Resampling Detection for Digital Image Forensics

John Ho, Derek Ma, and Justin Meyer

***Abstract*—A virtually unavoidable consequence of manipulations on digital images are statistical correlations introduced between the pixels. These correlations may not be visible to a human, but can be detected by statistical techniques. This paper presents a machine learning based approach to image resampling detection based on the detector by Popescu and Farid. We investigate ways to improve robustness and detection accuracy by using supervised learning techniques.**

I. Introduction

With the proliferation of digital images and powerful image editing tools such as Photoshop, it has become increasingly easy to manipulate images to alter content and meaning. Digital image forensics seeks to authenticate image based on statistical patterns left on an image by tampering. In [1], Popescu and Farid introduced a forgery detector by based on the correlations introduced between pixels by *resampling*; that is, the operation of stretching, shrinking, or rotating an image. Since objects in images are often on different scales, resampling is necessary to create a visually convincing forgery. However, resampling imposes periodic correlations between pixels that otherwise do not exist. These correlations can estimated by a learning algorithm.

In order to determine if the image has been resampled, the found correlations must be passed to a classifier. The classifier by Popescu and Farid characterized resampling by constructing a database of *synthetic maps* for different resampling ratios. This has the advantage of being conceptually simple, but requires an exhaustive database and does not take into account the fact that natural variations can be learned. In this paper, we show that a supervised classifier can improve the detector's accuracy and allow it to work in a much more general setting.

The paper is organized as follows. Section II describes the basis of correlations introduced by resampling and the EM procedure used to determine the correlation between pixels. Section III presents a supervised learning approach to classification. Finally, Section IV presents the results of our classifier and compares it with the performance by Popescu and Farid's classifer.

Submitted November 17, 2011 for CS229 Machine Learning, Prof. Andrew Ng, Stanford University.

II. Detecting Resampling Correlations

A. Resampling

Resampling causes certain pixels be a linear combination of its neighbors. These pixels are *correlated* with its neighbors and will appear periodically in the resampled image. For each pixel, we define its neighbors to be all pixels within a window of length $2N+1$.

Resampling by a factor p/q can be represented by the linear equation $\vec{y} = A_{p/q}\vec{x}$, where p and q are integers, $\vec{y}$ the resampled signal, $\vec{x}$ the original signal, and $A_{p/q} \in R^{n\times m}$ the resampling matrix.

For example, consider the case of $p/q = 2$. The resampling matrix is given by

$$A_{2/1} = \begin{bmatrix} 1 & 0 & 0 & \\ 0.5 & 0.5 & 0 & \\ 0 & 1 & 0 & \cdots \\ 0 & 0.5 & 0.5 & \\ 0 & 0 & 1 & \\ & \vdots & & \ddots \end{bmatrix}.$$

It can be seen that the odd samples do not change while the even samples are linearly dependent on their neighbors. We have

$$y_{2i-1} = x_i$$

$$y_{2i} = 0.5x_i + 0.5x_{i+1}$$

for $i = 1, \ldots, m$. This implies that

$$y_{2i-1} = 0.5y_{2i-1} + 0.5y_{2i+1}, \tag{1}$$

so, for this case, every even sample is exactly the same linear combination of its neighbors.

