images can be validated by these properties, but manipulated content is assumed to appear "anomalous", and can be detected as such. From a technical perspective, novelty detection determines whether a new observation notably differs from the training data. While several works observe their methods to be sensitive to a mismatch between training and test data [1], [2], only few works address this issue. Related work on open-set camera-model identification used one-class SVMs [15], [16] or binary SVMs [17] to contend with unknown camera models. One-and-a-half class SVMs have also been used to detect image manipulations in an adversarial environment [18].

Many safety-critical applications require knowledge of erroneous predictions. Therefore, the machine learning community developed principled approaches to assessing uncertainty in the prediction of a model. Recent work used classifier ensembles to identify out-of-distribution examples [19], [20]. These works follow the idea that the ensemble largely agrees in its predictions from in-distribution samples, but diverges for unseen examples. Recent efforts combine neural networks with Bayesian methods that allow training an infinite ensemble of classifiers and provide a principled way to reason about predictive uncertainty [21]. In a Bayesian neural network, each parameter is represented by a probability distribution that captures uncertainty about the parameters. Despite promising results on detecting out-of-distribution examples [22], [23], it is still a challenge to apply Bayesian neural networks to large data sets, mainly due to expensive hyper-parameter tuning and the requirement of using restrictive approximations [24]. In this work, we study Bayesian logistic regression, which is a simpler instance of the Bayesian framework. While this method is less powerful than Bayesian neural networks, we aim to demonstrate the benefits of modeling parameters as probability distributions, analyze the impact of the prior distribution, and outline a shortcoming of this method.

III. Bayesian Logistic Regression

We discuss the general background on Bayesian logistic regression here, more details can be found in [25].

Logistic regression models a linear decision boundary with scalar weights that are found via maximum likelihood. For a M-dimensional input $\boldsymbol{x}$, the output of the linear model is

$$y(\boldsymbol{x}, \boldsymbol{w}) = \sigma(\boldsymbol{w}^T \boldsymbol{x}) \ , \tag{1}$$

where σ denotes the logistic sigmoid function and $\boldsymbol{w}$ represents the M-dimensional parameter vector.

The posterior probability that $\boldsymbol{x}$ belongs to class $\mathcal{C}_1$ can be formulated as the expectation with respect to the posterior distribution over the weights, *i.e.*,

$$p(\mathcal{C}_1|\boldsymbol{x}) = \mathbb{E}_{\delta(\boldsymbol{w})}\left[y(\boldsymbol{x}, \boldsymbol{w})\right] = \int y(\boldsymbol{x}, \tilde{\boldsymbol{w}}) \delta(\tilde{\boldsymbol{w}} - \boldsymbol{w}) \mathrm{d}\tilde{\boldsymbol{w}} \ . \tag{2}$$

The model is parameterized by scalar values. Hence, the posterior distribution is given by the Dirac delta distribution $\delta(\boldsymbol{w})$, thus $p(\mathcal{C}_1|\boldsymbol{x}) = y(\boldsymbol{x}, \boldsymbol{w})$. As a consequence, this model obtains a so-called point estimate that does not represent or indicate uncertainty in its prediction.

Conversely, in the Bayesian formulation, each weight is represented by a probability distribution. During training, the posterior distribution over the weights $p(\boldsymbol{w}|\mathcal{D})$ is sought. It captures how probable each parameter configuration is, given the training data $\mathcal{D} = \{(\boldsymbol{x}_n, t_n)\}_{n=1}^N$. As the prior distribution we assume a zero-mean Gaussian with covariance matrix $\boldsymbol{S_0}$:

$$p(\boldsymbol{w}|\boldsymbol{S_0}) = \mathcal{N}(\boldsymbol{w}|\boldsymbol{0}, \boldsymbol{S_0}) \quad . \tag{3}$$

