PROTECTION AGAINST REVERSE ENGINEERING IN DIGITAL CAMERAS

Matthew C. Stamm and K. J. Ray Liu

Dept. of Electrical and Computer Engineering, University of Maryland, College Park

ABSTRACT

Over the past decade, a number of digital forensic techniques have been developed to authenticate digital signals. One important set of forensic techniques operates by estimating signal processing components of a digital camera's signal processing pipeline, then using these estimates to perform forensic tasks such as camera identification or forgery detection. However, because these techniques are capable of estimating a camera's internal signal processing components, these forensic techniques can be used for reverse engineering. In this paper, we propose integrating an anti-forensic module into a digital camera's processing pipeline to protect against forensic reverse engineering. Our proposed technique operates by removing linear dependencies amongst an output images interpolated color values and by disrupting the color sampling grid. Experimental results show that our proposed technique can be effectively used to protect against the forensic reverse engineering of key components of a digital camera's processing pipeline.

Index Terms— Anti-Forensics, Digital Forensics, Reverse Engineering, Color Interpolation

1. INTRODUCTION

In today's society, the majority of multimedia content that we encounter is digital. Because digital content can be easily altered, researchers have developed a number of forensic techniques to authenticate digital multimedia signals. Existing forensic techniques are capable of detecting multimedia forgeries and determining which editing operations were used to manipulate a signal [1, 2, 3, 4]. Others can identify the device used to capture a multimedia signal [1] and provide information about how that device processes the signal internally [5, 6, 7]. Though digital forensics is still a relatively young field, researchers are able to determine a surprising amount of information about a multimedia signal's processing history using only the signal itself.

Recently, researchers have begun examining anti-forensic operations designed to fool forensic techniques. By studying anti-forensics, researchers can identify weaknesses in existing forensic techniques that a forger may attempt to exploit. Additionally, new forensic techniques can be developed to identifying the use of anti-forensics. Anti-forensic techniques have been developed to hide fingerprints left by image resizing or rotation [8], disguise an image's compression history [9], cover up evidence of frame deletion in digital videos [10], and falsify the photo-response non-uniformity (PRNU) fingerprint left in digital images by sensor imperfections [11].

Though the intended use of multimedia forensics is to provide information security, researchers have overlooked an important unintended use of forensic techniques: *multimedia forensics can be used to reverse engineer proprietary signal processing components in digital devices*! Digital cameras are an important example of this. Forensic techniques exist to estimate the color filter array (CFA) pattern and interpolation coefficients used during the image formation process [12, 5, 13, 14]. Furthermore, a camera's white balancing parameters can be forensically estimated [6]. Since camera manufacturers likely wish to protect their proprietary implementations of both color interpolation and white balancing, digital forensic techniques may in fact pose an intellectual property threat.

Because forensic techniques pose an information security threat when viewed in this light, *we propose using anti-forensics to protect against reverse engineering*. To accomplish this, we propose placing an anti-forensic processing module at the end of a device's internal signal processing pipeline. This will prevent forensic techniques from using a device's output to estimate signal processing operations inside the device.

In this paper, we propose a proof-of-concept technique to prevent a digital camera's color interpolation method from being forensically reverse engineered. We accomplish this through a combination of nonlinear filtering and perturbations to an image's sampling grid. We demonstrate the effectiveness of our proposed technique by testing the ability of existing forensic algorithms to identify the color interpolation method used to form an image after our anti-forensic technique has been applied.

2. THE IMAGE PROCESSING PIPELINE

A digital camera operates by measuring the intensity of light reflected from a real world scene R onto an electronic sensor known as a charged coupling device (CCD), as is shown in Fig. 1. The light enters the camera by first passing through a lens. Since most CCDs are only capable of measuring one color of light at each pixel location, the light next passes through a color filter array ρ. The CFA is an optical filter consisting of a repeating fixed pattern (typically 2x2) which allows only one color band of light (red, green, or blue) to fall incident on the CCD at a particular pixel location.

The CCD then measures the light intensity of the corresponding color band at each pixel location. This yields an image S constructed of three partially sampled color layers such that

$$S_{x,y,c} = \begin{cases} R_{x,y,c} & \text{if } \rho_{x,y} = c, \\ 0 & \text{otherwise.} \end{cases} \tag{1}$$

where x and y are indices denoting a pixel's spatial location and c specifies its color layer.