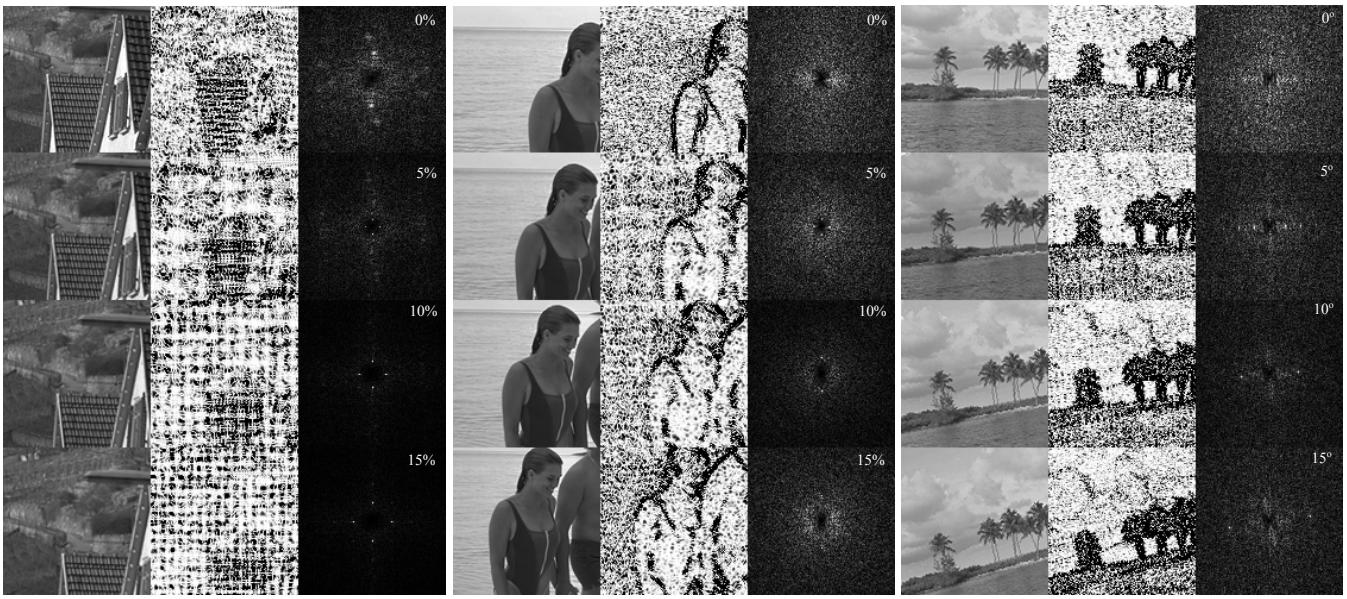


Fig. 1. Image, probability map, and periodicity map for (a) upsampling, (b) downsampling, and (c) rotation. The classifier must be able to distinguish between natural peaks in the image and those introduced by resampling.

To generalize this to a signal resampled by an arbitrary p/q it is necessary to determine when

$$y_i = \sum_{k=-N}^{N} \alpha_k y_{i+k} \tag{2}$$

where α_k is the set of weights across the neighborhood.

Let $\vec{a}_i$ be the i^{th} row of the resampling matrix $A_{p/q}$. It can be shown that (2) holds when

$$\vec{a}_i = \sum_{k=-N}^{N} \alpha_k \vec{a}_{i+k}. \tag{3}$$

That is, a resampled pixel y_i is correlated with its neighbors whenever its corresponding row of the resampling matrix a_i can be written as a linear combination of its neighboring rows [1][2]. This will lead to periodic correlations between resampled pixels since the resampling matrix is periodic.

B. Expectation-Maximization

If the resampling matrix $A_{p/q}$ is known, the problem is simply that of finding a set of weights $\vec{\alpha}$ that satisfies (3) for a subset of rows. However, in practice, the sampling factor p/q and interpolation method are not known. Popescu and Farid describe an expectation-maximization (EM) algorithm to learn the weights $\vec{\alpha}$.

Let M_1 be the set of samples that are correlated to their neighbors and M_2 the set of samples not correlated to their neighbors. EM is used to simultaneously estimate $\vec{\alpha}$ and a soft assignment $\Pr\{y_i \in M_1\}$ for each pixel y_i since neither is known. The output of the EM algorithm is a set of coefficients $\vec{\alpha}$ and a *probability map* P. The complete algorithm is given in Appendix A.

For faster computation of the probability map, the EM algorithm can be applied to only a random subset of the blocks in the image to train the weights $\vec{\alpha}$. Once the weights are weights are found, P can be quickly computed everywhere. We found that this does not significantly change the probability maps. Random block sampling was used throughout our experiments.

C. Periodicity

Neighboring pixels can be naturally correlated based on the statistics of the natural image. However, it is unlikely that these correlations are periodic. To detect periodicity, Fourier transform of the probability map is taken. We refer to the magnitude of the transform as the *periodicity map*. The periodicity map is typically high-pass filtered in order to suppress the natural DC peak. In [1], a peak detection algorithm is also described in order to enhance the peaks and suppress energy due to natural correlations in the image.

III. Supervised Learning

Given a periodicity map, we want to determine if the image has been resampled or non-resampled. In [1], an exhaustive database of synthetic maps was generated for various resampling ratios. A periodicity map was then classified by k-nearest neighbors (kNN) with $k = 1$ for a similarity score. While this yields reasonable accuracy, the kNN classifier does not work in a general setting with arbitrary block sizes and resampling ratios.