The posterior distribution over the weights is given by the Bayes theorem:

$$p(\boldsymbol{w}|\mathcal{D}) = \frac{p(\mathcal{D}|\boldsymbol{w}) p(\boldsymbol{w}|\boldsymbol{S_0})}{p(\mathcal{D}|\boldsymbol{S_0})} \ . \tag{4}$$

In this binary classification problem, the likelihood can be written in terms of a Bernoulli distribution

$$p(\mathcal{D}|\boldsymbol{w}) = \prod_{n=1}^{N} y(\boldsymbol{x}_n, \boldsymbol{w})^{t_n} (1 - y(\boldsymbol{x}_n, \boldsymbol{w}))^{1-t_n} \ . \tag{5}$$

The normalization over the evidence $p(\mathcal{D}|\boldsymbol{S_0})$ in Eqn. 4 ensures that the posterior is a valid probability distribution, which, unfortunately, is analytically intractable. We therefore

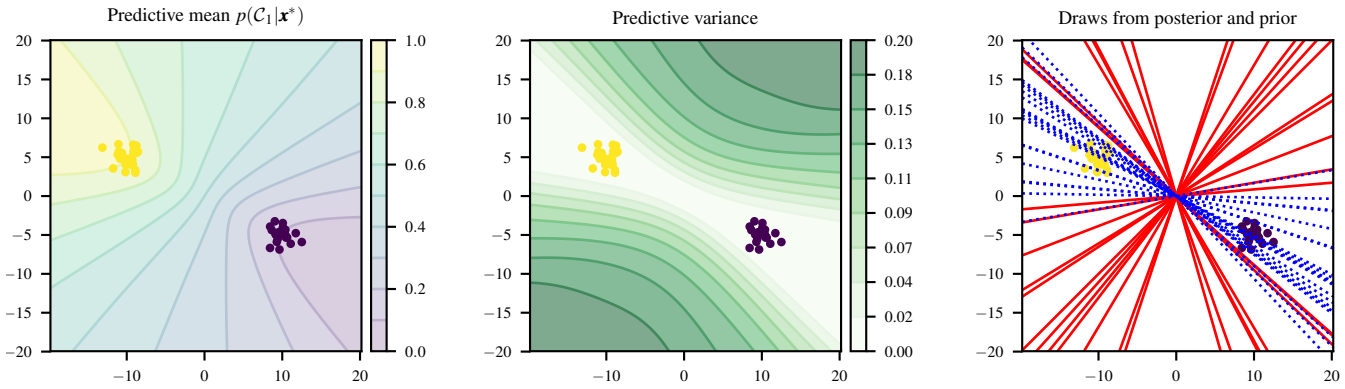

Fig. 1. Bayesian logistic regression on a 2-D toy example with regularization parameter $\beta = 2$. Left: Predictive mean for the posterior probability. Middle: Predictive variance. The predictive variance is low in regions where several draws from the posterior distribution result in similar predictions. It increases when several models disagree. Right: 20 decision boundaries drawn from the posterior (solid red) and 20 draws from the prior distribution (dotted blue). Aligning the prior along the data may appear counter-intuitive, but helps to spread the variety of posteriors (see text for details).

resort to a variational framework [26] that approximates the logistic likelihood function $p(\mathcal{D}|\boldsymbol{w})$ by a lower bound $h(\mathcal{D}|\boldsymbol{w}, \boldsymbol{\eta})$. This bound is conjugate to the Gaussian prior, at the cost of one additional parameter η_n per training example that controls the tightness of the approximation:

$$p(\mathcal{D}|\boldsymbol{w})p(\boldsymbol{w}|\boldsymbol{S_0}) \geq h(\mathcal{D}|\boldsymbol{w}, \boldsymbol{\eta})p(\boldsymbol{w}|\boldsymbol{S_0}) \quad . \tag{6}$$

