ON CONSTRUCTING A BETTER CORRELATION PREDICTOR FOR PRNU-BASED IMAGE FORGERY LOCALIZATION

Xufeng Lin and Chang-Tsun Li**

*School of Information Technology, Deakin University,
Waurn Ponds Campus, Geelong, VIC 3216, Australia
{xufeng.lin, changtsun.li}@deakin.edu.au

2021 IEEE International Conference on Multimedia and Expo (ICME) | 978-1-6654-3864-3/20/$31.00 ©2021 IEEE | DOI: 10.1109/ICME51207.2021.9428127

ABSTRACT

Localizing image forgeries is one of the key topics in multimedia forensics. Among many image forgery localization techniques, the one based on the photo-response non-uniformity (PRNU) noise has attracted substantial attention because of its capability of localizing forgeries regardless of the type of forgery. However, despite the devoted efforts to improving the performance of PRNU-based forgery localization, there remain challenges to be overcome, especially for detecting subtle forgeries in PRNU-attenuated regions due to complex image content. In this work, we investigate the feasibility and effectiveness of convolutional neural networks (CNN) in predicting PRNU correlations under complex backgrounds for more accurate forgery localization. The experimental results on 20 cameras and 200 realistic forgery images show that significant improvement in correlation prediction and forgery localization can be achieved even with a light-weight CNN model. The robustness of different correlation predictors against JPEG compression is also evaluated.

Index Terms— Convolutional neural networks, image forgery localization, Photo Response Non Uniformity noise

1. INTRODUCTION

The ease of manipulating images with increasingly powerful image editing tools has led to the growing occurrences of digitally altered forgeries, which raise serious public concerns about the credibility of digital images. Therefore, exposing image forgeries becomes essential under the circumstances where digital images serve as critical evidence. Among various image forgery localization techniques, Photo Response Non-Uniformity (PRNU) noise has been proven to be an effective tool for exposing image forgeries [1, 2, 3, 4]. It presents in every image and is unique to the sensor that captures the image. Therefore, its absence can be used to signify forgeries in a questionable image, provided that the reference PRNU pattern of its source camera is available.

Lukas *et. al* [2] first proposed a PRNU-based technique for image forgery localization, which was later modeled and improved in [3] under a binary hypothesis-testing framework. Given the estimated PRNU of a suspicious image and the reference PRNU of its source camera, the basic idea is to compare the estimated and the reference PRNU patterns within a detection window sliding over the image. If the image within the detection window has been forged, the reference PRNU is supposed to be absent and the correlation, which serves as the decision statistic, between the estimated and the reference PRNU patterns within the detection window is expected to be lower than a predetermined threshold.

However, due to the attenuation of PRNU in low-intensity, highly textured or saturated image regions [3, 5], even the authentic pixels in such regions will tend to show low correlations, which give rise to false positives, i.e. mislabeling authentic pixels as forged. To alleviate this problem, most existing methods, e.g. [6, 7, 8], resort to a correlation predictor [3] based on hand-crafted image features to predict the expected correlation for authentic image regions. With the auxiliary information provided by the correlation predictor, an image region is considered to be forged only if its correlation with the reference PRNU is "too" low compared to the predicted correlation. Nevertheless, as pointed out in [7] and observed in our experiments, despite the explicit care for the unreliable regions in the feature design of the correlation predictor, conspicuous false positives still occur, which call for a more accurate and robust correlation predictor.

Recent years have witnessed a trend of abandoning the hand-crafted features and favoring deep neural networks that can automatically learn expressive features from data. Inspired by the great success of convolutional neural networks (CNN) in the tasks of image classification and regression [9, 10, 11], we investigate the feasibility and effectiveness of CNN in correlation prediction for enhancing the performance of PRNU-based forgery localization. The rest of this paper is organized as follows. In Section 2, we will revisit the binary hypothesis-testing framework of PRNU-based image forgery localization. In Section 3, we will analyze the performance of correlation predictors based on hand-crafted features and give the details of the proposed CNN predictor. Section 4 presents the experimental results and analysis. Finally, Section 5 concludes the work.

