Detection of Adaptive Histogram Equalization Robust Against JPEG Compression

*Mauro Barni, Ehsan Nowroozi, Benedetta Tondi
Department of Information Engineering and Mathematics, University of Siena
Via Roma 56, 53100 - Siena, ITALY
barni@dii.unisi.it, ehsan.nowroozi@student.unisi.it, benedettatondi@gmail.com

***Abstract*— Contrast Enhancement (CE) detection in the presence of laundering attacks, i.e. common processing operators applied with the goal to erase the traces the CE detector looks for, is a challenging task. JPEG compression is one of the most harmful laundering attacks, which has been proven to deceive most CE detectors proposed so far. In this paper, we present a system that is able to detect contrast enhancement by means of adaptive histogram equalization in the presence of JPEG compression, by training a JPEG-aware SVM detector based on color SPAM features, i.e., an SVM detector trained on contrast-enhanced-then-JPEG-compressed images. Experimental results show that the detector works well only if the Quality Factor (QF) used during training matches the QF used to compress the images under test. To cope with this problem in cases where the QF cannot be extracted from the image header, we use a QF estimation step based on the idempotency properties of JPEG compression. Experimental results show good performance under a wide range of QFs.**

***Index Terms*— Multimedia forensics, Histogram equalization detection, Adversarial multimedia forensics, JPEG quality factor estimation.**

I. INTRODUCTION

When creating a forgery, contrast enhancement (CE) is often used to adjust the contrast and lighting conditions of image subparts. The detection of this manipulation has thus been widely studied in image forensics, and, more recently, in scenarios encompassing the presence of an adversary, e.g., in adversarial image forensics [1], [2]. Due to the peculiar traces left by contrast enhancement operators in the image histogram, most early works were based on the analysis of image first order statistics [3]–[5]. Anti-forensic methods have been developed as well; in addition to targeted approaches, aiming at removing the specific histogram artifacts the attacked detectors look at [6], universal approaches against generic histogram-based detectors have also been developed with good results [7]. Expectedly, it is quite easy to cope with such attacks by developing detectors based on second-order statistics [8], [9]. Such ad-hoc detectors, however, fail when different attacks are considered. Since, in real applications, the attack is not known in advance, targeted anti-counter-forensic methods are of little help. Moreover, in most cases, the attack consists in the application of one (or several) post-processing operations, e.g., a geometric transformation, filtering or compression. Coping with such attacks often referred to as laundering attacks, turns out to be a very challenging task. Since in most applications images are stored and distributed in JPEG format, JPEG compression is one of the most common laundering attacks contrast-enhanced images are subject to. Unfortunately, the performance of CE detectors proposed so far tend to decrease significantly in the presence of even mild post-processing and, in particular, all of them exhibit a poor robustness against JPEG compression [3], [5], [10]–[12], even when the compression is weak. Poor resilience to post-processing, and in particular to JPEG compression, is a common problem of state-of-the-art detectors of CE, and, to the best of our knowledge, it has not been addressed yet.

In order to cope with this problem, in this paper we propose an adversary-aware data-driven CE detector, inspired by [13], where an SVM detector is trained to recognize a specific class of attacks, corresponding, in our case, to JPEG compression. Specifically, a pool of JPEG-compression-aware SVM CE-detectors is trained for different values of the JPEG quality factor (QF) of the attacked images. Given a test image, the QF can be extracted from the header of the image bitstream and the most suitable SVM detector used to decide whether the image has been contrast-enhanced or not. Such an approach is obviously prone to attacks, since forging the header or re-saving the image in an uncompressed format (e.g. PNG or bitmap) would prevent the identification of the QF used to compress the image (or even to understand that the image has been compressed) forcing the system to select a wrong SVM detector. To prevent this problem, we devised a refined scheme that does not rely on the information contained in the image header, thus working only on image pixels. The refined system works as follows: it first gets an estimate of the QF by exploiting the idempotency property of JPEG compression, that is, the fact that JPEG compression with the same QF is an (almost) idempotent operation [14]; then, such an estimate is used to choose the proper SVM, i.e., the one that was trained on the QF closest to the estimated one.

Among contrast enhancement operators, we focused on Adaptive Histogram Equalization (AHE), which applying contrast enhancement on a local basis. To the best of our knowledge, the detection of such a local CE operator has not been addressed so far. Besides, it is more challenging than the detection of global CE operators (like for instance gamma correction and histogram stretching), since it does not

*The list of authors is provided in alphabetic order. The corresponding author of the paper is Ehsan Nowroozi.

introduce easily identifiable artifacts in the image histogram.