Because the Gaussian prior and the lower bound on the likelihood are conjugate, their product forms a Gaussian distribution $q(\boldsymbol{w})$, which is straightforward to normalize. As a result, the variational posterior $q(\boldsymbol{w})$ approximates the intractable posterior distribution $p(\boldsymbol{w}|\mathcal{D})$. The covariance of the variational posterior still depends on the variational parameters $\boldsymbol{\eta}$, which can be optimized with the expectation maximization (EM) algorithm as in [26]. After random initialization of the variational parameters, the E-step calculates the mean and covariance of the variational weight posterior. In the M-step, the expected complete-data log likelihood is maximized w.r.t. the variational parameters using the current estimate of $q(\boldsymbol{w})$:

$$\boldsymbol{\eta} = \arg\max_{\boldsymbol{\eta}} \mathbb{E}_{q(\boldsymbol{w})} \left[\ln h(\mathcal{D}|\boldsymbol{w}, \boldsymbol{\eta})p(\boldsymbol{w}|\boldsymbol{S_0})\right] \quad . \tag{7}$$

We run the EM algorithm for a maximum of 1 000 iterations or until the parameters converge to stable estimates.

Having obtained a variational approximation $q(\boldsymbol{w})$ to the weight posterior, the predictive distribution for an unseen example $\boldsymbol{x}^*$ can be computed by marginalizing over $\boldsymbol{w}$:

$$p(\mathcal{C}_1|\boldsymbol{x}^*) = \int y(\boldsymbol{x}^*, \boldsymbol{w})p(\boldsymbol{w}|\mathcal{D})d\boldsymbol{w} \approx \int y(\boldsymbol{x}^*, \boldsymbol{w})q(\boldsymbol{w})d\boldsymbol{w}. \tag{8}$$

Because the integral in Eqn. 8 cannot be solved analytically, we use Monte Carlo sampling to estimate the mean and the variance of the predictive distribution. The mean is

$$p(\mathcal{C}_1|\boldsymbol{x}^*) \approx \frac{1}{T}\sum_{t=1}^{T} y(\boldsymbol{x}^*, \boldsymbol{w}^{(t)}) \tag{9}$$

where $\boldsymbol{w}^{(t)} \sim q(\boldsymbol{w})$ is drawn from the variational posterior and T denotes the number of Monte Carlo draws.

Figure 1 demonstrates a 2-D toy example for a binary classification task. On the left, the contour lines show the posterior probability for the positive class $\mathcal{C}_1$. The predictive variance is shown in the middle, which encodes the disagreement or uncertainty of the models when drawing from the weight posterior. The plot on the right shows 20 draws from the weight posterior (solid red lines) and the prior distribution (dotted blue lines). At first glance, it may be counter-intuitive to pass the prior through the data (blue). However, we empirically found that this increases the diversity of the posterior decision boundaries (red), which implicitly increases the sensitivity to outliers. In regions with training data, draws from the posterior distribution yield similar predictions and hence a low predictive variance. Regions without training data yield different predictions and hence high predictive variance.

This toy example also hints at a possible failure case of the proposed method: The predictive variance has a blind spot "behind" the samples of a class. Here, the predictive variance is very close to 0, such that outliers cannot be found.

IV. Evaluation

To our knowledge, there is no work directly related to our method. However, the k-nearest-neighbors (kNN) classifier can potentially directly synthesize an uncertainty measure. Another alternative might be an SVM-based classification framework that selects a specialized classifier for each test sample.

The kNN classifier predicts the class label for a test example via a majority vote of the $\mathrm{k} = 5$ nearest training examples. To measure uncertainty, we calculate a sample's average Euclidean distance to its k closest training examples: This distance can be expected to be low for in-distribution test samples, and higher for out-of-distribution test samples.