978-1-6654-3864-3/21/$31.00 ©2021 Crown

2. BACKGROUND

Given a questionable image $\boldsymbol{I}$ captured by a camera with reference PRNU $\boldsymbol{K}$, image forgeries are exposed by localizing the pixels where the reference PRNU is absent. To this end, a binary hypothesis test can be carried out for each pixel i in the image:

$$\begin{cases} H_0 : W_i = \Theta_i \\ H_1 : W_i = Z_i + \Theta_i \end{cases} \tag{1}$$

where W_i is the noise residual (i.e. approximated PRNU pattern) at pixel i, Z_i=$K_i I_i$ is the product of the reference PRNU and pixel intensity at pixel i, and Θ_i is PRNU-irrelevant random noise. Under hypothesis H_0, K_i is absent in noise residual W_i, which indicates that pixel i is forged, while under hypothesis H_1, pixel i is authentic.

Due to the weak noise-like nature of PRNU, its reliable detection at each pixel requires jointly processing a large number of pixels, so the forgery localization is typically carried out by sliding a detection window across the image to visit each pixel and analyze the PRNU within the window. To make a decision for pixel i, the normalized cross-correlation ρ_i is calculated over the pixels within the detection window centered at i:

$$\rho_i = \frac{\sum_{j\in\Omega_i}(W_j - \overline{W})(Z_j - \overline{Z})}{\sqrt{\sum_{j\in\Omega_i}(W_j - \overline{W})^2}\sqrt{\sum_{j\in\Omega_i}(Z_j - \overline{Z})^2}}, \tag{2}$$

where Ω_i is a detection window of size $s\times s$ centered at pixel i and the mean value of Z or W within Ω_i is denoted by a bar. ρ_i is used as the decision statistic to assess the likelihood that H_0 or H_1 is true, provided that the probability density functions $p(\rho_i|H_0)$ and $p(\rho_i|H_1)$ under H_0 and H_1 are available.

$p(\rho_i|H_0)$ can be estimated using images coming from cameras that are different from the source camera. However, the estimation of $p(\rho_i|H_1)$ is more challenging because the strength of PRNU in the noise residual is highly dependent on the image content. To model the effect of image content on ρ_i under hypothesis H_1, a correlation predictor based on local image features, namely image intensity f_I, texture f_T, signal flattening f_S, texture-intensity f_{TI} and their second-order terms, was proposed in [3]. More specifically, the correlation is modeled as the linear combination of the above 4 features and their second-order terms:

$$\boldsymbol{\rho} = \boldsymbol{F}\boldsymbol{\theta} + \boldsymbol{\Phi}, \tag{3}$$

where $\boldsymbol{\rho}$ is a $n\times 1$ vector consisting of the correlations calculated from n image blocks with Eq. (2), $\boldsymbol{F}$ is a $n\times 15$ matrix of the corresponding 4 features and their second-order terms (11 terms including 1 constant term), $\boldsymbol{\theta}$ is the parameters to be estimated, and $\boldsymbol{\Phi}$ is the modeling error. By applying the lease square estimation (LSE), $\boldsymbol{\theta}$ is estimated as

$$\hat{\boldsymbol{\theta}} = (\boldsymbol{F}^T\boldsymbol{F})^{-1}\boldsymbol{F}^T\boldsymbol{\rho}. \tag{4}$$

Having obtained $\hat{\boldsymbol{\theta}}$ under hypothesis H_1, the predicted correlation $\hat{\rho}_i$ for the pristine image block centered at pixel i can be estimated based on its image features $\boldsymbol{f}_i$:

$$\hat{\rho}_i = \boldsymbol{f}_i\hat{\boldsymbol{\theta}}. \tag{5}$$

Alternatively, as mentioned in [3], we can also train another correlation predictor by feeding the features $\boldsymbol{F}$ and $\boldsymbol{\rho}$ into a feed-forward network (FFN). In the rest of this paper, we will refer to these two correlation predictors based on the handcrafted features as LSE and FFN predictors, respectively.