Next, unobserved color layer values at each pixel location are interpolated using nearby directly observed color layer values. This interpolation process can be performed in many ways and is typically camera model specific. After this, the image may be subject to internal post-processing, such as white balancing, before the final image I is stored or output.

Email: {mcstamm,kjrliu}@umd.edu

This work is supported in part by AFOSR grant FA95500910179.

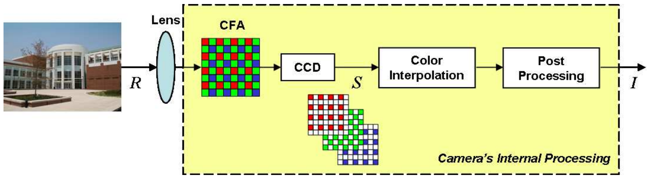

Fig. 1. A digital camera's signal processing pipeline.

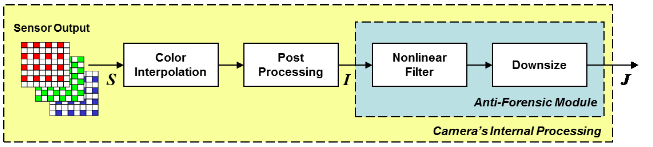

Fig. 2. A digital camera's internal processing pipeline with our proposed anti-forensic module integrated into it.

2.1. Component Forensics

Knowledge of a camera's color interpolation coefficients and CFA pattern can be used to perform a variety of forensic tasks. Forensic techniques that estimate a camera's color interpolation coefficients and CFA pattern, then use these to perform another forensic task are known as *component forensic* techniques. Component forensic techniques have been developed to identify forgeries by detecting localized interpolation irregularities [12, 15]. Because interpolation methods and their parameters are typically camera model specific, other component forensic techniques have been developed to identify an image's source camera [5, 13, 14]. Others use knowledge of the CFA pattern and interpolation coefficients to estimate parameters of a camera's internal post-processing operations such as white balancing [6].

While component forensic techniques vary in the specific way that they estimate a camera's color interpolation coefficients, they all share the same basic structure. Here we use the technique proposed by Swaminathan et al. [5] as a baseline and describe how it operates.

First, an image's CFA pattern is assumed. By doing this, a forensic examiner can separate directly observed pixels in a color layer from those that have been interpolated. Next, the directly observed color values are used to calculate the horizontal and vertical gradients of each pixel. These are used to classify each each pixel into one of three sets for each color layer depending on the strength of its horizontal and vertical gradient. For each of the nine pairings of color layer and gradient class, a the directly observed and interpolated color layer values are used to obtain a least squares estimate of the color interpolation filter coefficients.

Since in most cases the true CFA pattern is not known, this process is repeated for each of the 36 possible 2×2 CFA patterns. After the set of interpolation coefficients is estimated for a candidate CFA pattern, each color layer is resampled using the candidate CFA and the color layers are interpolated using the estimated coefficients. The difference between the original image and the re-interpolated image is then calculated for each set of estimated interpolation coefficients and CFA pattern. The CFA pattern and interpolation coefficients that result in the lowest difference are chosen as the final estimate.

The estimated color interpolation coefficients can be used to train a support vector machine (SVM) to identify the color interpolation method used or identify the model of the camera used to capture an image.

3. ANTI-FORENSIC REVERSE ENGINEERING PREVENTION

Any party attempting to reverse engineer a digital camera will need to determine what signal processing is performed inside of it. Because one of the key elements of a digital camera's internal signal processing pipeline is its color interpolation method M, the color interpolation method must be identified in order to reverse engineer the camera. This can be done using component forensic techniques, since they are capable of both identifying the color interpolation method used by a camera and estimating the color interpolation coefficients. Though reverse engineering is not the intended use of component forensic techniques, nothing prevents a malicious party from using them this way. As a result, camera manufacturers may wish to incorporate some form of protection against forensic reverse engineering into their devices.