Additionally, we evaluate a combined classification framework (CCF) composed of two one-class and one two-class SVMs, following a similar proposal for open-set camera model identification [15]. For a pair of compression parameters, one

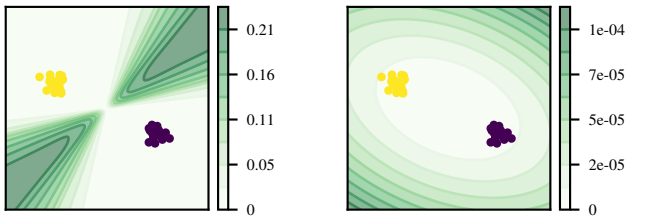

Fig. 2. Predictive variance for $\beta = 10^{-3}$ (left) and $\beta = 10^{6}$ (right). A lower prior precision allows the model to adapt more freely to the training data, while an over-regularized model is unable to adjust to the training data.

one-class SVM is trained on the single-compressed images, and another on the double-compressed images. The two-class SVM is discriminatively trained on both inputs. A sample is determined as single- or double-compressed if it is detected by exactly one of the one-class SVMs. If both one-class SVMs detect it, the two-class SVM decides between both classes. A test sample is rejected if it is not recognized by both the one-class SVMs. The one-class SVMs use radial basis functions, and their tightness parameter $\nu = 0.001$ is determined with a grid search. The two-class SVM uses a linear kernel with regularizer $C = 1.0$. As uncertainty score we use the binary decision whether a test example is recognized as an inlier or outlier.

The proposed Bayesian logistic regression with variational approximation (VLR) is evaluated with two choices of prior distributions. Both priors are zero-mean normal distributions, specified in terms of their precision matrix. The first variant, VLR_{iso}, is isotropic with precision α, *i.e.*, $\boldsymbol{S}_{\boldsymbol{0}}^{-1} = \alpha \boldsymbol{I}$, where $\boldsymbol{I}$ is the M-dimensional identity matrix. The second variant, VLR_{full}, uses a full precision matrix that is set to the training data covariance Σ, *i.e.* $\boldsymbol{S}_{\boldsymbol{0}}^{-1} = \beta\Sigma + \epsilon \boldsymbol{I}$. We add a small constant of $\epsilon = 1\text{e}{-5}$ to the diagonal entries to ensure that the resulting $\boldsymbol{S}_{\boldsymbol{0}}^{-1}$ is positive definite. Setting the prior precision matrix to the data covariance encourages the model to explore more diverse decision boundaries, as shown in Fig. 1.

The hyper-parameters α and β control the regularization strength of the prior, which is shown in Fig. 2 for the toy example: On the left, smaller values for β allow the posterior distribution to flexibly adapt to the training data, but large areas in the features space exhibit only low uncertainty. On the right, if the prior precision is set too high, the resulting over-regularized model cannot properly fit the training data. In our experiments, we set $\alpha = 10^{-5}$ and $\beta = 10^{3}$. The behavior of the prior precision is also subject to an ablation study in Sec. IV-D. While testing, we use $T = 100$ Monte Carlo draws to estimate the predictive mean and variance.

A. Experimental Setup

We evaluate the proposed method on the RAISE 1k dataset consisting of 1 000 images in uncompressed TIFF format [27]. For a given pair of quality factors *(QF1, QF2)*, we create a single-compressed version using *QF2* and a double-compressed version with *QF1* for the first and *QF2* for the second compression. All images are compressed using *libjpeg*. We evaluate quality factors between 50 and 95 in steps of 5.

First-digit features are extracted from the compressed images as described earlier [13]. More specifically, the first nine AC bands in zig-zag scan order are used. For each frequency band, the number of occurrences of each of the nine possible digits in the first position is counted. The resulting histogram is normalized to sum up to one. While previous work only used 27 out of the resulting 81 feature dimensions, we intentionally keep all 81 dimensions to detect when some of the feature dimensions assume values that were not seen during training.