Consequently, we model $p(\rho_i|H_1)$ as a Gaussian distribution with mean $\hat{\rho}_i$ and constant variance $\hat{\sigma}^2$=$Var(\boldsymbol{F}\hat{\boldsymbol{\theta}} - \boldsymbol{\rho})$ obtained from the training set. While for $p(\rho_i|H_0)$, it can be easily obtained by fitting a zero-mean Gaussian distribution with the correlations calculated under hypothesis H_0. With both $p(\rho_i|H_0)$ and $p(\rho_i|H_1)$, we can evaluate the probability of an image block being forged using the Bayes' rule

$$P_i = \frac{p(\rho_i|H_0)}{p(\rho_i|H_0) + p(\rho_i|H_1)}. \tag{6}$$

From the above analysis, we can see that the correlation predictor plays an important role in the decision-making as it provides information about what correlation value should be expected for an authentic image block. Thus, the performance of the correlation predictor will have a significant impact on the forgery localization in PRNU-attenuated regions.

3. CNN CORRELATION PREDICTOR

The output of correlation predictor conveys the information of what correlation should be expected for an authentic image block, thus the accurate prediction is essential for reducing false positives (i.e. authentic pixels are wrongly labeled as forged). In what follows, we will use two cameras in the Dresden dataset [12], namely FujiFilm FinePixJ50 and Nikon D70_1[1] as examples to highlight our observations on the performance of the LSE and FFN predictors. The scatter plots of the real and predicted correlations for the two cameras are shown in the first two rows of Fig. 1, where the corresponding adjusted R^2 coefficient for measuring the prediction accuracy is also displayed in the bottom right corner of each plot.

We categorize the "outliers" deviated far away from the red line into two types: *under-predicted outliers* and *over-predicted outliers*. The under-predicted outliers occur when the predicted correlations are far less than the real correlations, as circled in green in Fig. 1 for Nikon D70_1. Closer investigation revealed that these outliers are probably induced by some artifacts in the images (e.g. due to the stains or dust left on the lens or sensor of the camera). The over-predicted outliers arise when the predicted correlation values are much higher than the real correlations. One example is FujiFilm

[1]There are two Nikon D70 in the Dresden dataset. We used Nikon D70_0 to refer to the first one and Nikon D70_1 for the second one.

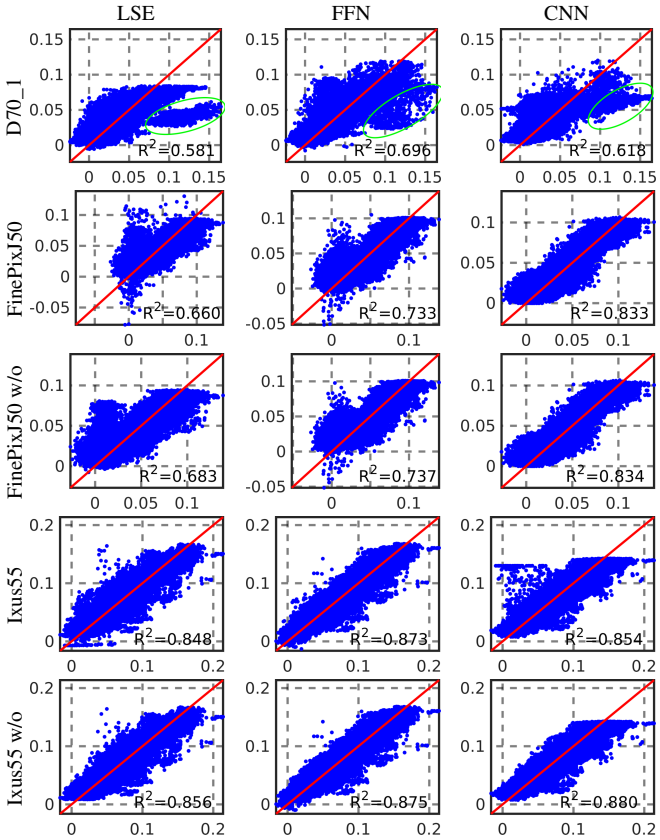


Fig. 1: Scatter plots of correlation predictions made by LSE (1st column), FFN (2nd column), and CNN (3rd column) predictors. 1st row: Nikon D70_1, 2nd row: FujiFilm FinePixJ50, 3rd row: FujiFilm FinePixJ50 w/o saturated blocks, 4th row: Canon Ixus55, 5th row: Canon Ixus55 w/o saturated blocks. x-axis: real correlations, y-axis: predicted correlations.

FinePixJ50, for which a bunch of points deviate from the red line, as shown in the second row of Fig. 1.