Since component forensic techniques can be used to estimate a digital camera's color interpolation method, we propose using anti-forensics to protect against reverse engineering. To assess the effectiveness of a reverse engineering protection technique, let $\hat{M}$ be the interpolation method identified by a component forensic technique and let $\mathcal{M}$ be the set of all candidate interpolation methods. For a given interpolation method $m \in \mathcal{M}$, the probability $P_C^{(m)}$ that the interpolation method was correctly identified by the component forensic technique is

$$P_C^{(m)} = P(M = m | \hat{M} = m). \tag{2}$$

A technique that prevents component forensics from being able to reverse engineer the interpolation method should seek to reduce $P_C^{(m)}$. This probability does not need to be reduced to zero, however. If $P_C^{(m)}$ can be reduced to the point that $\hat{M}$ is no better than a random guess, then perfect protection can be achieved. This is because the output of the component forensic algorithm will be completely unreliable. This is achieved if $P_C^{(m)} \leq 1/|\mathcal{M}|$ for all $m \in \mathcal{M}$, where $|\mathcal{M}|$ denotes the cardinality of the set $\mathcal{M}$.

When designing our reverse engineering protection technique, we note that Swaminathan et al.'s algorithm obtains an estimate of the camera's color interpolation coefficients as follows. For a given color layer and CFA pattern, each interpolated pixel B is written as a linear combination of nearby directly observed pixel values S according to the equation

$$B_{x,y} = \sum_{(i,j) \in \Omega_I} w_{i,j}^{(I)} S_{x+i,y+j} \tag{3}$$

where $w^{(I)}$ is the interpolation filter and Ω_I is its support. These equations are grouped by color layer and gradient class into systems of equations of the form $\mathbf{Sw} = \mathbf{b}$. A least squares approximation of the interpolation filter coefficients is calculated for each set of equations, resulting in nine sets of filter coefficients.

To combat forensic reverse engineering, we propose incorporating an anti-forensic module into a digital camera's processing pipeline as is shown in Fig. 2. This module is designed to interfere with two important aspects of component forensic algorithms:

1. The estimate of the interpolation method is linear.
2. This linear estimate depends on the ability of the forensic algorithm to guess which color layer values were directly observed and which were interpolated.

The first element of our anti-forensic module is a nonlinear filter. This is used to reduce linear dependencies between interpolated pixel values and nearby directly observed pixel values. In this

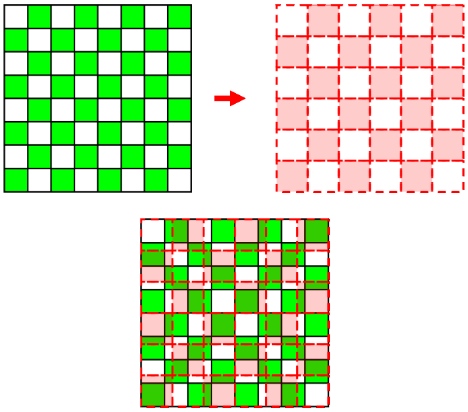

Fig. 3. Top: Changes in the effective area of each pixel after downsizing. Bottom: A downsized color layer overlaid on the pixels of the original color layer.

proof-of-concept implementation, we use a median filter with support $s = 2$ to obtain our nonlinearly filtered image F so that

$$F_{x,y} = med\{I_{i,j} | |i-x| \leq \lfloor s/2 \rfloor, |j-y| \leq \lfloor s/2 \rfloor\} \quad (4)$$

The second element of our anti-forensic module involves downsizing the image by a small factor. This is done to disrupt the color sampling grid and prevent the forensic algorithm from identifying directly observed and interpolated color values. Each pixel in the downsized image will correspond to a greater effective area than in the originally sized image. As a result, no pixel in the downsized image will correspond solely to a directly observed or color interpolated pixel. This phenomenon is shown in Fig. 3.

In this proof-of-concept implementation, we downscale using bilinear interpolation. Let each color layer be $X \times Y$ pixels before downsizing and $P \times Q$ pixels after. Also, let the integer pixel location (p, q) in the downsized image corresponds the real valued location (u, v) in the originally sized image. These locations are related according to the equations $u = (p(X-1) + P - X)/(P-1)$ and $v = (q(Y-1)+Q-Y)/(Q-1)$. Additionally, let $x \leq u < x+1$ and $y \leq v < y+1$ as is shown in Fig. 4. Each pixel $G_{u,v}$ in the downscaled color layer is given by

$$G_{u,v} = \sum_{(k,l) \in \Omega_D} w_{k,l}^{(D)}(u, v) F_{x+k,y+l}, \quad (5)$$

where $w_{k,l}^{(D)}(u, v)$ are the spatially varying downscaling coefficients and $\Omega_D = \{(0,0),(0,1),(1,0),(1,1)\}$. The coefficients of the bilinear downscaling filter are calculated using the following equations: $w_{0,0}^{(D)}(u, v) = (1-u+x)(1-v+y)$, $w_{0,1}^{(D)}(u, v) = (1-u+x)(v-y)$, $w_{1,0}^{(D)}(u, v) = (u-x)(1-v+y)$, and $w_{1,1}^{(D)}(u, v) = (u-x)(v-y)$.