TABLE I
IN-DISTRIBUTION ACCURACY AND DETECTION OF OUT-OF-DISTRIBUTION CASES

Method	In-distribution accuracy	*QF2* mismatch AUC	*QF1* mismatch AUC
kNN	0.93 $\pm$ 0.00	**1.00 $\pm$ 0.00**	0.85 $\pm$ 0.00
CCF	0.92 $\pm$ 0.01	0.97 $\pm$ 0.01	0.73 $\pm$ 0.01
VLR_{iso}	**0.97 $\pm$ 0.00**	0.97 $\pm$ 0.01	0.79 $\pm$ 0.03
VLR_{full}	**0.97 $\pm$ 0.00**	**1.00 $\pm$ 0.00**	**0.91 $\pm$ 0.00**

B. Evaluation Protocol

The 1 000 images are split into 500 images for training and 500 for testing. For a given pair of quality factors *(QF1, QF2)* we train the detector with 1 000 samples formed by the single- and double-compressed versions of the training images. As a pre-processing step, the training data is centered. We evaluate the detector's classification accuracy on the 1 000 test samples formed by the single- and double-compressed version of the test set. Additionally, we evaluate the detector's ability to distinguish between test examples that are aligned with the training data (in-distribution) and test examples that were compressed using different quality factors (out-of-distribution). The out-of-distribution examples use the same 500 images as the in-distribution test set. All three methods for comparison provide an uncertainty/novelty detection score to assess predictive uncertainty. The detector's ability to distinguish between in- and out-of-distribution examples is determined as follows: We compare the predictive uncertainties of the in-distribution test examples and the out-of-distribution test examples, and report the area under the curve (AUC) of the receiver-operator-characteristics (ROC) as a threshold-independent metric. Unless mentioned otherwise, all quantitative experiments are repeated ten times with randomized train-test splits and the results are averaged over these ten runs.

C. Detection of Out-of-Distribution Samples

Table I shows a comparison of kNN and CCF with the proposed method in terms of classification accuracy with in-distribution examples and out-of-distribution AUC. The classification accuracy is averaged over 90 scenarios of quality factor pairs *(QF1, QF2)*, excluding cases where *QF1* = *QF2*. All methods achieve an average accuracy of 92% or

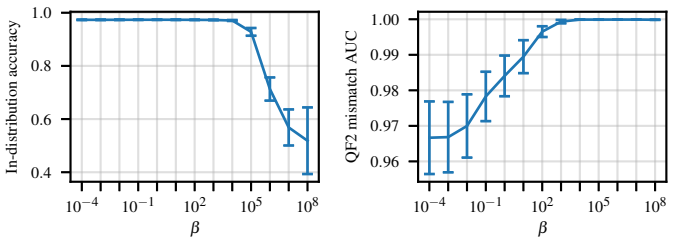

Fig. 3. Left: In-distribution accuracy. Right: Detection of unknown values for *QF2*. With increasing regularization parameter β, in-distribution accuracy is reduced but the detectability of out-of-distribution examples increases.

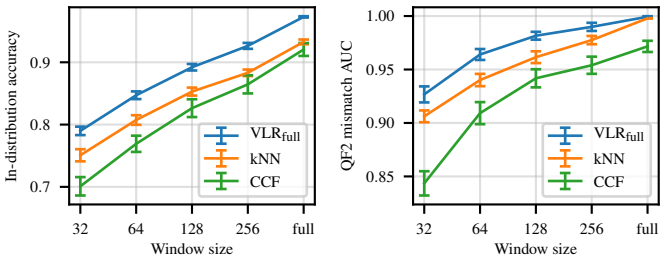

Fig. 4. Left: In-distribution accuracy. Right: Detection of unknown values for *QF2*. The proposed VLR$_{\text{full}}$ also performs best when only information from as little as 32×32 pixels is available.

more, the proposed method VLR$_{\text{full}}$ works best. Upon closer examination, CCF falsely rejects some in-distribution input examples, and kNN performs inferior when $QF2 < QF1$. All methods encounter difficulties for some specific choices of *QF2* when $QF1 = 95$, since in this case only one out of the nine frequency bands used has a quantization factor other than 1 (namely, 2), which makes it extremely difficult to distinguish single- and double-compressed images.