The reason why the predictor outputs over-predicted correlations can be complicated. We noticed that the majority of these over-predicted outliers correspond to unsaturated and visually "normal" image blocks. To see this, we removed all image blocks with a saturation measurement $\mathcal{S}_i \geq 0.7$ and showed the scatter plots for FujiFilm FinePixJ50 in the 3rd row of Fig. 1. $\mathcal{S}_i \in [0, 1]$ is defined as

$$\mathcal{S}_i = 1 - \frac{1}{d} \sum_{j \in \Omega_i} Q_j, \tag{7}$$

where d is the pixel number in an image block and Q_j is the attenuation coefficient defined in a similar way as the attenuation function in [3]:

$$Q_j = \begin{cases} e^{-(I_j - 240)^2/6}, & I_j > 240 \\ 1, & I_j \leq 240 \end{cases} \tag{8}$$

Thus, $\mathcal{S}_i$ measures the percentage of saturated pixels within an image block. As can be observed, most over-predicted outliers still persist even after the removal of saturated blocks.

From the above examples, we can see that the hand-crafted features proposed in [3] are incapable of modeling the complex relationship between the image content and PRNU correlations. Motivated by the great success of CNN in image regression tasks, such as object pose estimation [10] and apparent age estimation [11], we design a light-weight CNN model for the task of correlation prediction. As illustrated in Fig. 2, the proposed network takes RGB image blocks as input and returns a scalar value representing the predicted correlation. The feature extraction consists of three repeated network structures, each of which contains two consecutive convolutional layers followed by an average pooling layer with a kernel size of 2×2. The stride and padding of all convolution operations are set to 1 and 0, respectively. The number of feature maps is increased from 6 to 36 as the network goes deeper. The extracted features are mapped to a fully connected layer with 192 neurons and a output layer with 1 neuron.

Once the network has been trained, it is able to predict a correlation value for each image block fed into it. However, if we want to predict the dense pixel-wise correlation map, we need to run the model for every pixel, which can be computationally expensive for images of large size. For the LSE and FFN predictors based on hand-crafted features, the pixel-wise local features can be efficiently extracted in a sliding window manner using fast 2D convolution. Actually, a similar approach can be applied for convolutional layers because two adjacent blocks share a large portion of common computations. We thus computed the convolution only once for the entire input image or the feature maps generated at intermediate convolutional layers. The feature maps for blocks at different spatial locations are contained in the resulting "extended" feature map. For the last two fully connected layers of the proposed network, we replaced them with two convolutional layers with kernels of 1×1 spatial extent [13]. For the average pooling layers, we adopted the "fragmenting" method [14] by applying the average-pooling operation to the "extended" feature maps at four different offsets (each corresponding to one fragment) and reconstructing all the fragments to form a full-resolution dense correlation map.

4. EXPERIMENTS

4.1. Experimental Setup

Our experiments involved 20 cameras, 10 from the Dresden dataset [12] and 10 from the VISION dataset [15]. The information about the 20 cameras is summarized in Table 1. To estimate the PRNU pattern, we extracted the noise residuals from three channels of each image and converted them into a gray version of noise residual. For the training of

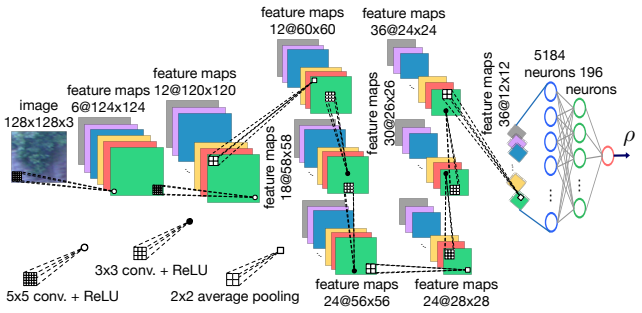


Fig. 2: The proposed CNN correlation predictor.

