

# Camera Identification task using VDNet and VDID

## Elaborazione e Protezione delle Immagini

Abdullah Chaudhry

Matricola: 7042261

Department of Information Engineering  
University of Florence  
Via di Santa Marta 3, Florence, Italy  
[abdullah.chaudhry@stud.unifi.it](mailto:abdullah.chaudhry@stud.unifi.it)

Simone Pezzulla

Matricola: 7072831

Department of Information Engineering  
University of Florence  
Via di Santa Marta 3, Florence, Italy  
[simone.pezzulla@stud.unifi.it](mailto:simone.pezzulla@stud.unifi.it)

**Abstract**—In this report, different algorithm are used in order to perform digital camera identification task. We compare the original implementation [2] and try to replace the original noise extractor algorithm with two Bayesian frameworks. The comparison is taken using VISION dataset[7]. In particular we used natural captured images from sensor and social network processed images.

**Availability and implementation**—Example code is available at <https://github.com/daikon899/PRNU>.

**Keywords**—PRNU, denoiser, VDID, VDNet, identification

## I. INTRODUCTION

As explained in [5], digital images and video continue to replace their analog counterparts, the importance of reliable, inexpensive and fast identification of digital image origin will only increase. Reliable identification of the device used to acquire a particular digital image would especially prove useful, for example, in the court for establishing the origin of images presented as evidence.

The problem of digital image authentication was addressed with different approaches from the simpler one that involves studying the information contained in the image header and any attachments to more complex solutions such watermarks inside the image itself [3][1] which contains information about the camera. Unfortunately, only relatively expensive cameras (DSLR) support this solution. In [8] they proposed supervised learning method where each image is represented using a vector of numerical features extracted from it. A multiclass support vector machine (SVM) classifier is then trained to classify images from five different cameras. The correct classification obtained 95% in the best case. Forensic applications in the court are likely to require a substantially higher level of accuracy.

Finally, the camera identification method based on a sensor's pattern noise implemented in [2] gives significantly

more reliable results compared to state-of-the-art approaches. In this report we propose two new methods and compare them with the original.

In the next Section, we give a description of the 2 methods proposed in this report for noise extraction in order to perform a camera identification task. In particular, in Section III, we focus on the properties of VDNet[11]. We repeat the same process for VDID[4] in Section IV. We briefly discuss in Section V about the dataset we used. In Section VI we discuss the implementation details adopted and show the results in section VI-B. Finally, the report concludes in section VII with some remarks. At the end of this report are the tables with the complete results.

## II. RESEARCH ACTIVITY

This project is based on the work of [2], the goal is to extract sensor's fingerprint from the captured image and compare this reference with other query fingerprints in order to perform a camera identification task. As suggested in [2], they use Photo-Response Non-Uniformity (PRNU) extracted from flat and not saturated images as a unique fingerprint of digital camera and use Peak-correlation-to-correlation-ratio (PCE) for the identification task. We use different algorithms in order to extract and compare digital camera fingerprints. In this report we extend the implementation provided in [2] by adding 2 new noise extraction methods discussed in the following sections. Later we will also show results on images processed from social networks such as Facebook and Whatsapp. Finally, our last experiments are on images processed by few recent Huawei devices [9].

All the code is implemented in Python 3. We want to thank the LESC lab for providing us the computers to run the experiments. The computers used the Ubuntu 20.04.2 LTS

operating system, with 128GB of RAM, it had an Intel(R) Core(TM) i9-7940X CPU @3.10GHz, one Nvidia Quadro P6000 with 24GB memory and four GeForce RTX 2080 with 12GB each.

### III. VDNET

VDNet uses a variational inference for non-iid real-noise estimation and image denoising in a unique Bayesian Network. Specifically, an approximate posterior, parameterized by deep neural networks, is presented by taking the intrinsic clean image and noise variances as latent variables conditioned on the input noisy image.

#### A. Objective function

Given the clean latent image  $z_j$ , the noise level  $\sigma_j$ , the noisy image  $y_j$ , the objective function is defined as follows:

$$\min_{W_D, W_S} - \sum_{j=1}^n \mathcal{L}(z_j, \sigma_j^2; y_j)$$

where  $\mathcal{L}(z_j, \sigma_j^2; y_j)$  can be expressend in functions of  $\mu$ ,  $m^2$ ,  $\alpha$ ,  $\beta$ , i.e. the parameters that approximate the posterior of  $z_j$  and  $\sigma_j$ .

#### B. Network Architecture

The Network is composed by two parts:

- D-Net: a U-Net [10] with depth 4, which contains 4 encoder blocks, 3 decoder blocks, and symmetric skip connection under each scale in order to capture multi-scale information of the image
- S-Net: a DnCNN [6] architecture with five layers, and the feature channels of each layer is set as 64.

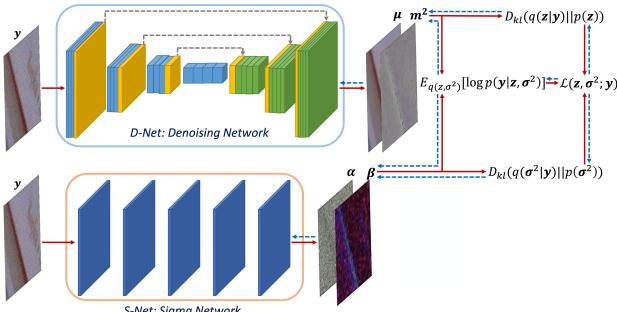


Figure 1: The overall architecture of VDNet

### IV. VDID

Variational Deep Image Denoise (VDID) is a bayesian framework that can handle blind scenarios based on the variational approximation of objective functions separating the complicated problem into simpler ones. Its main characteristics are:

- It handles **both AWGN and real-world noise**
- Trained in an end-to-end scheme without any additional noise information
- Requires fewer parameters than state-of-the-art denoisers

The objective is formulated in terms of **maximum a posterior** (MAP) inference. An approximated form of the objective is calculated by introducing a latent variable based on variational Bayes which incorporates the underlying noisy image distribution. Based on the latent space, VDID can focus on simpler subdistributions of the original problem.

#### A. VDID Architecture

The Denoiser is fully convolutional neural network adopting ResBlock as the basic building block. The last convolution layer infers the residual image (noise). The Autoencoder is a feedforward convolutional network with a reparametrization trick.