The second column in Tab. I shows the AUC for cases when the training set uses $QF2_{train}$ and test data uses a different $QF2_{test}$. The AUCs are averaged over the 900 different choices for *QF1*, $QF2_{train}$, and $QF2_{test}$. All methods attain an AUC of 0.97 or more. The exact AUC for kNN is 0.997859, while VLR$_{\text{full}}$ even achieves 0.999336.

The third column shows the AUC for cases where the training set uses $QF1_{train}$ and the test data uses a different $QF1_{test}$. Again, the AUCs are averaged over 900 different scenarios. All methods encounter difficulties when $QF1_{test}$ is 95 or when $QF1_{test} = QF2$.

Our proposed method with full covariance prior outperforms the isotropic covariance prior in the detection of training-test mismatches. This shows that initializing the prior close to the data indeed caused the posterior distribution to further spread out, thereby increasing its sensitivity to outlier samples.

D. Ablation Study

Figure 3 illustrates the impact of the hyper-parameter β for controlling the amount of regularization. On the left, the in-distribution accuracy is shown. On the right, the detection of unknown values for *QF2* is shown. As expected, lower values for β result in higher accuracy but reduced detectability of unknown examples. As β increases, the accuracy decreases but unknown examples can be identified more readily.

Figure 4 shows the in-distribution accuracy and recognition of out-of-distribution examples from smaller images. Here, we center-crop all images to a given resolution and extract first-digit features. The resulting features are expected to be more noisy. While kNN and our method achieve a comparable detection score for out-of-distribution examples on the full-resolution images, VLR$_{\text{full}}$ outperforms kNN with subwindows, down to only 32×32 pixels. For smaller windows all methods expose difficulties where $QF2 < QF1$.

E. Failure Cases for Images of Unknown Origin

We show the benefits of our method with two practically relevant cases that are difficult to tackle with existing approaches.

1) Mismatch in JPEG Quantization Tables: A standard approach to mitigating detection penalties from image compression is to train several detectors on different ranges of JPEG qualities. Given an image to analyze, the most suitable pre-trained detector is determined. This is done by estimating the quality factor from the quantization table or by directly using the detector with minimum distance between the quantization tables for the training set and the test image. Unfortunately, this strategy can fail, as demonstrated below.

VLR$_{\text{full}}$ is trained to distinguish single- and double-compressed images with $QF1 = 55$ and $QF2 = 60$. The test images are identical, except that single AC quantization factors differ by ± 1, randomly selected among the first 9 AC quantization factors. Figure 5 (left) shows the accuracy from 500 such test images in dependence of the number of changed quantization factors. As expected, the accuracy decreases when more entries of the quantization table are modified. The error bars denote the standard deviation over ten runs. Compared to our method, a linear SVM is more robust for a limited number of unseen quantization factors, but also quickly drops to guessing chance if five or more AC quantization factors are unseen. Hence, the detector with the smallest least-squares distance to the quantization table may still fail. Detection performance may drop to guessing chance, even if the differences between seen and unseen quantization tables are very small.

Figure 5 (right) shows that the predictive variance can detect such subtle failure cases. The predictive variance increases with more changes to the quantization table, from 0.004 to already 0.120 for only a single deviating quantization factor.

2) Mismatch in JPEG Encoders: Also differences in the JPEG encoders of the training and test data may introduce failure cases, even if the same quantization tables are used. To show this, the training images are compressed with the integer DCT, which is the default in *libjpeg*. The 500 test images use the same quantization table, but are compressed with *libjpeg*'s fast integer DCT. For some pairs of quality factors, our detector misclassified a considerable percentage of the fast DCT test images. For example, for $QF1 = 75$ and $QF2 = 90$, the detection accuracy drops to 0.53.