Table 1: Cameras used in our experiments

Dresden	VISION
C01 (Canon Ixus55)	D01 (Galaxy S3 Mini)
C02 (Canon Ixus70)	D02 (Iphone 4s)
C03 (Fujifilm FinePixJ50)	D03 (Huawei P9)
C04 (Nikon D70_0)	D04 (LG D290)
C05 (Nikon D70_1)	D05 (Iphone 5c)
C06 (Nikon D200)	D06 (Iphone 6)
C07 (Olympus mju 1050SW)	D07 (Lenovo P70-A)
C08 (Pentax OptioA40)	D08 (Galaxy Tab3)
C09 (Pentax OptioW60)	D09 (Iphone 4)
C10 (Sony DSC-H50)	D10 (Iphone 4s)

correlation predictors under hypothesis H_1, we randomly selected 50 natural images and generated 1000 image blocks of 128×128 px from each image, which result in a training set of 50000 blocks for each camera. Note that our experimental results will be based on the commonly used image block size 128×128 px, which gives a good balance between accuracy and localization.

4.2. Comparison of Prediction Performance

We trained the proposed CNN network for 100 epochs using stochastic gradient descent (SGD) with a momentum of 0.9. As the output of the network is a continuous value, we used mean square error (MSE) as the loss function for network training. For each camera, 70% image blocks were used for training and the remaining 30% blocks were used for validation. We set the initial learning rate $lr = 1e{-}5$ and applied the ReduceLROnPlateau scheduler to gradually reduce lr to $1e{-}7$ once the average MSE on the validation set does not drop in two consecutive epochs.

We first evaluated the proposed CNN predictor in comparison with the LSE and FFN predictors based on hand-crafted features [3]. The scatter plots of the CNN predictor for the above-mentioned Nikon D70_1 and FujiFilm FinePixJ50 are shown in the third column of Fig. 1. As we can see, with

Table 2: Average adjusted R^2 for different predictors.

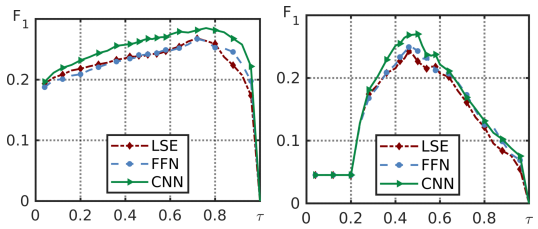
	Dresden				VISION		
	LSE	FFN	CNN		LSE	FFN	CNN
C01	0.848	**0.873**	0.854	D01	0.772	0.840	**0.891**
C02	0.782	0.836	**0.916**	D02	0.800	0.829	**0.858**
C03	0.660	0.733	**0.833**	D03	0.071	0.317	**0.526**
C04	0.716	0.794	**0.795**	D04	0.527	0.640	**0.688**
C05	0.581	**0.696**	0.618	D05	0.587	**0.743**	0.711
C06	0.778	**0.872**	0.808	D06	0.455	0.623	**0.818**
C07	0.796	0.845	**0.885**	D07	0.605	0.722	**0.843**
C08	0.708	0.758	**0.846**	D08	0.448	0.632	**0.840**
C09	0.756	0.825	**0.910**	D09	0.627	0.709	**0.844**
C10	0.843	0.884	**0.931**	D10	0.809	0.882	**0.899**
Avg.	0.747	0.812	**0.840**	Avg.	0.570	0.694	**0.792**

the features learned by CNN, most of the points are more compactly clustered along the red line, which imply a higher prediction accuracy. Note that for FujiFilm FinePixJ50, the over-predicted outliers have been largely eliminated by the CNN predictor. However, we also noticed that the CNN predictor may also produce over-predicted "outliers" for some cameras, e.g. Canon Ixus50 in the Dresden dataset (see the 4th row of Fig. 1). If we remove the image blocks with a saturation measure $\mathcal{S}_i \geq 0.7$ and showed the scatter plots again in the 5th row of Fig. 1, we can see that the majority of the over-predicted outliers have been removed. This indicates that the over-predicted outliers produced by the CNN predictor can be mostly attributed to saturated blocks. Since saturated pixels can be determined beforehand, the over-predicted errors of the CNN predictor can be easily corrected by taking special care for saturated pixels. By contrast, the LSE and FFN predictors may generate over-predicted correlations even for unsaturated and seemingly normal blocks. Such behavior makes them less predictable and more prone to prediction errors. More details about the average adjusted R^2 coefficients for the three predictors on the 20 cameras are summarized in Table 2. As can be seen, the CNN predictor outperforms the LSE and FFN predictors by a large margin especially for the 10 cameras in the VISION dataset, with the average R^2 coefficient increased from 0.570 for LSE and 0.694 for FFN to 0.792 for CNN.