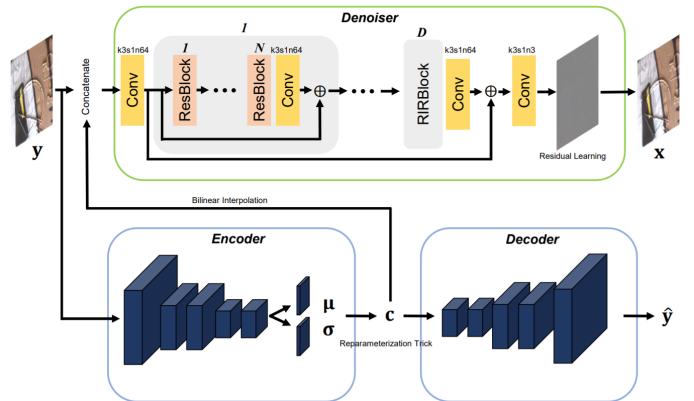


Figure 2: VDID architecture

#### B. VDID loss

The loss is composed by 3 terms:

- 1) **Mean absolute error (MAE)** between the ground-truth clean image and the inferred output to minimize the distortion  

$$\mathcal{L}_{denoise} = \|x - \hat{x}\|_1$$
- 2) The **KL divergence** between the posterior and the prior. The latent variable  $c$  contains an abstract of the noisy image distribution  

$$D_{KL}(q(c|y) || p(c))$$

- 3) The third term is composed by

- **MAE** between the **noisy input image** and the **decoder's output**
- **Adversarial loss [advloss]**: to better learn the noisy image distribution
- **Noise estimation loss**:  $\sigma_N$  is the noise level and  $EST(\cdot)$  is a simple two-layer CNN.

$$\mathcal{L}_{recon} = \|y - \hat{y}\|_1 + \lambda_1 \mathcal{L}_{adv} + \lambda_2 \|\sigma_N - EST(c)\|_1$$

The overall loss is the sum of three terms aforementioned. Where the first term is just a naive approach totally relying on the discriminative power of CNN. In the second term the KL divergence forces  $c$  to have discriminative power over observed noisy images  $y$ . Finally, the third term forces  $c$  to include the information on a noisy image.

## V. DATASET

For the experiments we used the VISION dataset provided by LESC laboratory [7]. It contains about 30 devices and for each device there are 50 flat and not saturated images from which the reference fingerprints are extracted. Each of them is then compared with 20 natural image query fingerprints from all devices (Figure 3 shows the subdirectory structure for a device).

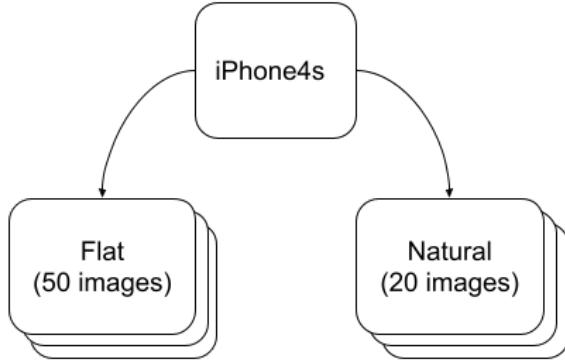


Figure 3: iPhone 4s subdirectory example

## VI. EXPERIMENTS

For implementation settings we only used horizontal images and discarded the vertical ones. For each device we chose randomly 50 flat images and 20 natural images. The experiments were conducted initially on 9 different devices: iPhone 4, iPhone 4s, iPhone 5c, Samsung Galaxy S3, Samsung Galaxy Tab 3, Samsung Galaxy S3, Samsung Galaxy S5, Wiko Ridge 4G and OnePlus A3000.

Later, we used all the devices available and we also compared pictures for each device processed through Facebook, WhatsApp and Huawei.

### A. Implementation details

The experiments were carried out in four different configurations. In particular, the first configuration is unaltered compared to the original implementation, while the following three are slight modifications after the noise extraction:

- 1) Original implementation
- 2) Removing zero mean normalization from the original implementation
- 3) Removing Wiener filter from the original implementation
- 4) Removing both zero mean normalization and Wiener filter from the original implementation

For each strategy we compared the results obtained with original noise extraction implementation with those obtained using VDNet and VDID as noise extractor. Note that we used the pre-trained weights provided on the respective networks' github repository.

	Original		VDNet		VDID	
	TPR	FPR	TPR	FPR	TPR	FPR
Apple_iPhone4s	1	0	1	0	1	0
Apple_iPhone5c	0.9	0	0.85	0	0.6	0
Samsung_GalaxyTab3	0.85	0	0.85	0	0.75	0
Apple_iPhone4	1	0	1	0	1	0
Samsung_GalaxyS3	1	0	1	0	0.85	0
Sony_XperiaZ1Compact	1	0	1	0	1	0
Wiko_Ridge4G	1	0	1	0	1	0
OnePlus_A3000	0.7	0	0.7	0	0.6	0
Samsung_GalaxyS5	1	0	1	0	1	0
AUC on CC	0.93		0.93		0.93	
AUC on PCE	0.99		0.98		0.96	

(a) Original implementation

	Original		VDNet		VDID	
	TPR	FPR	TPR	FPR	TPR	FPR
Apple_iPhone4s	1	0	1	0	1	0
Apple_iPhone5c	0.9	0	0.85	0	0.7	0
Samsung_GalaxyTab3	0.85	0	0.85	0	0.8	0
Apple_iPhone4	1	0	1	0	1	0
Samsung_GalaxyS3	1	0	1	0.01	0.85	0
Sony_XperiaZ1Compact	1	0	1	0	1	0
Wiko_Ridge4G	1	0	1	0	1	0
OnePlus_A3000	0.7	0	0.7	0	0.65	0
Samsung_GalaxyS5	1	0	1	0	1	0
AUC on CC	0.93		0.93		0.93	
AUC on PCE	0.98		0.97		0.96	

(b) Removing zero mean normalization

	Original		VDNet		VDID	
	TPR	FPR	TPR	FPR	TPR	FPR
Apple_iPhone4s	1	0	0.95	0	1	0
Apple_iPhone5c	0.9	0	0.9	0	0.7	0
Samsung_GalaxyTab3	0.9	0	0.9	0	0.6	0
Apple_iPhone4	1	0	1	0	0.4	0
Samsung_GalaxyS3	1	0	0.85	0	0.85	0
Sony_XperiaZ1Compact	1	0	1	0	1	0
Wiko_Ridge4G	1	0	1	0	1	0
OnePlus_A3000	0.8	0	0.7	0	0.6	0
Samsung_GalaxyS5	1	0	0.95	0	1	0
AUC on CC	0.95		0.94		0.94	
AUC on PCE	0.99		0.98		0.93	

(c) Removing Wiener filter

	Original		VDNet		VDID	
	TPR	FPR	TPR	FPR	TPR	FPR
Apple_iPhone4s	1	0	0.85	0	0.4	0
Apple_iPhone5c	0.9	0	0.9	0	0	0
Samsung_GalaxyTab3	0.75	0	0.5	0	0	0
Apple_iPhone4	1	0	1	0	0	0
Samsung_GalaxyS3	1	0	0.6	0	0	0
Sony_XperiaZ1Compact	1	0	0.95	0	0	0
Wiko_Ridge4G	1	0	1	0	0.45	0
OnePlus_A3000	0.8	0	0.35	0	0	0
Samsung_GalaxyS5	1	0	0.95	0	0.55	0
AUC on CC	0.95		0.77		0.52	
AUC on PCE	0.99		0.94		0.68	

(d) Removing both Wiener and zero mean

Figure 4: TPR and FPR comparison using VISION's natural images (only 9 devices)

We measured PCR, script run time, TPR and FPR scores in order to evaluate the goodness of each strategy.

### B. Results

Analyzing natural images, as shown in Figure 4, the original method remains the most robust in all the different configurations. Regarding the additional implementation, VDNet has also good performances, specially in the 1st configuration (Figure 4a). The VDID, on the other hand, obtains rather poor results in all the strategies adopted.