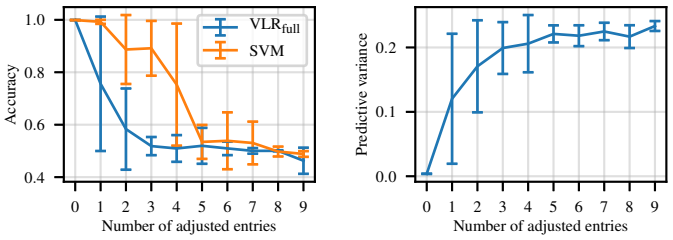

Fig. 5. Left: Accuracies drop with increasing number of ± 1 deviations in AC quantization factors. Right: The predictive variance detects this mismatch.

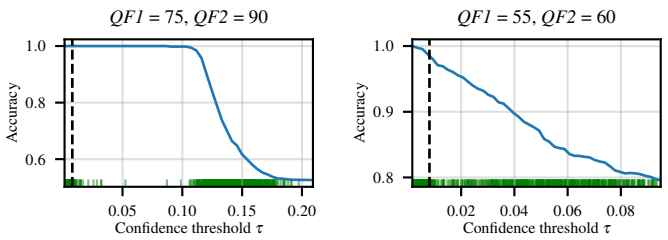

Fig. 6. Detection of training-testing mismatch from JPEG library implementations (for two example pairs QF1 and QF2): Samples with lower predictive variance achieve higher accuracy. Green: Test data distribution. Dashed line: 95% percentile of predictive variances for in-distribution test data.

The exact amount of this performance drop may vary with different classifiers. In any case, the predictive variance can detect this issue. Figure 6 shows the accuracy on the test images where the predictive variance is below a threshold τ. Accuracy decreases with higher predictive variance, which shows that the predictive variance can indicate the alignment of test and training data, and by extension the degree of trust that can be put into such a forensic detector.

V. Conclusions

We have shown that a Bayesian detector can achieve high accuracy in distinguishing single- and double-compressed images, and at the same time reliably identify out-of-distribution input. We studied the properties of this approach on an analytically tractable feature set (first-digit features) and an analytically tractable classifier (Bayesian logistic regression).

The proposed method detects out-of-distribution inputs in a wider range of scenarios than the kNN classifier, which is conceptually simpler, but less flexible. The proposed method also outperforms the more complicated cascaded SVM-based predictors while also being mathematically more elegant.

Promising results are shown for two difficult tasks in applied forensics, namely the detection of training-test mismatches for slightly different JPEG quantization tables and slightly different JPEG library implementations.

References

[1] C. Liu and M. Kirchner, "CNN-based rescaling factor estimation," in *ACM Workshop Inf. Hiding and Multimedia Security*, 2019, pp. 119–124.

[2] B. Diallo, T. Urruty, P. Bourdon, and C. Fernandez-Maloigne, "Improving robustness of image tampering detection for compression," in *Int. Conf. Multimedia Modeling*, 2019, pp. 387–398.

[3] M. Boroumand and J. Fridrich, "Deep learning for detecting processing history of images," *Electronic Imaging*, no. 7, pp. 213–1–213–9, 2018.

[4] D. Cozzolino and L. Verdoliva, "Noiseprint: A CNN-based camera model fingerprint," *IEEE Trans. Inf. Forensics Security*, vol. 15, pp. 144–159, 2020.

[5] M. Barni, A. Costanzo, E. Nowroozi, and B. Tondi, "CNN-based detection of generic contrast adjustment with JPEG post-processing," in *IEEE Int. Conf. Image Processing*, 2018, pp. 3803–3807.

[6] B. Li, Y. Q. Shi, and J. Huang, "Detecting doubly compressed JPEG images by using mode based first digit features," in *IEEE Workshop Multimedia Signal Processing*, 2008, pp. 730–735.