Combining (4), (5), and the expressions relating p and q to u and v, the output of the anti-forensic module can be written as

$$G_{p,q} = \sum_{(k,l) \in \Omega_D} \Big(w_{k,l}^{(D)}\big(\tfrac{p(X-1)+P-X}{P-1}, \tfrac{q(Y-1)+Q-Y}{Q-1}\big) \times med\{I_{i,j} | |i-x-k| \leq \lfloor s/2 \rfloor, |j-y-l| \leq \lfloor s/2 \rfloor\}\Big). \quad (6)$$

When our proposed anti-forensic module is employed, both directly observed and interpolated color layer values are modified according to this expression. As a result, both $S_{x,y}$ and $B_{x,y}$ are modified in (3) causing the least squares estimate to result in a poor approximation of the interpolation method.

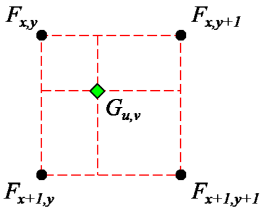

Fig. 4. Bilinear interpolation example.

While these anti-forensic measures help prevent component forensic techniques from being able to reverse engineer a camera's color interpolation method, they affect the quality of the output image. In practice, we have found that the image needs to be downscaled only a minimal amount in order to protect against forensic reverse engineering. This corresponds to an unavoidable but extremely minor cost to the output image's quality.

The anti-forensic component that has the greatest effect on the output image's quality is the nonlinear filter. Since the median filter has the desirable property of preserving edge content in an image, it is well suited for our purposes. Still, we may wish to balance the trade-off between the level of reverse engineering protection and the quality of the output image. To do this, we create a second image H by downscaling I to the same size as G without applying median filtering. We then form the final output image J by randomizing between G and H according to the equation

$$J_{x,y} = \theta_{x,y} G_{x,y} + (1 - \theta_{x,y}) H_{x,y} \quad (7)$$

where $\theta x, y$ is a random variable taking the value 0 or 1 and $P(\theta_{x,y} = 1) = \phi$. The choice of ϕ controls the balance between anti-forensic protection and image quality.

4. SIMULATIONS AND RESULTS

In order to evaluate the performance of our anti-forensic reverse engineering prevention technique, we created a test database of images whose color interpolation method and CFA pattern was known as ground truth.This was done by first creating a set of 100 640×480 pixel images from version 1 of the Uncompressed Colour Image Database [16]. Next, we resampled the color components of each image using the Bayer pattern as the CFA pattern. We then performed color interpolation using five different color interpolation methods: bilinear, bicubic, nearest neighbor, median filter, and smooth hue transition. Descriptions of the interpolation methods used can be found in [5] and [12] (we note that the median filter color interpolation technique is not simply applying a median filter to each color layer).

The resulting 500 images model the direct output of a digital camera. Because post-processing such as compression decreases the performance of component forensic techniques [5], we did not subject these images to post-processing. This allowed us to evaluate the performance of our anti-forensic module operating under worst case conditions; i.e. ideal conditions for component forensic techniques. Additionally, this mimics the settings that would likely be chosen by someone wishing to reverse engineer a camera using component forensics if they had access to the camera. Finally, we passed these images through our proposed anti-forensic module while varying ϕ

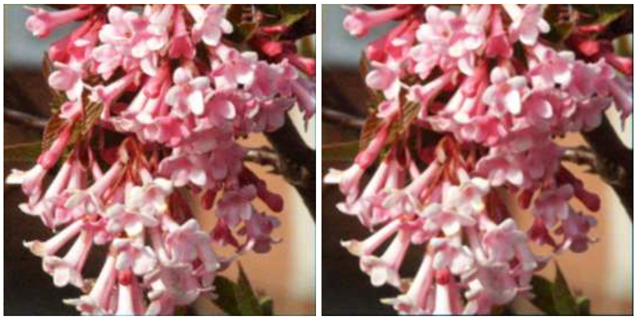

Fig. 5. Left: A typical image formed using bilinear color interpolation. Right: The same image after being passed through our anti-forensic module.

between 0.1 and 1 in increments of 0.1. In our anti-forensic module, we downscaled each image by 4 pixels in both the horizontal and vertical directions. This created 5000 anti-forensically modified images in addition to the 500 unmodified images. Fig. 5 shows an example of an image before and after it passes through our anti-forensic module.