Removing zero mean normalization (Figure 4b) gives better results compared to the previous configuration. VDID gets slightly better results even though it still remains the worst choice. Note that this configuration introduces non-negative FPR values for some cases. In the third configuration VD-

Net and VDID gets slightly worse results by removing the Wiener filter. Finally, removing both zero mean normalization and wiener filter in the fourth configuration (Figure 4d) we obtained really poor performances (the original method is still robust).

Overall we can see that the original method using the first three configurations is the most efficient strategy getting TPR values really close to 1 almost for every device. VDNet also achieves similar numerical results to the original implementation in the first two configurations. VDID is a more recent denoiser but it seems to be not a good method for fingerprint extraction in order to perform a camera identification task. We report here average TPR values for these four configurations (see complete results in Figure 7):

- 1) original: 0.83, VDNet: 0.82, VDID: 0.74
- 2) original: 0.83, VDNet: 0.83, VDID: 0.76
- 3) original: 0.83, VDNet: 0.73, VDID: 0.71
- 4) original: 0.82, VDNet: 0.70, VDID: 0.14

The next experiments we performed consisted in using images processed by social networks such as Facebook and WhatsApp. In this case we got really poor results compared to the previous dataset's images. In Figure 5 we compared results for images processed by Facebook: from all devices only the Samsung Galaxy Tab 3 produced a TPR greater than 0. For this device in particular the original method and VDNet perform quite similarly as before (in the first three configurations). Figure 5b shows a slightly higher TPR in the second configuration. WhatsApp processed images gives really poor performances, as shown in Figure 6, with TPR values equal to 0 for each model and for each configuration.

As discussed in [9], with the advent of computational photography, most recent devices heavily process the acquired pixels, possibly introducing non-unique artifacts that may reduce PRNU noise's distinctiveness. Therefore, in the last experiments we have also treated images processed by the latest Huawei devices. As expected, there are many more non-negative FPR values as seen in Figure 10. The high FPR values are not a problem of a single device but of all the devices of the same model that heavily process the acquired image, as stated in [9]. All three denoisers are affected by this problem. Only in the last 2 configurations the FPR values decrease (when we remove the Wiener filter). As always, the original implementation is the most robust, VDNet gets similar values (only in the first configuration in this case) and VDID is the worse. Note the very high PCE values in all 4 configurations. For each configuration we annotated the script run time: the original method and VDNet are executed in few minutes, VDID on the other hand takes a few hours. Comparing the four strategies in all our experiments we can see that removing the zero mean normalization or the Wiener filter which give the best results in some cases can also save a small amount of time.

## VII. CONCLUSION AND FUTURE DEVELOPMENTS

In this report we faced the problem of camera identification task from its images based on the sensor's noise pattern. We

	Original		VDNet		VDID	
	TPR	FPR	TPR	FPR	TPR	FPR
Apple_iPhone4s	0	0	0	0	0	0
Apple_iPhone5c	0	0	0	0	0	0
Samsung_GalaxyTab3	0.6	0	0.6	0	0.15	0
Apple_iPhone4	0	0	0	0	0	0
Samsung_GalaxyS3	0	0	0	0	0	0
Sony_XperiaZ1Compact	0	0	0	0	0	0
Wiko_Ridge4G	0	0	0	0	0	0
OnePlus_A3000	0	0	0	0	0	0
Samsung_GalaxyS5	0	0	0	0	0	0
AUC on CC	<b>0.58</b>		<b>0.58</b>		0.56	
AUC on PCE	0.52		0.55		<b>0.57</b>	
Time (s)	<b>191</b>		275		6693	

	Original implementation		VDNet		VDID	
	TPR	FPR	TPR	FPR	TPR	FPR
Apple_iPhone4s	0	0	0	0	0	0
Apple_iPhone5c	0	0	0	0	0	0
Samsung_GalaxyTab3	0.6	0	0.65	0	0.25	0
Apple_iPhone4	0	0	0	0	0	0
Samsung_GalaxyS3	0	0	0	0	0	0
Sony_XperiaZ1Compact	0	0	0	0	0	0
Wiko_Ridge4G	0	0	0	0	0	0
OnePlus_A3000	0	0	0	0	0	0
Samsung_GalaxyS5	0	0	0	0	0	0
AUC on CC	<b>0.59</b>		0.58		0.57	
AUC on PCE	0.52		0.55		<b>0.57</b>	
Time (s)	<b>184</b>		214		6416	

	Removing zero mean normalization		VDNet		VDID	
	TPR	FPR	TPR	FPR	TPR	FPR
Apple_iPhone4s	0	0	0	0	0	0
Apple_iPhone5c	0	0	0	0	0	0
Samsung_GalaxyTab3	0.7	0	0.5	0	0.15	0
Apple_iPhone4	0	0	0	0	0	0
Samsung_GalaxyS3	0	0	0	0	0	0
Sony_XperiaZ1Compact	0	0	0	0	0	0
Wiko_Ridge4G	0	0	0	0	0	0
OnePlus_A3000	0	0	0	0	0	0
Samsung_GalaxyS5	0	0	0	0	0	0
AUC on CC	<b>0.58</b>		0.58		0.56	
AUC on PCE	0.51		<b>0.57</b>		0.55	
Time (s)	<b>173</b>		203		6550	

	Removing Wiener filter		VDNet		VDID	
	TPR	FPR	TPR	FPR	TPR	FPR
Apple_iPhone4s	0	0	0	0	0	0
Apple_iPhone5c	0	0	0	0	0	0
Samsung_GalaxyTab3	0.5	0	0.2	0	0	0
Apple_iPhone4	0	0	0	0	0	0
Samsung_GalaxyS3	0	0	0	0	0	0
Sony_XperiaZ1Compact	0	0	0	0	0	0
Wiko_Ridge4G	0	0	0	0	0	0
OnePlus_A3000	0	0	0.05	0	0	0
Samsung_GalaxyS5	0	0	0	0	0	0
AUC on CC	<b>0.58</b>		0.54		0.49	
AUC on PCE	0.50		<b>0.56</b>		0.50	
Time (s)	<b>170</b>		192		6332	

(d) Removing both Wiener and zero mean

Figure 5: TPR and FPR comparison using downloaded images from Facebook (only 9 devices)

	Original		VDNet		VDID	
	TPR	FPR	TPR	FPR	TPR	FPR
Normal comparison	0	0	0	0	0	0
Removing zero mean	0	0	0	0	0	0
Removing Wiener	0	0	0	0	0	0
Removing Both	0	0	0	0	0	0

Figure 6: A summary of comparisons using downloaded images from WhatsApp (only 9 devices)

determined their noise, which serves as a unique identification fingerprint, using three different noise extractor. Our work is an extension of [2], we implemented VDNet and VDID to extract fingerprints from captured images and compare the

results with the original method. We have tried 4 different strategies and found that removing the zero normalization or the Wiener filter can improve performances in some cases in terms of TPR, PCR and execution time. VDID, unexpectedly, performs poorly despite being a better denoiser than VDNet. The removal of both zero mean normalization and Wiener filter produces worse results in many cases and it doesn't take much better time benefits.