Regarding the feature set, we considered residual-based features, that is features extracted by high-pass filtering the image [15]. Such features have been recently used to detect several types of image processing operations [5], [16]. In particular, since we focus on color images, we considered a feature model, inspired by the CRMQ1 model proposed in [17], which also takes into account the pixel relationships among the color channels. As contrast enhancement modifies the inter-channel relationships among pixels [10], in fact, possibly useful information may be discarded by converting to grey-scale or considering the luminance channel only.

Experiments show that our system provides improved performance in the presence of JPEG compression over a wide range of QFs, while it maintains good performance in the absence of attacks, that is when the AHE is the last step of the manipulation chain.

As the further contribution, we also assess the performance of our system when JPEG compression is carried out with a different software concerning the one used to generate the training images and to perform the QF estimation (JPEG-compression software mismatch).

The paper is organized as follows: in Section II, we define the detection task addressed in the paper and describe the proposed JPEG-aware detector. In Section III, we describe the methodology we followed for conducting our experiments. The results of the experiments are discussed in Section IV. Conclusions and some considerations on future work are finally given in Section V.

II. PROPOSED SYSTEM

Our goal is to design a detector to reveal if an image has undergone contrast enhancement even when the enhanced image is JPEG compressed. Specifically, we focus on contrast enhancement using Adaptive Histogram Equalization [18]. In the following, we first formalize the detection problem, then we describe our choice of the feature set and present the architecture of the detector, which is based on a pool of Support Vector Machines (SVMs) trained in an adversary-aware modality.

The detection task is schematised in Figure 1. We let hypothesis H_0 correspond to the case of pristine images and H_1 to the case of contrast enhanced images. In both cases, the images are JPEG compressed at the end (post-processing operation) with a given Quality Factor (QF). In this scheme, JPEG compression can also be viewed as a counter-forensic, laundering-type, attack, due to its effectiveness in erasing the traces of contrast manipulations [3], [5], [10]–[12].

A. THE CSPAM FEATURE SET

For our detection task, we need to select a sufficiently large number features which are capable of capturing peculiar types of dependencies among neighboring pixels. On the other hand, we want to limit the dimensionality of the feature set, so to be able to train an SVM. In fact, using a very large feature set could provide better modeling capabilities, but it would require resorting to multiple classifier approaches (e.g., ensemble classifiers) [19], which are more difficult to train, especially in the adversary-aware modality. Residual-based features (e.g., [15], [20]) have been largely used for the detection of a wide variety of global manipulations [11]. For a given choice of the residual, the feature set is computed by evaluating the residual noises in all the directions (horizontal left, horizontal right, vertical left, vertical right, diagonal left, diagonal right), truncating the values at a certain T and then computing the co-occurrences of order d. However, these feature extractors, widely used in forensics, are designed for grayscale images and cannot be directly applied to color images. In this case, a possibility would be to extract the features on the luminance channel; however, contrast enhancement also modifies the relationship among color channels, and hence considering the luminance only (or, similarly, converting the images to grayscale), would result in a loss of possibly useful information. To take into account the relationships among color channels, we considered the rich model for color images proposed in [17] for staganalysis (CSRMQ1), and adapted it to our case. Basically, the rich color feature space proposed in [17] consists of two different components. The first component is derived from the spatial rich model as in [15] (SRMQ1): specifically, the SRMQ1 features are computed for each color channel and added to keep the same dimensionality of grayscale images. The second component is a collection of 3-D color co-occurrences, computed from the same noise residuals as for the SRMQ1 model but formed across the three channels of each pixel[1]. For more details, the reader may refer to [17]. In its complete form, the SRMQ1 model considers many different types of residuals and then the final feature space has a very large dimensionality (it consists of 12.753 features), which cannot be adopted for standard detectors based on a single classifier. Therefore, in our case, we adopted a new feature model by using the SPAM (Subtractive Pixel Adjacency Matrix) feature set [20] as the base set. More specifically, according to the SPAM model, the first component of the feature vector is obtained by considering the second-order co-occurrences (i.e., $d = 2$) of the first order residuals, with a truncation parameter $T = 3$, computed for each channel and then merged. For the second component, the residual co-occurrences are computed with respect to the three channels.

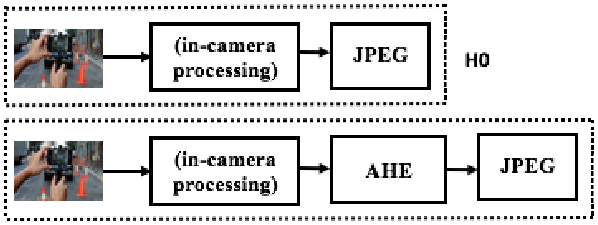


Fig. 1: Detection task considered in this paper: adaptive histogram equalization detection in the presence of JPEG compression.

[1] These are always second-order co-occurrences ($d = 2$).