After constructing our image database, we used the component forensic technique proposed by Swaminathan et al. in [5] to estimate the CFA pattern and color interpolation coefficients for each of the 5500 images. We then trained a support vector machine (SVM) with a radial basis function kernel to identify the color interpolation method used to form each image [17].

To achieve a baseline assessment of the component forensic technique's ability to identify each color interpolation method, we first evaluated it using only unmodified images. This was done using cross validation by dividing the set of unmodified images into 10 subsets. The color interpolation method was identified for every image in a given subset after training the SVM using the remaining 9 subsets. This process was repeated for each of the 10 subsets. The results were used to calculate $P_C^{(m)}$ for each interpolation method according to the equation

$$P_C^{(m)} = \sum_n \frac{\mathbb{1}(M_n = m, \hat{M}_n = m)}{\mathbb{1}(\hat{M}_n = m)}. \qquad (8)$$

where n is the picture index and $\mathbb{1}(\cdot)$ is the indicator function. When testing on unmodified images, the component forensic technique achieved perfect performance, i.e. $P_C^{(m)} = 1$ for each of the 5 color interpolation techniques.

Next, we tested the effectiveness of our anti-forensic module. We did this by using the trained SVM to identify the color interpolation method used to form each of the 5000 anti-forensically modified images. To ensure that image content had no influence on the identification results, the anti-forensically modified images were divided into 10 subsets corresponding to the unmodified training images. During testing, the SVM was trained using the 9 subsets of training data corresponding to the unused testing subsets. This data was used to calculate $P_C^{(m)}$ for every pairing of interpolation method and downscaling amount. Additionally, we measured the quality of the anti-forensically modified images by calculating the structural similarity (SSIM) between each image and its corresponding original version before it passed through our anti-forensic module [18]. Since our anti-forensic module changes the dimension of the image, we rescaled our anti-forensically modified images back to their original size in order to calculate the SSIM. Rescaling in this manner will cause a further decrease in the quality of the anti-forensically modified images, therefore our results can be seen as a lower bound on the image quality.

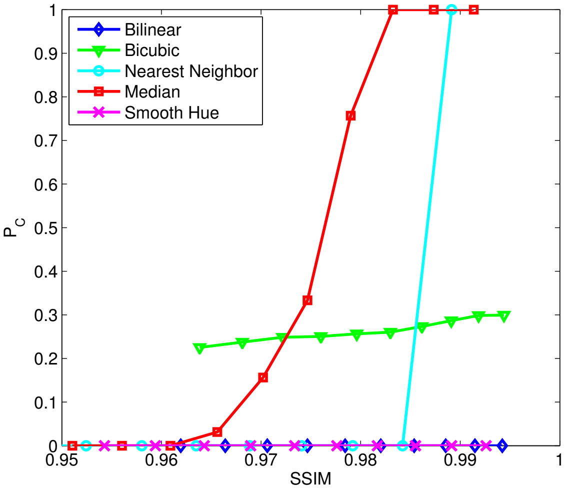

Fig. 6. Experimental results showing output image quality (SSIM) versus the probability of correctly estimating the color interpolation technique (P_C).

The results of this test are shown in Fig. 6. Since 5 candidate interpolation methods were considered, the component forensic technique performs better than a random guess only when $P_C^{(m)} \leq 0.2$. Our results show that $P_C^{(m)} > 0.2$ only when bicubic color interpolation is considered. Even in this case, $P_C^{(bicubic)} = 0.225$ which exhibits little improvement over a random guess. Furthermore, we note our anti-forensic module can provide reverse engineering protection while keeping the SSIM above 0.96. These results suggest that our proposed anti-forensic module is very effective at protecting against reverse engineering.