Finally, all methods fail the analysis of images processed by social networks. These methods unfortunately don't work for some newest devices and social network uploaded images. This is due to the fact that image elaboration algorithms may remove the fingerprint as we experimented with Huawei devices.

According to experiments, VDNet brings good results but it has script runtime and TPR score slightly worse compared to the original method. For future works, given the very fast progress of deep learning, it would be interesting to integrate VDNet with some changes in terms of loss terms or other hyper-parameters together with the original method as it provides quite similar results. It might be interesting to retrain VDNet with a more specific dataset for this task.

## REFERENCES

- [1] Z. Geradts et al. “Methods for identification of images acquired with digital cameras”. In: *Enabling Technologies for Law Enforcement and Security* (2001).
- [2] *Binghamton University*. [http://dde.binghamton.edu/download/camera\\_fingerprin](http://dde.binghamton.edu/download/camera_fingerprin).
- [3] P. Blythe and J. Fridrich. “Secure digital camera”. In: *Proc. Digital Forensic Research Workshop* (2004).
- [4] Nam Ik Cho Jae Woong Soh. “Variational Deep Image Denoising”. In: *ArXiv e-prints* (2021).
- [5] Jessica Fridrich Jan Lukas and Miroslav Goljan. “Digital Camera Identification From Sensor Pattern Noise”. In: *IEEE TRANSACTIONS ON INFORMATION FORENSICS AND SECURITY* (2006).
- [6] Y. Chen K. Zhang W. Zuo. “Beyond a Gaussian denoiser: Residual learning of deep CNN for image denoising”. In: *IEEE Transactions on Image Processing* (2017).
- [7] *LESC laboratory*. <https://lesc.dinfo.unifi.it>.
- [8] H. T. Sencar M. Kharrazi and N. Memon. “Blind source camera identification”. In: *Proc. ICIP* (2004).
- [9] Marco Fontani Massimo Iuliani and Alessandro Piva. “A Leak in PRNU Based Source Identification — Questioning Fingerprint Uniqueness”. In: *IEEE ACCESS* (2021).
- [10] Philipp Fischer Olaf Ronneberger and Thomas Brox. “U-net: Convolutional networks for biomedical image segmentation”. In: *International Conference on Medical image computing and computer-assisted intervention* (2015).
- [11] Qian Zhao Zongsheng Yue Hongwei Yong. “Variational Denoising Network: Toward Blind Noise Modeling and Removal”. In: *33rd Conference on Neural Information Processing Systems (NeurIPS 2019), Vancouver, Canada* (2019).

	Original		VDNet		VDID	
	TPR	FPR	TPR	FPR	TPR	FPR
Apple_iPhone4s	1	0	1	0	1	0
Apple_iPhone5c	0.9	0	0.85	0	0.6	0
Samsung_GalaxyTab3	0.85	0	0.85	0	0.75	0
Apple_iPhone4	1	0	1	0	1	0
Samsung_GalaxyS3	1	0	1	0	0.85	0
Sony_XperiaZ1Compact	1	0	1	0	1	0
Wiko_Ridge4G	1	0	1	0	1	0
OnePlus_A3000	0.7	0	0.7	0	0.6	0
Samsung_GalaxyS5	1	0	1	0	1	0
Huawei_P9	0.25	0	0.2	0	0.15	0
LG_D290	0.95	0	0.95	0	0.8	0
Lenovo_P70A	1	0	1	0	1	0
Apple_iPhone5c	0.75	0	0.75	0	0.5	0
Apple_iPhone6	1	0	1	0	0.8	0
Huawei_P9Lite	0.4	0	0.35	0	0.35	0
Microsoft_Lumia640LTE	0.95	0	0.95	0	0.85	0
Apple_iPhone5c	0.95	0	0.95	0	0.8	0
Apple_iPhone6Plus	1	0	1	0	0.75	0
Asus_Zenfone2Laser	1	0	1	0	1	0
Xiaomi_RedmiNote3	1	0	1	0	1	0
Huawei_P8	0.15	0	0.1	0	0	0
Apple_iPhone5	1	0	1	0	0.9	0
Huawei_Honor5c	0.35	0	0.35	0	0.35	0
AUC on CC	0.93		0.93		0.93	
AUC on PCE	0.96		0.94		0.91	
Time (s)	188		268		6679	

(a) Original implementation  
(mean TPR for original: 0.83, VDNet: 0.82, VDID: 0.74)

	Original		VDNet		VDID	
	TPR	FPR	TPR	FPR	TPR	FPR
Apple_iPhone4s	1	0	1	0	1	0
Apple_iPhone5c	0.9	0	0.85	0	0.6	0
Samsung_GalaxyTab3	0.85	0	0.85	0	0.75	0
Apple_iPhone4	1	0	1	0	1	0
Samsung_GalaxyS3	1	0	0.85	0	0.85	0
Sony_XperiaZ1Compact	1	0	1	0	1	0
Wiko_Ridge4G	1	0	1	0	1	0
OnePlus_A3000	0.8	0	0.7	0	0.6	0
Samsung_GalaxyS5	1	0	0.95	0	1	0
Huawei_P9	0.05	0	0	0	0	0
LG_D290	0.95	0	0.95	0	0.9	0
Lenovo_P70A	1	0	0.85	0	0.85	0
Apple_iPhone5c	0.75	0	0.7	0	0.5	0
Apple_iPhone6	1	0	0.95	0	0.85	0
Huawei_P9Lite	0.35	0	0.3	0	0.25	0
Microsoft_Lumia640LTE	0.95	0	0.95	0	0.85	0
Apple_iPhone5c	0.95	0	0.95	0	0.9	0
Apple_iPhone6Plus	0.95	0	0.95	0	0.8	0
Asus_Zenfone2Laser	1	0	1	0	1	0
Xiaomi_RedmiNote3	1	0	1	0	0.9	0
Huawei_P8	0.3	0	0.05	0	0.05	0
Apple_iPhone5	1	0	1	0	1	0
Huawei_Honor5c	0.35	0	0.35	0	0.35	0
AUC on CC	0.94		0.94		0.94	
AUC on PCE	0.97		0.94		0.93	
Time (s)	178		224		6345	

(b) Removing zero mean normalization  
(mean TPR for original: 0.83, VDNet: 0.83, VDID: 0.76)