We call CSPAM this simplification of the CSRMQ1 feature set. Since the dimensionality of the SPAM set is 686, the final dimensionality of the CSPAM set is $2 \times 686 = 1372$.

B. ADVERSARY-AWARE DETECTOR

Similarly to the what happens with the CE detectors proposed in the literature, if we train the SVM classifier based on the CSPAM feature set on pristine and enhanced images (without taking into account the JPEG compression in the end), the detector can correctly reveal the enhancement in the ideal scenario in which CE is the last step of the manipulation chain, but it completely fails in the presence of JPEG post-processing, even when JPEG compression is very mild (high-quality factors).

To design a contrast enhancement detector robust to JPEG laundering attack, we trained several adversary-aware versions of the SVM classifier, where the classifier is trained with JPEG compressed images on one hand (H_0) and images subject to contrast enhancement followed by JPEG compression with different QFs on the other hand (H_1).

The overall architecture of the detector is reported in Figure 2. For a given image, the value of the QF used for JPEG compression can be easily extracted from the quantization table provided in the header of the image bitstream and used to select the most suitable version of the SVM classifier (i.e. the one trained with the QF which is most similar to the one used to compress the test image).

In principle, we should train an SVM for any value of the QF and then, given the image QF, use the corresponding SVM model for testing[2]. However, similar results can be obtained by training a lower number of SVMs for some selected values of QF and then using the SVM corresponding to the closest QF. By referring to Figure 5, we see that the performance decay rather slowly (in terms of AUC) when the QF of the test image departs from the QF used for training (matched value). Notice also that, not surprisingly, the performance in the matched case increases for larger values of QF, since a weaker compression is less effective in erasing the traces of AHE. Based on our tests, we argued that a quantization step equal to 5 for medium-high QF values, and 2 for very high QFs is appropriate. Then, we built our classifier by considering the 6 SVM models reported in the scheme of Figure 2.

1) Idempotency-based QF estimation: As we said, given a JPEG image, the QF can be almost perfectly estimated from the JPEG bitstream. However, if the compressed image is re-saved in uncompressed format (e.g., png, bitmap), or the image header is manipulated, it is clear that the proposed detector does not work. In this case, in fact, an SVM with a large QF mismatch would likely be chosen thus significantly impairing the detection performance. Therefore, as a further contribution, we propose an algorithm to estimate the QF directly from image pixels, thus extending the applicability of our detector. Another possibility to design a system which is robust to attacks would be to train a single SVM classifier for all or some selected values of the QF. However, based on our experiments, by following this approach, we get lower performance.

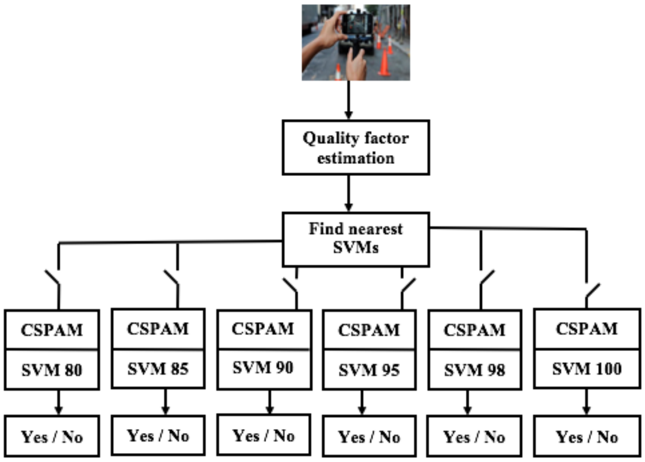


Fig. 2: Scheme of the proposed JPEG-aware detector.

In order to get an estimate of the QF from the pixel domain, we propose to exploit the fact that JPEG compression is an (almost) *idempotent* operator, that is, whenever applied multiple times with the same QF, it produces the same result obtained with a single application. A similar property has been exploited in [14] for video codec identification. Our algorithm works as follow: first, the image under analysis is compressed with various QFs[3], then the value leading to the minimum distance between the images before and after compression is searched for (the $L1$ distance is adopted as distance measure). In particular, we identified a critical QF value, say QF*. A local minimum is then searched below QF* (and above 50). If no local minimum is found inside this range, then a finer search is performed over the QFs larger than QF*, and up to 98. For higher QFs, namely 99 and 100, the system guess is always 98.[4] We point out that, as a consequence, when QF estimation is performed in the pixel domain with the idempotency-based algorithm, the SVM model for QF=100 (SVM100 in Figure 2) is never selected by the detector. We experimentally set QF* = 93.