4.3. Evaluation on Realistic Forgeries

We also compared the performance of LSE, FFN and CNN predictors in localizing realistic image forgeries. We manually generated 200 forged images (10 for each of the 20 cameras) using Photoshop. Different tools in Photoshop such as transformation, content-aware filling, and stamp clone, were used to generate various types of forgeries, e.g. texture replacement, object insertion, and object removal. The forg-

eries are of various sizes ranging from a few thousand to a few million pixels. Some example forgery localization results can be found in Fig. 4. It is worth mentioning that we did not use the Realistic Tampering Dataset provided by the authors of [7] because the training set for constructing the correlation predictor is unavailable.



(a) Average F_1 of STM (b) Average F_1 of CRF

Fig. 3: Comparison of average F_1 scores on 200 forged images when different correlation predictors and localization algorithms are used.

Base on the forgery probability map $\boldsymbol{P}$ obtained with Eq. (6), we adopted two different algorithms to localize the forgeries:

1. Simple thresholding with morphological post-processing (STM). Any pixel with a forgery probability higher than a threshold τ is considered as forged. The binary forgery map will be further post-processed by removing the connected forged regions that contain fewer than 64×64 px and applying a dilation operation with a disk kernel of 20×20 px.

2. Conditional random field (CRF) approach [7]. This algorithm models the neighborhood interactions with a conditional random field and encourages spatially-smooth results. We used the same parameters as provided in the source code of [7], i.e. tampering penalty $\alpha=-1$, content independent and adaptive neighborhood interaction penalties ($\beta_0=0.5, \beta_1=5.6$), and pixel similarity attenuation $\phi=25$.

For CRF, we applied the data term generalization [7]

$$E(P_i,\tau)=\begin{cases}\max(0.001, 1-\frac{P_i}{2\tau}), & P_i \text{ is authentic}\\ \max(0.001, 1-\frac{1-P_i}{2(1-\tau)}), & P_i \text{ is forged}\end{cases} \tag{9}$$

to ensure the equal potential for both decisions. This generalization also provides an opportunity to evaluate the localization performance under different thresholds $\tau\in(0,1)$. To reduce the running time of CRF, we applied a subsampling of 8×8 for each image in the optimization phrase, but we still used the full-size images for STM.

We evaluated the localization performance by calculating the average F_1 score over the $M=200$ images in the dataset:

$$F_1(\tau)=\frac{1}{M}\sum_{m=1}^{M}\frac{2\cdot\mathcal{P}_m(\tau)}{2\cdot\mathcal{P}_m(\tau)+\mathcal{N}_m(\tau)+\mathcal{F}_m(\tau)}, \tag{10}$$

where $\mathcal{P}_m(\tau)$, $\mathcal{N}_m(\tau)$, and $\mathcal{F}_m(\tau)$ are, respectively, the true positives, false negatives, and false positives associated with a given threshold τ for image m.

We varied τ from 0 to 1 and showed the average F_1 scores for different localization algorithms and correlation predictors in Fig. 3. As can be seen, LSE and FNN deliver comparable localization performance, but CNN clearly outperforms LSE and FNN when combined with both STM and CRF. Some examples of forgery localization in Fig. 4 show that CNN is able to make more accurate predictions in unreliable regions, e.g. the saturated regions in the 2nd and 4th rows and the low-intensity and textured regions in the 1st and 3rd rows. This effectively helps to reduce the false positives in the final decision maps.

4.4. Robustness Evaluation

While deep neural networks are able to achieve superior prediction accuracy, recent studies have also suggested that they are highly vulnerable to subtle perturbations in the images. Such phenomenon is expected to substantially affect the behavior of the CNN predictor under JPEG compression. To investigate the impact of JPEG compression on the correlation prediction and forgery localization, we generated 6 new versions of each forgery image for JPEG quality factors 100, 95, 90, 85, 80 and 75, and showed the average F_1 scores obtained with CRF in Fig. 5. We can observe that as the image quality decreases, the performance of CNN declines more rapidly than that of LSE and FNN. This indicates that the high prediction accuracy of CNN predictor might be at the cost of robustness because the CNN predictor relies more on features that can not survive JPEG compression.