We train a support vector machine (SVM) to classify a periodicity map as resampled or non-resampled. Figure 1 shows the probability and periodicity map for different images and resampling ratios. There are distinct peaks that appear when an image has been resampled. The classifier must be able to distinguish between natural peaks in the periodicity map from those introduced by resampling.

We extract the following 5 features from the periodicity map:

1) n **largest coefficients**. Higher peaks tend to indicate resampling. The coefficients c_i are sorted by magnitude and aggregated to a single feature

$$f_1 = \sum_{i=1}^{n} \sqrt[m]{|c_i|}. \tag{4}$$

2) n **largest coefficients after peak detection**. The peak detection algorithm used in [1] is applied to the periodicity map. The sorted coefficients c_i' are aggregated by

$$f_2 = \sum_{i=1}^{n} \sqrt[m]{|c_i'|}. \tag{5}$$

3) n **largest coefficient distance from center**. Peaks in natural images tend to be concentrated in low frequencies. The positions (u_i, v_i) are used to compute

$$f_3 = \sum_{i=1}^{n} \sqrt{u_i^2 + v_i^2}. \tag{6}$$

4) n **largest coefficient to local energy ratio**. The ratios r_i are computed for a local rectangular window of width W and aggregated by

$$f_4 = \left(\sum_{i=1}^{n} 100\, r_i \right)^m / 100. \tag{7}$$

TABLE I
Training Resampling Rates.

	Start	Step Size	End
Upsampling ratio	1.05	0.03	1.5
Downsampling ratio	0.5	0.03	0.95
Rotation (°)	1	1	45

5) **Peak to total energy ratio**. The ratio R was scaled by

$$f_5 = (10^4 R)^m / 10^2. \tag{8}$$

Since all the features are non-negative, they are scaled to $[0, 1]$. We selected $n = 4$ coefficients and a scaling factor of $m = 2$. LIBSVM was used with the radial basis kernel [3].

A. Experiment Setup

For the test images, we use the uncompressed Kodak PhotoCD set of 24 images. These images are not compressed and are true color; that is, free of demosaicking, which can also introduce linear correlations [4]. For simplicity, we extract the green color channel from the image to obtain a grayscale test image. We used a 128×128 window size, which is significantly smaller than 512×512 used in [1]. Smaller block sizes can better localize tampering, but result in peaks that are less distinct.

For each resampling operation, we first randomly select an image on which to apply resampling and then randomly select a window within the resampled image. The resampling ratios used in training are shown in Table I. In order to obtain non-resampled samples, the images are divided into non-overlapping blocks. We used 80% of the images for training and 20% for testing. Since false positives are highly undesirable in a forensic setting, the parameters of the classifier are set to have no false positives over the training set.

We also implemented the kNN classifier in [1] for comparison. The database consists of synthetic maps using the same resampling ratios in Table I.

IV. Results

A. Classification Results

The overall recall and precision rates for our classifier are shown in Table II. There is only one false positive ($< 1\%$) over the test set. The performance of the SVM classifier is compared to the kNN

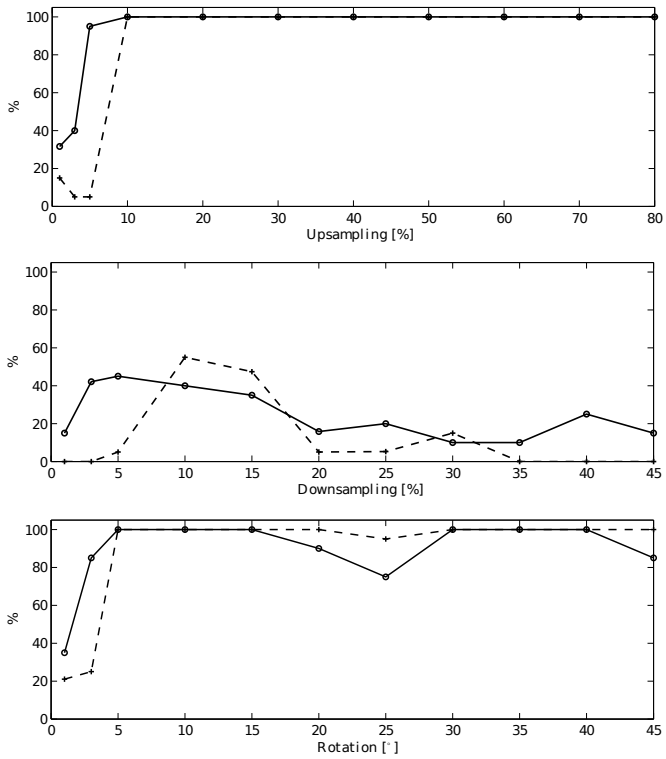


Fig. 2. Recall rate for the SVM classifier (black solid) and the kNN (dashed gray) classifier. The false positive rate for both cases are $< 1\%$.