III. EXPERIMENTAL METHODOLOGY

To produce the datasets for our experiments, we started from color images in uncompressed (TIFF) format, part of which used for training and part for testing. The images for the H_0 and H_1 classes were built as detailed in the scheme in Figure 1. The images were JPEG compressed with quality factor QF for producing the H_0 samples, while, the images for the H_1 class were generated by first applying

[2] Since we are interested in medium-high quality factor, we take $QF > 80$ (much lower QF are not very common in practice since the visual image quality degrades too much).

[3] We start from QF = 50 assuming that the image is never compressed with a lower QF.

[4] Very high values of QF (99 and 100) are difficult to estimate with no errors. However, the detection performance of our system training in these cases are generally very good (see Figure 5), so getting a very accurate estimation in this range is not of primary importance.

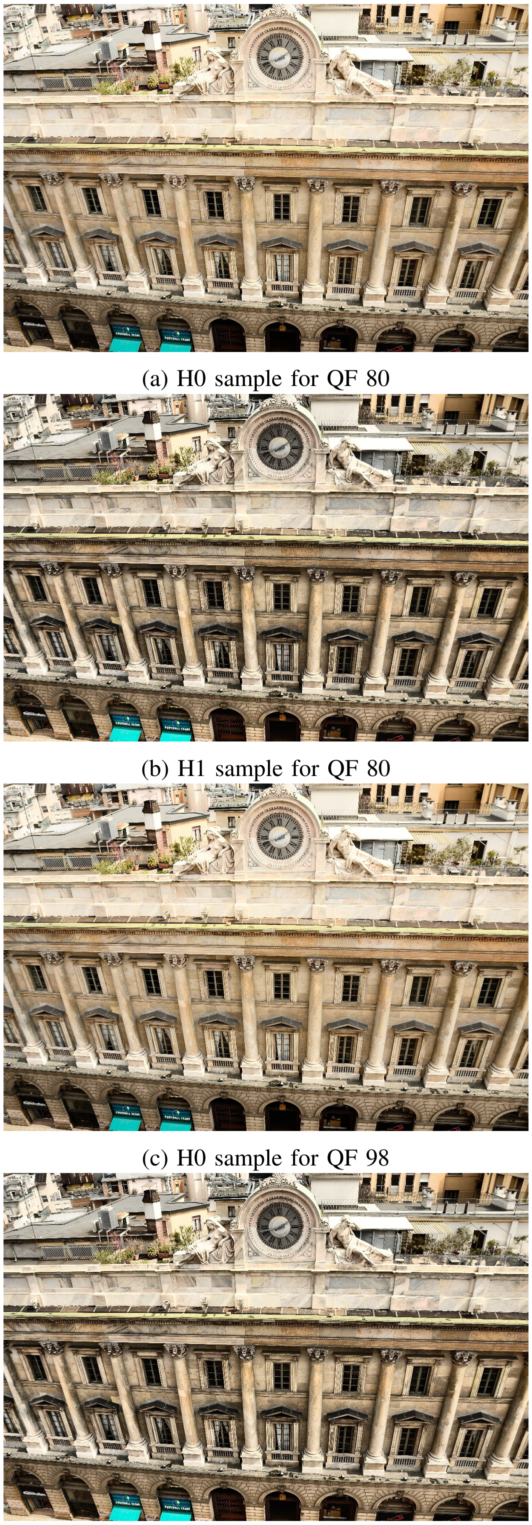
the AHE operator and then compressing them with quality factor QF (the same as for H_0). For the unaware case, the training images were built according to the same scheme but without considering the JPEG compression stage at the end. The Contrast-Limited implementation of AHE (CLAHE) was used for contrast enhancement [21]. With respect to the ordinary AHE, CLAHE prevents the overamplification of noise (that adaptive histogram equalization can give rise to) in relatively homogeneous regions. This is done by clipping the histogram at a predefined value before computing the cumulative distribution function (CDF); this limits the slope of the transformation function (given by the CDF) which determines the contrast amplification. The value at which the histogram is clipped, called clip limit, depends on the normalization of the histogram and thereby on the size of the neighborhood region, which by default is 8×8. On color images, the straightforward application of CLAHE to each channel separately unnaturally changes the color balance and produces a visually unpleasant image. A common strategy is to convert the image from the RGB to the HSV color space and then applying CLAHE only to the luminance channel, namely the V channel. Then, the image is converted back to the RGB domain. We then followed this strategy to produce the AHE manipulated images in our case. In our experiments, the clip limit parameter for CLAHE is set to 0.004 (resulting in a not too strong enhancement). Some sample of images for the hypotheses H_0 and H_1 are provided in Figure 3 for QF equal to 80 and 98. Regarding QF values, the images used for training (both under H_0 and H_1) were compressed with $QF \in \{80, 85, 90, 95, 98, 100\}$, whereas for the test images all the QFs in the range $[80, 100]$ were considered.