One remedy for improving the robustness against JPEG compression is to train separate predictors for different quality levels. This incurs more training time but enforces the CNN predictor to learn robust feature representations from compressed images. We showed in Fig. 6 the localization results after retraining separate predictors for different quality levels. Compared to Fig. 5, we can see that retraining is effective in enhancing the robustness for all three predictors and it also helps to maintain the advantageous position for CNN predictor.

5. CONCLUSIONS

In this work, we proposed a CNN correlation predictor to improve the performance of PRNU-based image forgery localization. Compared to the correlation predictors based on hand-crafted features, the CNN predictor has been shown to give more accurate predictions and tends to be more robust against JPEG compression if separate predictors are trained for different compression quality levels, thus facilitating better localization performance as shown by the localization results on 200 realistic forgery images.

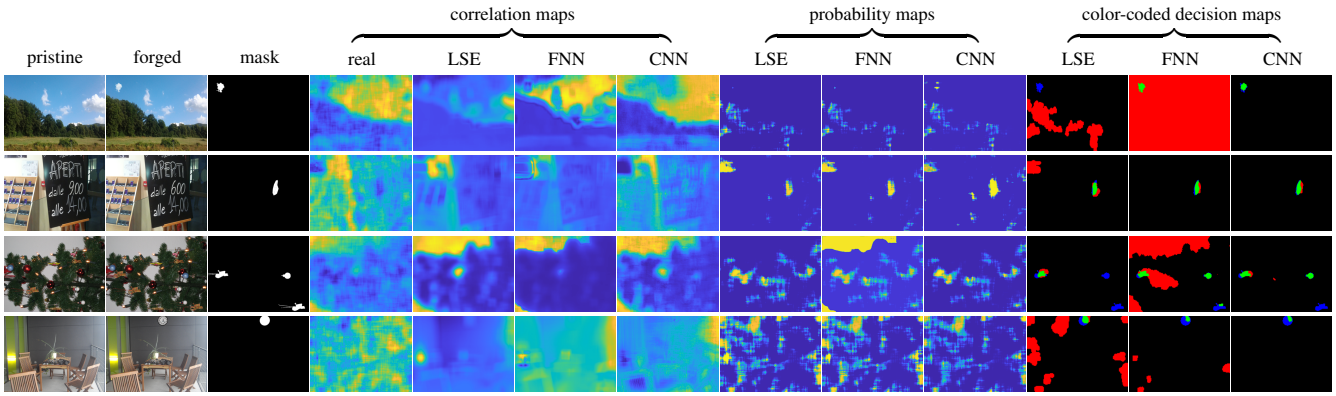
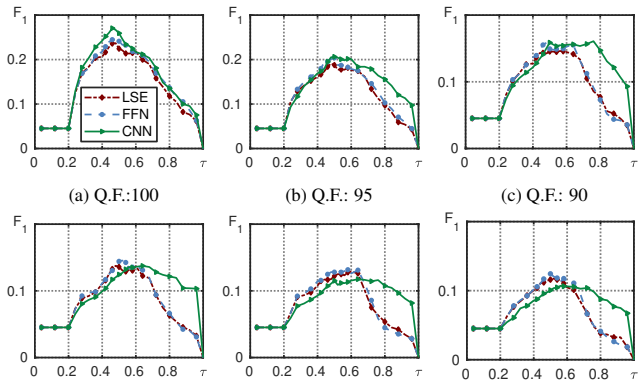


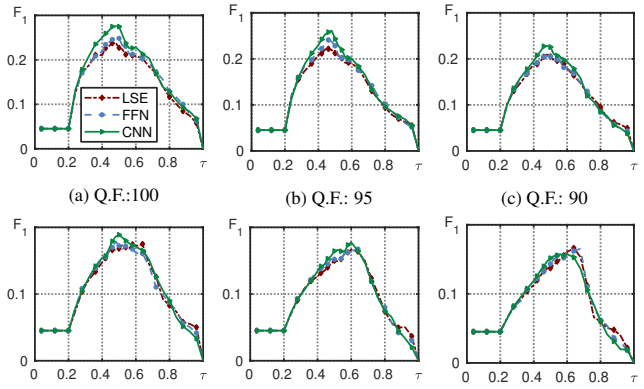
Fig. 4: Example forgery localization results. Color coding: *green*: detected forged regions ($\mathcal{P}$); *red*: detected authentic regions ($\mathcal{F}$); *blue*: missed forged regions ($\mathcal{N}$).