[7] Y.-L. Chen and C.-T. Hsu, "Image tampering detection by blocking periodicity analysis in JPEG compressed images," in *IEEE Workshop Multimedia Signal Processing*, 2008, pp. 803–808.

[8] T. Bianchi and A. Piva, "Detection of nonaligned double JPEG compression based on integer periodicity maps," *IEEE Trans. Inf. Forensics Security*, vol. 7, no. 2, pp. 842–848, 2012.

[9] X. Feng and G. Doërr, "JPEG recompression detection," in *IS&T/SPIE Electronic Imaging*, 2010, pp. 188–199.

[10] Y. Chen and C. Hsu, "Detecting recompression of JPEG images via periodicity analysis of compression artifacts for tampering detection," *IEEE Trans. Inf. Forensics Security*, vol. 6, no. 2, pp. 396–406, 2011.

[11] D. Fu, Y. Q. Shi, and W. Su, "A generalized Benford's law for JPEG coefficients and its applications in image forensics," in *Electronic Imaging*, 2007, pp. 574–584.

[12] S. Milani, M. Tagliasacchi, and S. Tubaro, "Discriminating multiple JPEG compression using first digit features," in *IEEE Int. Conf. Acoustics, Speech and Signal Processing*, 2012, pp. 2253–2256.

[13] I. Amerini, R. Becarelli, R. Caldelli, and A. D. Mastio, "Splicing forgeries localization through the use of first digit features," in *IEEE Int. Workshop Inf. Forensics and Security*, 2014, pp. 143–148.

[14] D. Cozzolino and L. Verdoliva, "Single-image splicing localization through autoencoder-based anomaly detection," in *IEEE Int. Workshop Inf. Forensics and Security*, 2016, pp. 1–6.

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

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

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

[18] M. Barni, E. Nowroozi, and B. Tondi, "Improving the security of image manipulation detection through one-and-a-half-class multiple classification," *Multimedia Tools and Appl.*, vol. 79, no. 3, pp. 2383–2408, 2020.

[19] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian approximation: Representing model uncertainty in deep learning," in *Int. Conf. Machine Learning*, 2016, pp. 1050–1059.

[20] B. Lakshminarayanan, A. Pritzel, and C. Blundell, "Simple and scalable predictive uncertainty estimation using deep ensembles," in *Advances in Neural Inf. Processing Syst.*, 2017, pp. 6402–6413.

[21] C. Blundell, J. Cornebise, K. Kavukcuoglu, and D. Wierstra, "Weight Uncertainty in Neural Network," in *Int. Conf. Machine Learning*, 2015, pp. 1613–1622.

[22] A. Kendall and Y. Gal, "What uncertainties do we need in Bayesian deep learning for computer vision?" in *Int. Conf. Neural Inf. Processing Syst.*, 2017, pp. 5580–5590.

[23] A. Maier, B. Lorch, and C. Riess, "Toward Reliable Models for Authenticating Multimedia Content: Detecting Resampling Artifacts With Bayesian Neural Networks," *arXiv e-prints*, p. arXiv:2007.14132, 2020.

[24] Y. Ovadia, E. Fertig, J. Ren, Z. Nado, D. Sculley, S. Nowozin, J. Dillon, B. Lakshminarayanan, and J. Snoek, "Can you trust your model's uncertainty? Evaluating predictive uncertainty under dataset shift," in *Advances in Neural Inf. Processing Syst.*, 2019, pp. 13 991–14 002.

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

[26] T. S. Jaakkola and M. I. Jordan, "Bayesian parameter estimation via variational methods," *Statistics and Computing*, vol. 10, no. 1, pp. 25–37, 2000.

[27] D.-T. Dang-Nguyen, C. Pasquini, V. Conotter, and G. Boato, "RAISE: A raw images dataset for digital image forensics," in *ACM Multimedia Syst. Conf.*, 2015, pp. 219–224.