The Matlab environment was used to process the images, to train and test the SVMs (with the LibSVM library package [22]) and run the idempotency-based QF estimator. In our tests, we also considered the GIMP software (and also Photoshop) for compressing the test images, in order to assess the performance of the detector in the presence of a mismatch in the compression software.

Each SVM classifier was fed with the 1372-dimensional features (CSPAM) extracted from the color images. A Gaussian kernel was adopted, and the kernel parameters were determined by 5-fold cross-validation. In the unaware case, we trained the SVM with uncompressed pristine and contrast manipulated images. In the aware case, to build the pool of SVMs detectors depicted in Figure 2, we separately trained the 6 SVMs (namely SVM80, SVM85, SVM90, SVM95, SVM98 and SVM100) on the corresponding JPEG compressed versions of the images.

IV. EXPERIMENTAL RESULTS

We considered uncompressed (TIFF) images taken from the RAISE8K dataset [23], consisting of camera-native images. Specifically, we built our dataset as follows: 6000 images were used for the training set (1000 of which were used for tuning the kernel parameters, i.e., for internal cross-validation) and 1997 images for the tests. To get a faster feature computation, the images were subsampled to a size



(a) H0 sample for QF 80

(b) H1 sample for QF 80

(c) H0 sample for QF 98

(d) H1 sample for QF 98

Fig. 3: Visual comparison between an H_0 and H_1 sample for two different QFs.

QF	80	81	82	83	84	85
AUC	0.5441	0.5429	0.5415	0.5390	0.5378	0.5370
QF	86	87	88	89	90	91
AUC	0.5331	0.5310	0.5287	0.5274	0.5242	0.5199
QF	92	93	94	95	96	97
AUC	0.5184	0.5118	0.5081	0.5027	0.4955	0.4871
QF	98	99	100			
AUC	0.4754	0.4570	0.4507			

TABLE I: AUC values of the unaware SVM classifier.

1072x770. Concerning the system hardware, we run our experiments on an Intel(R) Core(TM) i7-6700 CPU @ 3.40 GHz with four cores, 32 GB of RAM and with graphics card NVIDIA Geforce GT 730 (no GPU used).

A. RESULTS IN THE UNAWARE CASE

In this section, we show the results of unaware classification. The unaware SVM can perfectly classify uncompressed pristine and manipulated images, and the Area Under Curve (AUC) of the ROC curve for the classification is 100%. We also run some tests in the presence of laundering attacks, that is when both the pristine and manipulated images are subject to post-processing operations. In particular, we considered a case of filtering (median filtering with window size 3x3) and geometric transformations (resize with scaling factor 0.9, rotation with an angle of 5 degrees). In all these cases, the performance only slightly decreases, and the AUC remains above 90%. This shows that the CSPAM feature set that we defined is very discriminative for our classification task. When JPEG laundering is considered, however, the detector fails to classify the images, thus confirming that the JPEG compression is very harmful. Table I shows the detection performance of the unaware SVM in this case.

B. PERFORMANCE OF THE AWARE DETECTOR

We now focus on the results achieved by the aware detector illustrated in Figure 2. Figure 4 shows the performance of each SVM classifier tested under matched condition, that is, when the images considered for training and testing are compressed with the same QF. The performance improves significantly with respect to the unaware case. We observe that performance reduces when the QF decreases. This is expected since the lower the QF, the more the traces of AHE are erased by compression, and the detection task becomes harder. Arguably, for much lower quality factors, it is possible that the traces are almost completely erased; however, the quality of the images would also be seriously impaired.

In Figure 5 we report the results of the 6 SVMs for QF $\in$ [80,100]. The performance of the system based on the pool of aware SVM classifiers when the QF value is extracted from the JPEG header (i.e., perfect estimation) can be easily argued from these plots by considering for each image the closest QF value in the set {80,85,90,95,98,100} and then select the corresponding SVM for testing. They are reported in Figure 6. From Figure 5, we also observe that using the minimum distance criterium for the SVM selection is a good choice. We also verified that by training only one SVM considering all the 6 QF values above, the performance degrades significantly (the average AUC is 87%).

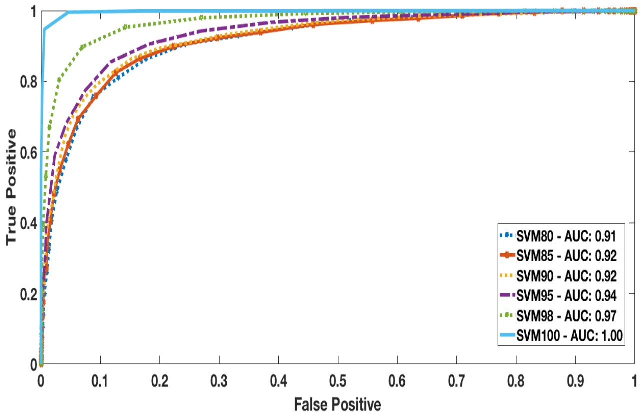


Fig. 4: Performance of the aware SVMs for the classification task (under matched QF).