(a) Q.F.:100 (b) Q.F.: 95 (c) Q.F.: 90

(d) Q.F.: 85 (e) Q.F.: 80 (f) Q.F.: 75

Fig. 5: Evaluation of robustness of predictors against JPEG compression without retraining.



(a) Q.F.:100 (b) Q.F.: 95 (c) Q.F.: 90

(d) Q.F.: 85 (e) Q.F.: 80 (f) Q.F.: 75

Fig. 6: Evaluation of robustness of predictors against JPEG compression with retraining.

6. REFERENCES

[1] J. Lukáš, J. Fridrich, and M. Goljan, "Digital camera identification from sensor pattern noise," *IEEE Trans. Inf. Forensics Security*, vol. 1, no. 2, pp. 205–214, 2006.

[2] J. Lukáš, J. Fridrich, and M. Goljan, "Detecting digital image forgeries using sensor pattern noise," in *J. Electron. Imaging*. Int. Society for Optics and Photonics, 2006, pp. 60720Y–60720Y.

[3] M. Chen, J. Fridrich, M. Goljan, and J. Lukás, "Determining image origin and integrity using sensor noise," *IEEE Trans. Inf. Forensics Security*, vol. 3, no. 1, pp. 74–90, 2008.

[4] C.-T. Li and Y. Li, "Color-decoupled photo response non-uniformity for digital image forensics," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 2, pp. 260–271, 2011.

[5] C.-T. Li, "Source camera identification using enhanced sensor pattern noise," *IEEE Trans. Inf. Forensics Security*, vol. 5, no. 2, pp. 280–287, 2010.

[6] G. Chierchia, G. Poggi, C. Sansone, and L. Verdoliva, "A Bayesian-MRF Approach for PRNU-Based Image Forgery Detection," *IEEE Trans. Inf. Forensics Security*, vol. 9, no. 4, pp. 554–567, Apr 2014.

[7] P. Korus and J. Huang, "Multi-Scale Analysis Strategies in PRNU-Based Tampering Localization," *IEEE Trans. Inf. Forensics Security*, vol. 12, no. 4, pp. 809–824, Apr 2017.

[8] X. Lin and C.-T. Li, "PRNU-based content forgery localization augmented with image segmentation," *IEEE Access*, vol. 8, pp. 222645–222659, 2020.

[9] I. Amerini, C.-T. Li, and R. Caldelli, "Social network identification through image classification with cnn," *IEEE Access*, vol. 7, pp. 35264–35273, 2019.

[10] Y. Xiang, T. Schmidt, V. Narayanan, and D. Fox, "PoseCNN: A convolutional neural network for 6d object pose estimation in cluttered scenes," *arXiv preprint arXiv:1711.00199*, 2017.

[11] R. Rothe, R. Timofte, and L. Van Gool, "Dex: Deep expectation of apparent age from a single image," in *Proc. the IEEE Int. Conf. Computer Vision Workshops*, 2015, pp. 10–15.

[12] T. Gloe and R. Böhme, "The Dresden image database for benchmarking digital image forensics," *J. Digit. Forensic Practice*, vol. 3, no. 2-4, pp. 150–159, 2010.

[13] P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun, "Overfeat: Integrated recognition, localization and detection using convolutional networks," *arXiv preprint arXiv:1312.6229*, 2013.

[14] A. Giusti, D. C Ciresan, J. Masci, L. M Gambardella, and J. Schmidhuber, "Fast image scanning with deep max-pooling convolutional neural networks," in *Proc. Int. Conf. Image Process.*, 2013, pp. 4034–4038.

[15] D. Shullani, M. Fontani, M. Iuliani, O. Al Shaya, and A. Piva, "VISION: a video and image dataset for source identification," *EURASIP J. Information Security*, vol. 2017, no. 1, pp. 15, Oct 2017.