	Original		VDNet		VDID	
	TPR	FPR	TPR	FPR	TPR	FPR
Apple_iPhone4s	1	0	0.95	0	1	0
Apple_iPhone5c	0.9	0	0.9	0	0.7	0
Samsung_GalaxyTab3	0.9	0	0.9	0	0.6	0
Apple_iPhone4	1	0	1	0	0.4	0
Samsung_GalaxyS3	1	0	0.85	0	0.85	0
Sony_XperiaZ1Compact	1	0	1	0	1	0
Wiko_Ridge4G	1	0	1	0	1	0
OnePlus_A3000	0.8	0	0.7	0	0.6	0
Samsung_GalaxyS5	1	0	0.95	0	1	0
Huawei_P9	0.05	0	0	0	0	0
LG_D290	0.95	0	0.95	0	0.9	0
Lenovo_P70A	1	0	0.85	0	0.85	0
Apple_iPhone5c	0.75	0	0.7	0	0.5	0
Apple_iPhone6	1	0	0.95	0	0.85	0
Huawei_P9Lite	0.35	0	0.3	0	0.25	0
Microsoft_Lumia640LTE	0.95	0	0.95	0	0.85	0
Apple_iPhone5c	0.95	0	0.95	0	0.9	0
Apple_iPhone6Plus	0.95	0	0.95	0	0.8	0
Asus_Zenfone2Laser	1	0	1	0	1	0
Xiaomi_RedmiNote3	1	0	1	0	0.9	0
Huawei_P8	0.3	0	0.05	0	0.05	0
Apple_iPhone5	1	0	1	0	1	0
Huawei_Honor5c	0.35	0	0.35	0	0.35	0
AUC on CC	0.97		0.95		0.95	
AUC on PCE	0.96		0.94		0.91	
Time (s)	169		210		6676	

(c) Removing Wiener filter  
(mean TPR for original: 0.83, VDNet: 0.73, VDID: 0.71)

	Original		VDNet		VDID	
	TPR	FPR	TPR	FPR	TPR	FPR
Apple_iPhone4s	1	0	0.85	0	0.4	0
Apple_iPhone5c	0.9	0	0.9	0	0	0
Samsung_GalaxyTab3	0.75	0	0.5	0	0	0
Apple_iPhone4	1	0	1	0	0	0
Samsung_GalaxyS3	1	0	0.6	0	0	0
Sony_XperiaZ1Compact	1	0	0.95	0	0	0
Wiko_Ridge4G	1	0	1	0	0.45	0
OnePlus_A3000	0.8	0	0.35	0	0	0
Samsung_GalaxyS5	1	0	0.95	0	0.55	0
Huawei_P9	0.05	0	0	0	0	0
LG_D290	0.95	0	0.95	0	0	0
Lenovo_P70A	1	0	1	0	0.6	0
Apple_iPhone5c	0.75	0	0.2	0	0.05	0
Apple_iPhone6	1	0	0.75	0	0	0
Huawei_P9Lite	0.4	0	0.35	0	0.25	0
Microsoft_Lumia640LTE	0.95	0	0.95	0	0.35	0
Apple_iPhone5c	0.95	0	0.95	0	0.15	0
Apple_iPhone6Plus	0.95	0	0.8	0	0	0
Asus_Zenfone2Laser	1	0	0.9	0	0	0
Xiaomi_RedmiNote3	1	0	1	0	0	0
Huawei_P8	0.15	0	0.05	0	0	0
Apple_iPhone5	1	0	0.95	0	0.2	0
Huawei_Honor5c	0.35	0	0.35	0	0.35	0
AUC on CC	0.95		0.75		0.52	
AUC on PCE	0.96		0.92		0.75	
Time (s)	182		207		6679	

(d) Removing both Wiener and zero mean  
(mean TPR for original: 0.82, VDNet: 0.70, VDID: 0.14)

Figure 7: Complete results: TPR and FPR comparison using VISION’s natural images

	Original		VDNet		VDID			Original		VDNet		VDID	
	TPR	FPR	TPR	FPR	TPR	FPR		TPR	FPR	TPR	FPR	TPR	FPR
Apple_iPhone4s	0	0	0	0	0	0		0	0	0	0	0	0
Apple_iPhone5c	0	0	0	0	0	0		0	0	0	0	0	0
Samsung_GalaxyTab3	<b>0.6</b>	0	<b>0.6</b>	0	0.15	0		0.6	0	<b>0.65</b>	0	0.25	0
Apple_iPhone4	0	0	0	0	0	0		0	0	0	0	0	0
Samsung_GalaxyS3	0	0	0	0	0	0		0	0	0	0	0	0
Sony_XperiaZ1Compact	0	0	0	0	0	0		0	0	0	0	0	0
Wiko_Ridge4G	0	0	0	0	0	0		0	0	0	0	0	0
OnePlus_A3000	0	0	0	0	0	0		0	0	0	0	0	0
Samsung_GalaxyS5	0	0	0	0	0	0		0	0	0	0	0	0
Huawei_P9	0	0	0	0	0	0		0	0	0	0	0	0
LG_D290	0	0	0	0	0	0		0	0	0	0	0	0
Lenovo_P70A	0	0	0	0	0	0		0	0	0	0	0	0
Apple_iPhone5c	0	0	0	0	0	0		0	0	0	0	0	0
Apple_iPhone6	0	0	0	0	0	0		0	0	0	0	0	0
Huawei_P9Lite	0	0	0	0	0	0		0	0	0	0	0	0
Microsoft_Lumia640LTE	0	0	0	0	0	0		0	0	0	0	0	0
Apple_iPhone5c	0	0	0	0	0	0		0	0	0	0	0	0
Apple_iPhone6Plus	0	0	0	0	0	0		0	0	0	0	0	0
Asus_Zenfone2Laser	0	0	0	0	0	0		0	0	0	0	0	0
Xiaomi_RedmiNote3	0	0	0	0	0	0		0	0	0	0	0	0
Huawei_P8	0	0	0	0	0	0		0	0	0	0	0	0
Apple_iPhone5	0	0	0	0	0	0		0	0	0	0	0	0
Huawei_Honor5c	0	0	0	0	0	0		0	0	0	0	0	0
AUC on CC	0.53		<b>0.54</b>		0.51			<b>0.54</b>		0.53		0.52	
AUC on PCE	0.49		0.53		<b>0.54</b>			0.51		<b>0.53</b>		<b>0.53</b>	
Time (s)	<b>189</b>		272		6654			<b>182</b>		220		6479	

(a) Original implementation

	Original		VDNet		VDID			Original		VDNet		VDID	
	TPR	FPR	TPR	FPR	TPR	FPR		TPR	FPR	TPR	FPR	TPR	FPR
Apple_iPhone4s	0	0	0	0	0	0		0	0	0	0	0	0
Apple_iPhone5c	0	0	0	0	0	0		0	0	0	0	0	0
Samsung_GalaxyTab3	<b>0.7</b>	0	0.5	0	0.15	0		<b>0.5</b>	0	0.2	0	0	0
Apple_iPhone4	0	0	0	0	0	0		0	0	0	0	0	0
Samsung_GalaxyS3	0	0	0	0	0	0		0	0	0	0	0	0
Sony_XperiaZ1Compact	0	0	0	0	0	0		0	0	0	0	0	0
Wiko_Ridge4G	0	0	0	0	0	0		0	0	0	0	0	0
OnePlus_A3000	0	0	0	0	0	0		0	0	<b>0.05</b>	0	0	0
Samsung_GalaxyS5	0	0	0	0	0	0		0	0	0	0	0	0
Huawei_P9	0	0	0	0	0	0		0	0	0	0	0	0
LG_D290	0	0	0	0	0	0		0	0	0	0	0	0
Lenovo_P70A	0	0	0	0	0	0		0	0	0	0	0	0
Apple_iPhone5c	0	0	0	0	0	0		0	0	0	0	0	0
Apple_iPhone6	0	0	0	0	0	0		0	0	0	0	0	0
Huawei_P9Lite	0	0	0	0	0	0		0	0	0	0	0	0
Microsoft_Lumia640LTE	0	0	0	0	0	0		0	0	0	0	0	0
Apple_iPhone5c	0	0	0	0	0	0		0	0	0	0	0	0
Apple_iPhone6Plus	0	0	0	0	0	0		0	0	0	0	0	0
Asus_Zenfone2Laser	0	0	0	0	0	0		0	0	0	0	0	0
Xiaomi_RedmiNote3	0	0	0	0	0	0		0	0	0	0	0	0
Huawei_P8	0	0	0	0	0	0		0	0	0	0	0	0
Apple_iPhone5	0	0	0	0	0	0		0	0	0	0	0	0
Huawei_Honor5c	0	0	0	0	0	0		0	0	0	0	0	0
AUC on CC	0.53		<b>0.54</b>		0.52			<b>0.53</b>		0.53		0.52	
AUC on PCE	0.51		0.53		<b>0.54</b>			0.49		<b>0.54</b>		0.53	
Time (s)	<b>174</b>		210		6524			<b>169</b>		199		6395	