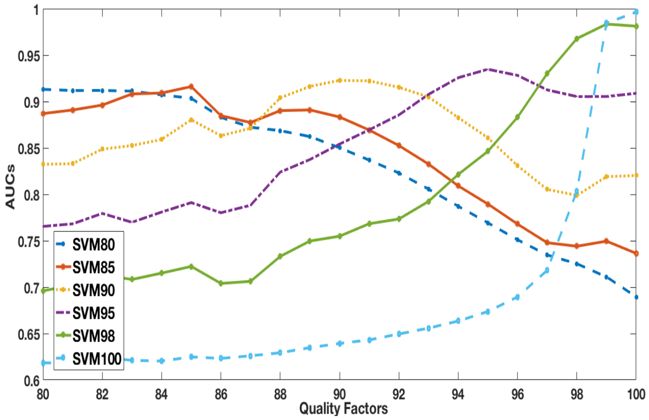


Fig. 5: Performance of the SVMs as a function of the QF.

When the QF is estimated on the pixel image according to the proposed idempotency-based approach, the performance is expected to decrease because of estimation errors. The average error in terms of L1 distance between real and estimated QFs under H_0 and H_1 is reported in Table II and Table III respectively. The average is computed on the 1997 images in the test set. The performance of the idempotency-based algorithm are pretty good, always leading to an average estimation error below 0.1% for every $QF \leq 98$ Note that the average errors equal to 1 and 2 obtained, respectively, for QF = 99 and 100 are expected, given that for such QFs the algorithm always decide for 98 (see discussion in Section II-B.1). Besides, we observe that the average error in the various cases is slightly larger under H_1 than under H_0. Figure 7 shows the performance of the system when QF estimation is based on JPEG-idempotency. The performance reduction with respect to the case of perfect QF estimation is pretty slight (of order 10^{-3} on the average) and, expectedly (see the discussion in Section II-B.1), pertains mainly to the case of higher QF (99 and 100), where, however, the performance of the detector remains very good. We also verified that the

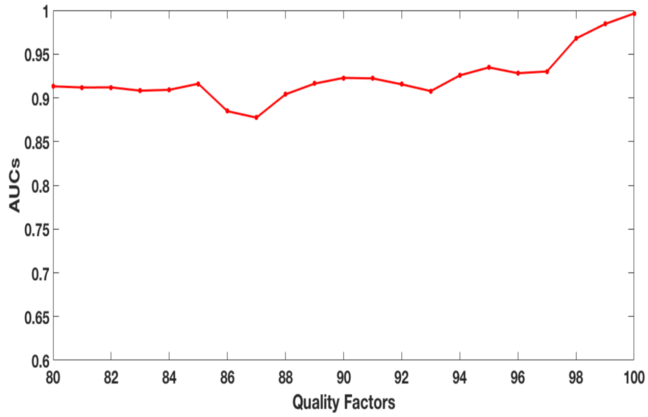


Fig. 6: Performance of the system based on the pool of aware SVM classifiers.

QF	80	81	82	83
Average Error	0.0856	0.1002	0.0586	0.0676
QF	84	85	86	87
Average Error	0.0656	0.0976	0.0190	5.0075e-04
QF	88	89	90	91
Average Error	0.0010	5.0075e-04	5.0075e-04	0
QF	92	93	94	95
Average Error	0	0.1202	0.1022	0.0315
QF	96	97	98	99
Average Error	0.0220	0.0015	0	1
QF	100			
Average Error	2			

TABLE II: Average error of QF estimation for H_0.

performance for the case of uncompressed images remains good. In this case, as a result of the idempotency-based QF estimation, the SVM98 is always selected. The AUC is 97,2% and then the performance reduction with respect to the unaware case is very small.

C. PERFORMANCE IN THE PRESENCE OF SOFTWARE MISMATCH

All the results reported so far were obtained by working in the Matlab environment. However, the performance can be sensitive to a mismatch of the software used for compression, as different software may use different JPEG quantization tables. We may expect that this is especially the case when the idempotency-based QF estimator is used; in this case, in fact, software mismatch might also lead to a wrong estimation. Then, we also assessed the performance of the detector when the software used for compressing the test images is different from the one used to compress the images for training and inside the idempotency-based estimator. In particular, we used the GIMP software for JPEG compression of the test images[5].