5. CONCLUSIONS

In this paper, we have proposed a new anti-forensic module to be incorporated into a digital camera's signal processing pipeline to protect against reverse engineering. By introducing nonlinearities into an image and disrupting its color sampling grid, our anti-forensic module prevents component forensic techniques from accurately estimating the color interpolation method used by a digital camera during the image formation process. Through a set of experiments, we have demonstrated that our proposed anti-forensic technique is able to reduce the performance of a component forensic technique to that of a random guess or worse in nearly all cases maintaining a high image quality.

6. REFERENCES

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

[2] H. Farid, "Image forgery detection," *IEEE Signal Processing Magazine*, vol. 26, no. 2, pp. 16 –25, Mar. 2009.

[3] M. C. Stamm and K. J. R. Liu, "Forensic detection of image manipulation using statistical intrinsic fingerprints," *IEEE Trans. Information Forensics and Security*, vol. 5, no. 3, pp. 492 –506, Sep. 2010.

[4] M. Barni, A. Costanzo, and L. Sabatini, “Identification of cut & paste tampering by means of double-jpeg detection and image segmentation,” in *Proc. IEEE ISCAS*, June 2010, pp. 1687 –1690.

[5] A. Swaminathan, M. Wu, and K.J.R. Liu, “Nonintrusive component forensics of visual sensors using output images,” *IEEE Trans. Information Forensics and Security*, vol. 2, no. 1, pp. 91 –106, Mar. 2007.

[6] A. Swaminathan, M. Wu, and K.J.R. Liu, “Optimization of input pattern for semi non-intrusive component forensics of digital cameras,” in *Proc. IEEE ICASSP*, Honolulu, HI, Apr. 2007, pp. II–225 –II–228.

[7] X. Chu, M.C. Stamm, W.S. Lin, and K.J.R. Liu, “Forensic identification of compressively sensed images,” in *Proc. IEEE ICASSP*, Mar. 2012, pp. 1837 –1840.

[8] M. Kirchner and R. Bohme, “Hiding traces of resampling in digital images,” *IEEE Trans. Information Forensics and Security*, vol. 3, no. 4, pp. 582–592, Dec. 2008.

[9] M. C. Stamm and K. J. R. Liu, “Anti-forensics of digital image compression,” *IEEE Trans. Information Forensics and Security*, vol. 6, no. 3, pp. 1050 –1065, Sep. 2011.

[10] M.C. Stamm, W.S. Lin, and K.J.R. Liu, “Temporal forensics and anti-forensics for motion compensated video,” *IEEE Trans. Information Forensics and Security*, vol. 7, no. 4, pp. 1315 –1329, Aug. 2012.

[11] T. Gloe, M. Kirchner, A. Winkler, and R. Böhme, “Can we trust digital image forensics?,” in *15th Int. Conf. Multimedia*, Augsburg, Germany, 2007, pp. 78–86.

[12] A.C. Popescu and H. Farid, “Exposing digital forgeries in color filter array interpolated images,” *IEEE Trans. Signal Processing*, vol. 53, no. 10, pp. 3948 – 3959, Oct. 2005.

[13] H. Cao and A.C. Kot, “Accurate detection of demosaicing regularity for digital image forensics,” *IEEE Trans. Information Forensics and Security*, vol. 4, no. 4, pp. 899 –910, Dec. 2009.

[14] W. H. Chuang and M. Wu, “Semi non-intrusive training for cell-phone camera model linkage,” in *Proc. IEEE WIFS*, Seattle, WA, Dec. 2010, pp. 1 –6.

[15] A. Swaminathan, M. Wu, and K.J.R. Liu, “Digital image forensics via intrinsic fingerprints,” *IEEE Trans. Information Forensics and Security*, vol. 3, no. 1, pp. 101–117, Mar. 2008.

[16] G. Schaefer and M. Stich, “UCID: an uncompressed color image database,” in *Proc. SPIE: Storage and Retrieval Methods and Applications for Multimedia*, 2004, vol. 5307, pp. 472–480.

[17] C. C. Chang and C. J. Lin, “LIBSVM: A library for support vector machines,” *ACM Trans. Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011.

[18] Z. Wang, A.C. Bovik, H.R. Sheikh, and E.P. Simoncelli, “Image quality assessment: from error visibility to structural similarity,” *IEEE Trans. Information Forensics and Security*, vol. 13, no. 4, pp. 600 –612, Apr. 2004.