(c) Removing Wiener filter

	Original		VDNet		VDID			Original		VDNet		VDID	
	TPR	FPR	TPR	FPR	TPR	FPR		TPR	FPR	TPR	FPR	TPR	FPR
Apple_iPhone4s	0	0	0	0	0	0		0	0	0	0	0	0
Apple_iPhone5c	0	0	0	0	0	0		0	0	0	0	0	0
Samsung_GalaxyTab3	<b>0.5</b>	0	0.2	0	0	0		<b>0.5</b>	0	0.2	0	0	0
Apple_iPhone4	0	0	0	0	0	0		0	0	0	0	0	0
Samsung_GalaxyS3	0	0	0	0	0	0		0	0	0	0	0	0
Sony_XperiaZ1Compact	0	0	0	0	0	0		0	0	0	0	0	0
Wiko_Ridge4G	0	0	0	0	0	0		0	0	0	0	0	0
OnePlus_A3000	0	0	<b>0.05</b>	0	0	0		0	0	<b>0.05</b>	0	0	0
Samsung_GalaxyS5	0	0	0	0	0	0		0	0	0	0	0	0
Huawei_P9	0	0	0	0	0	0		0	0	0	0	0	0
LG_D290	0	0	0	0	0	0		0	0	0	0	0	0
Lenovo_P70A	0	0	0	0	0	0		0	0	0	0	0	0
Apple_iPhone5c	0	0	0	0	0	0		0	0	0	0	0	0
Apple_iPhone6Plus	0	0	0	0	0	0		0	0	0	0	0	0
Asus_Zenfone2Laser	0	0	0	0	0	0		0	0	0	0	0	0
Xiaomi_RedmiNote3	0	0	0	0	0	0		0	0	0	0	0	0
Huawei_P8	0	0	0	0	0	0		0	0	0	0	0	0
Apple_iPhone5	0	0	0	0	0	0		0	0	0	0	0	0
Huawei_Honor5c	0	0	0	0	0	0		0	0	0	0	0	0
AUC on CC	<b>0.53</b>		<b>0.54</b>		0.52			<b>0.53</b>		0.53		0.52	
AUC on PCE	0.49		0.53		<b>0.54</b>			0.49		<b>0.54</b>		0.53	
Time (s)	<b>169</b>		199		6395			<b>169</b>		199		6395	

(b) Removing zero mean normalization

	Original		VDNet		VDID			Original		VDNet		VDID	
	TPR	FPR	TPR	FPR	TPR	FPR		TPR	FPR	TPR	FPR	TPR	FPR
Apple_iPhone4s	0	0	0	0	0	0		0	0	0	0	0	0
Apple_iPhone5c	0	0	0	0	0	0		0	0	0	0	0	0
Samsung_GalaxyTab3	<b>0.5</b>	0	0.2	0	0	0		<b>0.5</b>	0	0.2	0	0	0
Apple_iPhone4	0	0	0	0	0	0		0	0	0	0	0	0
Samsung_GalaxyS3	0	0	0	0	0	0		0	0	0	0	0	0
Sony_XperiaZ1Compact	0	0	0	0	0	0		0	0	0	0	0	0
Wiko_Ridge4G	0	0	0	0	0	0		0	0	0	0	0	0
OnePlus_A3000	0	0	<b>0.05</b>	0	0	0		0	0	<b>0.05</b>	0	0	0
Samsung_GalaxyS5	0	0	0	0	0	0		0	0	0	0	0	0
Huawei_P9	0	0	0	0	0	0		0	0	0	0	0	0
LG_D290	0	0	0	0	0	0		0	0	0	0	0	0
Lenovo_P70A	0	0	0	0	0	0		0	0	0	0	0	0
Apple_iPhone5c	0	0	0	0	0	0		0	0	0	0	0	0
Apple_iPhone6Plus	0	0	0	0	0	0		0	0	0	0	0	0
Asus_Zenfone2Laser	0	0	0	0	0	0		0	0	0	0	0	0
Xiaomi_RedmiNote3	0	0	0	0	0	0		0	0	0	0	0	0
Huawei_P8	0	0	0	0	0	0		0	0	0	0	0	0
Apple_iPhone5	0	0	0	0	0	0		0	0	0	0	0	0
Huawei_Honor5c	0	0	0	0	0	0		0	0	0	0	0	0
AUC on CC	<b>0.53</b>		<b>0.54</b>		0.52			<b>0.53</b>		0.53		0.52	
AUC on PCE	0.49		0.53		<b>0.54</b>			0.49		<b>0.54</b>		0.53	
Time (s)	<b>169</b>		199		6395			<b>169</b>		199		6395	

(d) Removing both Wiener and zero mean (mean TPR for original: 0.83, VDNet: 0.82, VDID: 0.74)

Figure 8: Complete results: TPR and FPR comparison using downloaded images from Facebook

	Original		VDNet		VDID	
	TPR	FPR	TPR	FPR	TPR	FPR
Apple_iPhone4s	0	0	0	0	0	0
Apple_iPhone5c	0	0	0	0	0	0
Samsung_GalaxyTab3	0	0	0	0	0	0
Apple_iPhone4	0	0	0	0	0	0
Samsung_GalaxyS3	0	0	0	0	0	0
Sony_XperiaZ1Compact	0	0	0	0	0	0
Wiko_Ridge4G	0	0	0	0	0	0
OnePlus_A3000	0	0	0	0	0	0
Samsung_GalaxyS5	0	0	0	0	0	0
Huawei_P9	0	0	0	0	0	0
LG_D290	0	0	0	0	0	0
Lenovo_P70A	0	0	0	0	0	0
Apple_iPhone5c	0	0	0	0	0	0
Apple_iPhone6	0	0	0	0	0	0
Huawei_P9Lite	0	0	0	0	0	0
Microsoft_Lumia640LTE	0	0	0	0	0	0
Apple_iPhone5	0	0	0	0	0	0
Apple_iPhone6Plus	0	0	0	0	0	0
Asus_Zenfone2Laser	0	0	0	0	0	0
Xiaomi_RedmiNote3	0	0	0	0	0	0
Huawei_P8	0	0	0	0	0	0
Apple_iPhone5	0	0	0	0	0	0
Huawei_Honor5c	0	0	0	0	0	0
AUC on CC	0.49		<b>0.5</b>		0.49	
AUC on PCE	0.49		0.49		<b>0.52</b>	
Time (s)	<b>192</b>		270		6645	