QF	80	81	82	83
Average Error	0.1017	0.1202	0.0686	0.0721
QF	84	85	86	87
Average Error	0.0631	0.1107	0.0325	0.0015
QF	88	89	90	91
Average Error	0.0015	0.0010	0.0010	0
QF	92	93	94	95
Average Error	0	0.1778	0.1402	0.0451
QF	96	97	98	99
Average Error	0.0451	0.0015	0	1
QF	100			
Average Error	2			

TABLE III: Average error of QF estimation for H_1.

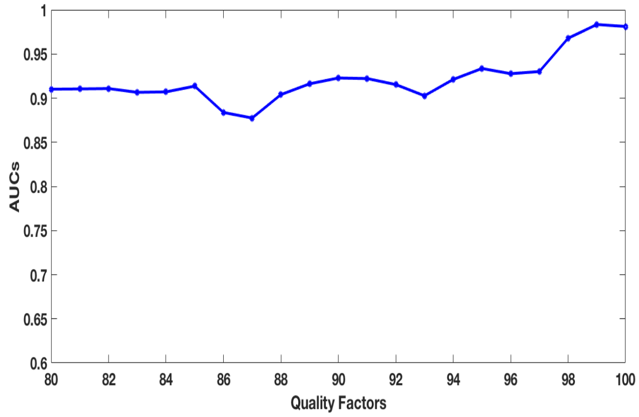


Fig. 7: Performance of the system based on the pool of aware SVM classifier when the QF is estimated by means of idempotency.

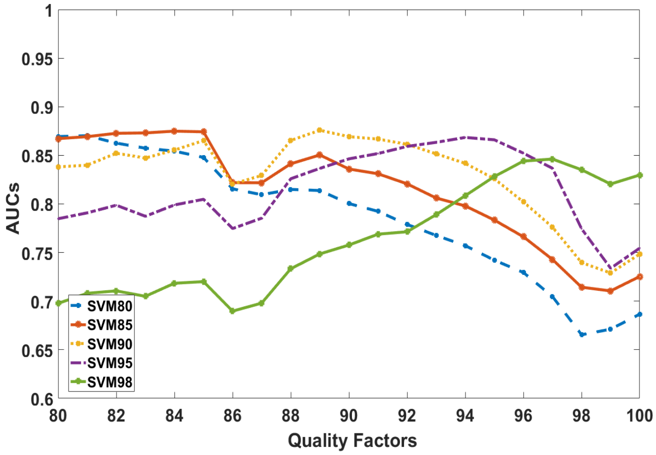


Fig. 8: Performance of the SVMs as a function of the QF in the presence of software mismatch. The test images are compressed with GIMP while the images used for training are compressed in Matlab.

Figure 8 shows the results of the 5 SVMs for QF ∈ [80,100]. When the QF value can be read from the JPEG header (i.e., perfect estimation), the performance of the overall system based on the pool of aware SVM classifiers can be easily derived from these plots. The performance reduces with respect to the matched case, although not drastically so (the AUC always remains above 80%). The average error in terms of L1 distance between real and estimated QF is reported in Table IV and Table V under

[5] This corresponds to implement the last step of the processing chain in Figure 1 with the GIMP software instead than with Matlab.

QF	80	81	82	83
Average Error	0.0200	0.0250	0.0130	0.0090
QF	84	85	86	87
Average Error	0.0100	0.0195	5.0075e-04	0
QF	88	89	90	91
Average Error	0	0	0	0
QF	92	93	94	95
Average Error	0	0.0401	0.0401	0.0045
QF	96	97	98	99
Average Error	0.0030	0	0	1
QF	100			
Average Error	2			

TABLE IV: Average error of the QF estimation for H_0 in the presence of software mismatch (test images compressed with GIMP).

QF	80	81	82	83
Average Error	0.0250	0.0300	0.0100	0.0090
QF	84	85	86	87
Average Error	5.0075e-04	0.0260	5.0075e-04	0
QF	88	89	90	91
Average Error	0	0	0	0
QF	92	93	94	95
Average Error	0	0.0376	0.0200	0.0030
QF	96	97	98	99
Average Error	0.0030	0	0	1
QF	100			
Average Error	2			

TABLE V: Average error of the QF estimation for H_1 in the presence of software mismatch (test images compressed with GIMP).