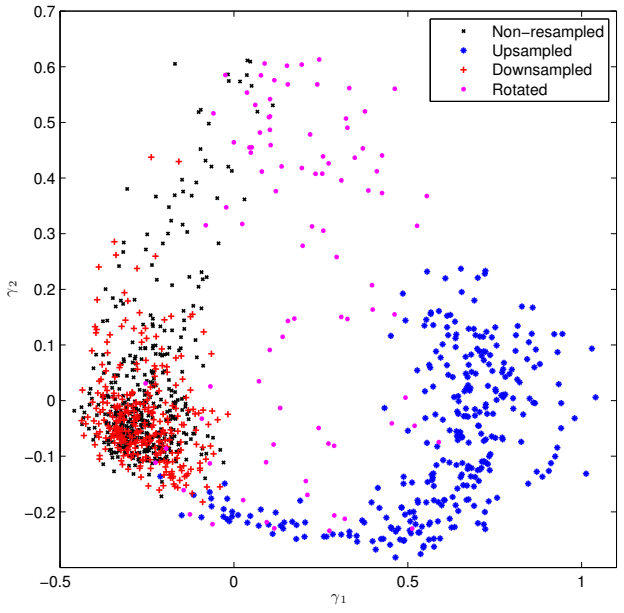


Fig. 3. Training set features projected onto the first two principal components γ_1 and γ_2. The upsampled and rotated data is clearly separable from the non-resampled data even in low dimension, but downsampled data is difficult to separate.

TABLE II
RECALL AND PRECISION RATE FOR SVM CLASSIFIER.

	Non-resampled	Resampled
Recall (%)	119/120 99.16%	369/394 93.66%
Precision (%)	119/144 82.639%	369/370 99.730%

classifier over a wide range of resampling ratios in Figure 2. The threshold of the synthetic classifier was set to have a similar $< 1\%$ false positive rate. Our classifier clearly outperforms kNN at low resampling rates, which compared to results in [1] suffers from the smaller block. It has comparable accuracy to kNN at large resampling ratios and can detect some highly downsampled images where the kNN classifier has zero recall.

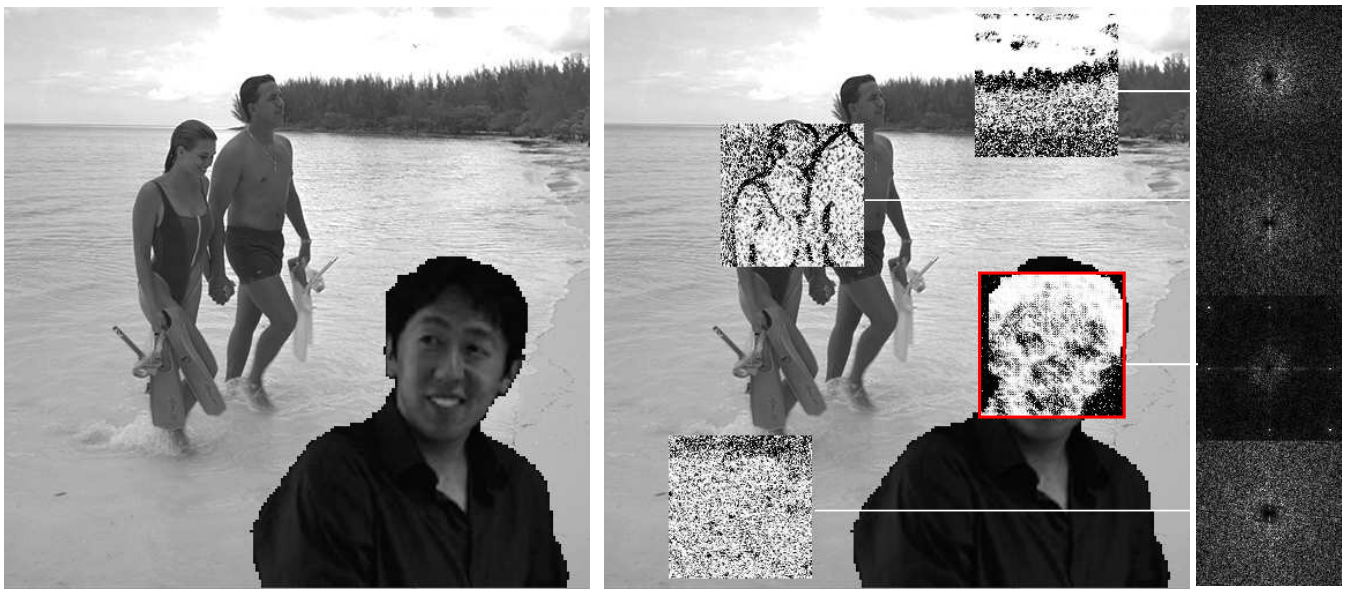
Figure 3 shows the training set features projected onto the first two principal components. The upsampled and rotated data are clearly separable from the non-resampled data, but the down-sampling data appears to be clustered. This is due to fact that downsampling peaks in the periodicity map are indistinct and can be buried in the natural image correlations. Interestingly, it also appears that upsampled and rotated images can be distinguished from one another.