(a) Original implementation

	Original		VDNet		VDID	
	TPR	FPR	TPR	FPR	TPR	FPR
Apple_iPhone4s	0	0	0	0	0	0
Apple_iPhone5c	0	0	0	0	0	0
Samsung_GalaxyTab3	0	0	0	0	0	0
Apple_iPhone4	0	0	0	0	0	0
Samsung_GalaxyS3	0	0	0	0	0	0
Sony_XperiaZ1Compact	0	0	0	0	0	0
Wiko_Ridge4G	0	0	0	0	0	0
OnePlus_A3000	0	0	0	0	0	0
Samsung_GalaxyS5	0	0	0	0	0	0
Huawei_P9	0	0	0	0	0	0
LG_D290	0	0	0	0	0	0
Lenovo_P70A	0	0	0	0	0	0
Apple_iPhone5c	0	0	0	0	0	0
Apple_iPhone6	0	0	0	0	0	0
Huawei_P9Lite	0	0	0	0	0	0
Microsoft_Lumia640LTE	0	0	0	0	0	0
Apple_iPhone5	0	0	0	0	0	0
Apple_iPhone6Plus	0	0	0	0	0	0
Asus_Zenfone2Laser	0	0	0	0	0	0
Xiaomi_RedmiNote3	0	0	0	0	0	0
Huawei_P8	0	0	0	0	0	0
Apple_iPhone5	0	0	0	0	0	0
Huawei_Honor5c	0	0	0	0	0	0
AUC on CC	0.49		<b>0.5</b>		0.49	
AUC on PCE	0.5		0.49		<b>0.51</b>	
Time (s)	<b>180</b>		215		6390	

(b) Removing zero mean normalization

	Original		VDNet		VDID	
	TPR	FPR	TPR	FPR	TPR	FPR
Apple_iPhone4s	0	0	0	0	0	0
Apple_iPhone5c	0	0	0	0	0	0
Samsung_GalaxyTab3	0	0	0	0	0	0
Apple_iPhone4	0	0	0	0	0	0
Samsung_GalaxyS3	0	0	0	0	0	0
Sony_XperiaZ1Compact	0	0	0	0	0	0
Wiko_Ridge4G	0	0	0	0	0	0
OnePlus_A3000	0	0	0	0	0	0
Samsung_GalaxyS5	0	0	0	0	0	0
Huawei_P9	0	0	0	0	0	0
LG_D290	0	0	0	0	0	0
Lenovo_P70A	0	0	0	0	0	0
Apple_iPhone5c	0	0	0	0	0	0
Apple_iPhone6	0	0	0	0	0	0
Huawei_P9Lite	0	0	0	0	0	0
Microsoft_Lumia640LTE	0	0	0	0	0	0
Apple_iPhone5c	0	0	0	0	0	0
Apple_iPhone6Plus	0	0	0	0	0	0
Asus_Zenfone2Laser	0	0	0	0	0	0
Xiaomi_RedmiNote3	0	0	0	0	0	0
Huawei_P8	0	0	0	0	0	0
Apple_iPhone5	0	0	0	0	0	0
Huawei_Honor5c	0	0	0	0	0	0
AUC on CC	0.47		<b>0.50</b>		0.49	
AUC on PCE	<b>0.52</b>		<b>0.52</b>		0.50	
Time (s)	<b>179</b>		209		6638	

(c) Removing Wiener filter

	Original		VDNet		VDID	
	TPR	FPR	TPR	FPR	TPR	FPR
Apple_iPhone4s	0	0	0	0	0	0
Apple_iPhone5c	0	0	0	0	0	0
Samsung_GalaxyTab3	0	0	0	0	0	0
Apple_iPhone4	0	0	0	0	0	0
Samsung_GalaxyS3	0	0	0	0	0	0
Sony_XperiaZ1Compact	0	0	0	0	0	0
Wiko_Ridge4G	0	0	0	0	0	0
OnePlus_A3000	0	0	0	0	0	0
Samsung_GalaxyS5	0	0	0	0	0	0
Huawei_P9	0	0	0	0	0	0
LG_D290	0	0	0	0	0	0
Lenovo_P70A	0	0	0	0	0	0
Apple_iPhone5c	0	0	0	0	0	0
Apple_iPhone6	0	0	0	0	0	0
Huawei_P9Lite	0	0	0	0	0	0
Microsoft_Lumia640LTE	0	0	0	0	0	0
Apple_iPhone5	0	0	0	0	0	0
Apple_iPhone6Plus	0	0	0	0	0	0
Asus_Zenfone2Laser	0	0	0	0	0	0
Xiaomi_RedmiNote3	0	0	0	0	0	0
Huawei_P8	0	0	0	0	0	0
Apple_iPhone5	0	0	0	0	0	0
Huawei_Honor5c	0	0	0	0	0	0
AUC on CC	0.48		<b>0.49</b>		<b>0.49</b>	
AUC on PCE	0.51		0.49		<b>0.52</b>	
Time (s)	<b>190</b>		204		6530	

(d) Removing both Wiener and zero mean

Figure 9: Complete results: TPR and FPR comparison using downloaded images from Whatsapp

	Original		VDNet		VDID	
	TPR	FPR	TPR	FPR	TPR	FPR
ane-lx1 10808472_N05	<b>0.25</b>	0	<b>0.25</b>	0	0.05	0
ane-lx1 12664195_N06	0	0	0	0	0	0
ctt-al00 151215112_N02	<b>0.3</b>	<b>0.01</b>	<b>0.3</b>	<b>0.01</b>	<b>0.3</b>	<b>0.01</b>
ctt-al00 97411860_N07	<b>0.05</b>	<b>0.01</b>	<b>0.05</b>	<b>0.01</b>	<b>0.1</b>	<b>0.01</b>
ele-i29 103949957_N08	0	<b>0.01</b>	0	<b>0.01</b>	0	<b>0.02</b>
ele-i29 109358335_N06	0	0	0	0	0	0
lya-i29 10457804_N06	<b>0.05</b>	<b>0.01</b>	<b>0.05</b>	<b>0.01</b>	<b>0.05</b>	<b>0.01</b>
lya-i29 11345399_N07	0.1	0	0.1	0	0	0
mar-lx1a 101212512_N07	<b>0.35</b>	0	<b>0.35</b>	0	0.3	0
mar-lx1a 82398988_N03	<b>0.8</b>	0	<b>0.8</b>	0	0.45	0
pot-lx1 11827711_N06	<b>0.25</b>	0	<b>0.25</b>	0	0.05	0
pot-lx1 121953481_N02	<b>0.15</b>	0	<b>0.15</b>	0	<b>0.15</b>	0
sne-lx1 107743946_N08	<b>0.05</b>	0	<b>0.05</b>	0	<b>0.05</b>	0
sne-lx1 119610003_N04	<b>0.45</b>	0	<b>0.45</b>	0	0.15	0
vog-i29 12965595_N05	0	0	0	0	<b>0.05</b>	0
vog-i29 13639096_N06	0	0	0	0	0	0
vtr-l09 10471762_N02	<b>0.05</b>	0	<b>0.05</b>	0	0	0
vtr-l09 113815707_N06	0.4	0	0.4	0	0.25	0
AUC on CC	<b>0.82</b>		<b>0.82</b>		<b>0.82</b>	
AUC on PCE	<b>1</b>		0.76		<b>1</b>	
Time (s)	<b>185</b>		275		6339	

(a) Original implementation  
(mean TPR for original: 0.18, VDNet: 0.18, VDID: 0.10)

	Original		VDNet		VDID	
	TPR	FPR	TPR	FPR	TPR	FPR
ane-lx1 10808472_N05	<b>0.3</b>	0	0	0	0	0
ane-lx1 12664195_N06	0	0	0	0	0	0
ctt-al00 151215112_N02	<b>0.25</b>	<b>0.01</b>	0.2	0	0.2	0
ctt-al00 97411860_N07	<b>0.05</b>	0	0	0	0	0
ele-i29 103949957_N08	0	<b>0.01</b>	0	<b>0.01</b>	0	<b>0.01</b>
ele-i29 109358335_N06	0	0	0	0	0	0
lya-i29 10457804_N06	<b>0.05</b>	<b>0.01</b>	<b>0.05</b>	<b>0.01</b>	<b>0.05</b>	<b>0.01</b>
lya-i29 11345399_N07	0	0	0	0	0	0
mar-lx1a 101212512_N07	<b>0.15</b>	0	0	0	0	0
mar-lx1a 82398988_N03	<b>0.35</b>	0	0.05	0	0	0
pot-lx1 11827711_N06	<b>0.25</b>	0	0.05	0	0	0
pot-lx1 121953481_N02	<b>0.1</b>	0	0.05	0	0.05	0
sne-lx1 107743946_N08	<b>0.1</b>	0	0	0	0	0
sne-lx1 119610003_N04	<b>0.45</b>	0	0.05	0	0	0
vog-i29 12965595_N05	0	0	0	0	0	0
vog-i29 13639096_N06	0	0	0	0	0	0
vtr-l09 10471762_N02	<b>0.05</b>	0	0	0	0	0
vtr-l09 113815707_N06	<b>0.15</b>	0	0.05	0	0	0
AUC on CC	<b>0.88</b>		0.81		0.8	
AUC on PCE	<b>1</b>		1		<b>1</b>	
Time (s)	<b>170</b>		221		6579	

(c) Removing Wiener filter  
(mean TPR for original: 0.12, VDNet: 0.02, VDID: 0.01)

	Original		VDNet		VDID	
	TPR	FPR	TPR	FPR	TPR	FPR
ane-lx1 10808472_N05	<b>0.35</b>	0	0.15	0	0.1	0
ane-lx1 12664195_N06	0	0	<b>0.05</b>	0	0	0
ctt-al00 151215112_N02	<b>0.3</b>	<b>0.01</b>	0.3	<b>0.01</b>	<b>0.45</b>	<b>0.01</b>
ctt-al00 97411860_N07	<b>0.1</b>	<b>0.01</b>	0	<b>0.01</b>	<b>0.1</b>	<b>0.01</b>
ele-i29 103949957_N08	0	<b>0.01</b>	0	<b>0.01</b>	0	<b>0.06</b>
ele-i29 109358335_N06	0	0	0	0	0	0
lya-i29 10457804_N06	<b>0.05</b>	<b>0.01</b>	<b>0.05</b>	<b>0.01</b>	<b>0.05</b>	<b>0.02</b>
ly-a-i29 11345399_N07	0.1	0	0	0	0	0
mar-lx1a 101212512_N07	<b>0.4</b>	0	0.25	0	0.3	0
mar-lx1a 82398988_N03	<b>0.85</b>	0	0.5	0	0.65	0
pot-lx1 11827711_N06	<b>0.3</b>	0	0.05	0	0.1	0
pot-lx1 121953481_N02	<b>0.2</b>	0	0.05	0	0.15	0
sne-lx1 107743946_N08	<b>0.1</b>	0	0.05	0	0.05	0
sne-lx1 119610003_N04	<b>0.45</b>	0	0.15	0	0.25	0
vog-i29 12965595_N05	0	0	0	0	<b>0.05</b>	0
vog-i29 13639096_N06	0	0	0	0	0	0
vtr-l09 10471762_N02	<b>0.1</b>	0	0	0	0	0
vtr-l09 113815707_N06	<b>0.45</b>	0	0.4	0	<b>0.45</b>	0
AUC on CC	0.83		0.82		<b>0.85</b>	
AUC on PCE	<b>1</b>		1		<b>1</b>	
Time (s)	<b>187</b>		270		6620	

(b) Removing zero mean normalization  
(mean TPR for original: 0.20, VDNet: 0.11, VDID: 0.17)

	Original		VDNet		VDID	
	TPR	FPR	TPR	FPR	TPR	FPR
ane-lx1 10808472_N05	<b>0.3</b>	0	0	0	0	0
ane-lx1 12664195_N06	0	0	0	0	0	0
ctt-al00 151215112_N02	<b>0.25</b>	<b>0.01</b>	0.2	0	0.15	0
ctt-al00 97411860_N07	<b>0.05</b>	0	0	0	0	0
ele-i29 103949957_N08	0	<b>0.01</b>	0	<b>0.01</b>	0	0
ele-i29 109358335_N06	0	0	0	0	0	0
lya-i29 10457804_N06	<b>0.05</b>	<b>0.01</b>	<b>0.05</b>	<b>0.01</b>	<b>0.05</b>	<b>0.01</b>
ly-a-i29 11345399_N07	0	0	0	0	0	0
mar-lx1a 101212512_N07	<b>0.15</b>	0	0	0	0	0
mar-lx1a 82398988_N03	<b>0.35</b>	0	0.05	0	0	0
pot-lx1 11827711_N06	<b>0.25</b>	0	0.05	0	0	0
pot-lx1 121953481_N02	<b>0.1</b>	0	0.05	0	0.05	0
sne-lx1 107743946_N08	<b>0.1</b>	0	0	0	0	0
sne-lx1 119610003_N04	<b>0.45</b>	0	0.05	0	0	0
vog-i29 12965595_N05	0	0	0	0	0	0
vog-i29 13639096_N06	0	0	0	0	0	0
vtr-l09 10471762_N02	<b>0.05</b>	0	0	0	0	0
vtr-l09 113815707_N06	<b>0.15</b>	0	0.05	0	0	0
AUC on CC	<b>0.88</b>		0.81		0.8	
AUC on PCE	<b>1</b>		1		<b>1</b>	
Time (s)	<b>170</b>		221		6579	

(d) Removing both Wiener and zero mean  
(mean TPR for original: 0.18, VDNet: 0.03, VDID: 0.008)

Figure 10: Complete results: TPR and FPR comparison using Huawei devices