B. Detection Example

Figure 4 shows how the SVM classifier could be used to perform tamper detection in a more general setting. A beach scene contains an image of Prof. Andrew Ng of Stanford University that has been upsampled and inserted into the foreground. Several regions in the image are selected over which the periodicity maps are computed. The features are extracted from the periodicity maps and passed to our classifier, which determines if the region has been resampled. The classifier successfully distinguishes the resampled and non-resampled regions. If not computationally prohibitive, a sliding window could be used to perform detection at every location in the image to enable automatic detection without user input.

V. CONCLUSION

In this project, the Popescu and Farid's EM algorithm for learning correlations between pixels



(a) (b)

Fig. 4. (a) Tampered image and (b) probability map and periodicity map for four blocks. The classifier correctly determines that the red block has been resampled and the remaining blocks are not resampled.

was implemented and tested for a smaller block size. An SVM classifier was trained to determine if the correlations found by the EM algorithm result from resampling. This classifier was shown to have better performance than the kNN classifier at low resampling rates and does not require and exhaustive database of synthetic maps.

APPENDIX A

The *expectation* step consists of assigning

$$\Pr\{y_i \in M_1 | y_i\} = \frac{\Pr\{y_i | y_i \in M_1\}\Pr\{y_i \in M_1\}}{\sum_{k=1}^{2} \Pr\{y_i | y_i \in M_k\}} \tag{9}$$

where $\Pr\{y_i \in M_1\} = \Pr\{y_i \in M_2\} = 1/2$.

The model M_1 is assumed be normally distributed

$$\Pr\{y_i | y_i \in M_1\} = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(y_i - \sum_{k=-N}^{N} \vec{\alpha}_k y_{i+k})^2}{2\sigma^2} \right) \tag{10}$$

and M_2 uniformly distributed $\Pr\{y_i | y_i \in M_2\} = 1/N$ where N is the range of the pixel values.

The *maximization step* reduces to finding the parameters $\vec{\alpha}$ by weighted least-squares

$$\vec{\alpha} = (Y^T W Y)^{-1} Y^T W \vec{y} \tag{11}$$

where the rows of Y are the blocks in the image with the center value removed and W is a diagonal matrix with diagonal elements equal to $\Pr\{y_i | y_i \in M_1\}$.

The EM steps are iterated until $||\vec{\alpha}^{(i)} - \vec{\alpha}^{(i-1)}|| < \epsilon$ where $\vec{\alpha}^{(i)}$ is the set of weights obtained from the i^{th} iteration.

REFERENCES

[1] A. Popescu and H. Farid, "Exposing digital forgeries by detecting traces of resampling," *Signal Processing, IEEE Transactions on*, vol. 53, no. 2, pp. 758 – 767, feb. 2005.

[2] M. Kirchner, "Fast and reliable resampling detection by spectral analysis of fixed linear predictor residue," in *Proceedings of the 10th ACM Workshop on Multimedia and Security*, ser. Sec '08. New York, NY, USA: ACM, 2008, pp. 11–20.

[3] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, pp. 27:1–27:27, 2011, software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.

[4] A. Popescu and H. Farid, "Exposing digital forgeries in color filter array interpolated images," *Signal Processing, IEEE Transactions on*, vol. 53, no. 10, pp. 3948 – 3959, oct. 2005.