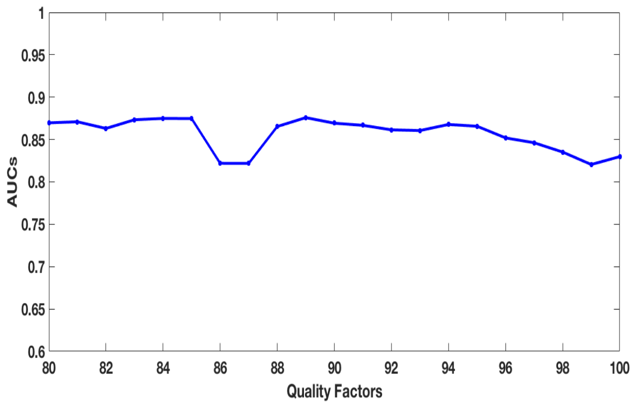


Fig. 9: Performance of the detector in the presence of software mismatch. The test images are compressed with GIMP while the compression of the images used for training and the idempotency-based estimation is implemented in Matlab.

H_0 and H_1 respectively. By comparing these results with those reported in the previous section for the matched case, we notice that the estimation is still very good.

Figure 9 shows the performance of the proposed detector in the presence of software mismatch when the QF estimation is made in the pixel domain by means of the idempotency-based approach.

We also considered a mismatched case in which the Photoshop software is used (instead of GIMP) for compressing the test images. In this case, the idempotency-based approach has poorer performance with respect to the GIMP software case. By focusing on the Photoshop qualities 10, 11, 12 for the compression (which correspond to medium-high values of QF)[6], the performance (AUC values) in the case of QF estimated from the image pixels are 79, 85 and 90 respectively.

V. CONCLUDING REMARKS

Detection of contrast-enhanced images in the presence of JPEG post-processing is known to be a hard task. This is a serious problem since JPEG compression is often the last step in any processing chain and also because an attacker may use JPEG compression as a laundering attack. In this paper, we used adversary-aware training to cope with this problem. The performance of the aware detector is very good when the QF used during training matches the QF used to compress the images under test, but decrease significantly when the two QFs are not matched. Hence, we proposed a system where an SVM detector is chosen, within a pool of detectors trained with different QFs, according to the detector's estimate of the QF used to compress the test image. When such a QF cannot be reliably extracted from the image header, the detector estimates the QF by exploiting the approximate idempotency of JPEG compression. Experimental results prove that the proposed system provides good performance for a wide range of QFs (larger than 80) also in the presence of software mismatch.

Our system is based on a set of image features obtained by adapting the feature set proposed in [17], however we did not make any attempt to optimise the feature set to the problem at hand (as done in [11], [24] for the case of SRMQ1 features [15]), hence it is possible that better results could be obtained by using a different feature set (e.g., by using higher order residuals).

Throughout the paper, we focused on the detection of adaptive histogram equalization (namely CLAHE), a problem that got lesser attention and is substantially more difficult than the detection of global operators based, for instance, on histogram stretching or gamma correction. As a further work, we could evaluate the generalization capability of the detector trained on CLAHE when used to detect other kinds of contrast enhancement operators.

Eventually, we would like to apply our approach to CNN (Convolutional Neural Networks) detectors, which have been proven to provide significantly better results on other multimedia forensic applications.

ACKNOWLEDGMENTS

This work was supported partially by Defense Advanced Research Projects Agency (DARPA) and Air Force Research Laboratory (AFRL) under the research grant number

[6] In Photoshop, the strength of the compression is determined by setting a parameter for the image quality in a (non linear) scale from 0 to 12.

FA8750-16-2-0173. The United States Government is certified to reproduce and distribute reprints for Governmental objectives notwithstanding any copyright notation thereon. The views and conclusions consist of herein are those of the authors and should not be explained as necessarily representing the official policies or authorization, either expressed or implied, DARPA and AFRL or U.S. Government.

References

[1] R. Böhme and M. Kirchner, “Counter-forensics: Attacking image forensics,” in *Digital Image Forensics*, H. T. Sencar and N. Memon, Eds. Springer Berlin / Heidelberg, 2012.

[2] M. Barni and F. Pérez-González, “Coping with the enemy: advances in adversary-aware signal processing,” in *ICASSP 2013, IEEE International Conference on Acoustics, Speech and Signal Processing*, Vancouver, Canada, 26-31 May 2013, pp. 8682–8686.

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

[4] G. Cao, Y. Zhao, and R. Ni, “Forensic estimation of gamma correction in digital images,” in *Image Processing (ICIP), 2010 17th IEEE International Conference on*. IEEE, 2010, pp. 2097–2100.

[5] G. Cao, Y. Zhao, R. Ni, and X. Li, “Contrast enhancement-based forensics in digital images,” *IEEE transactions on information forensics and security*, vol. 9, no. 3, pp. 515–525, 2014.

[6] G. Cao, Y. Zhao, R. Ni, and H. Tian, “Anti-forensics of contrast enhancement in digital images,” in *Proceedings of the 12th ACM Workshop on Multimedia and Security*. ACM, 2010, pp. 25–34.

[7] M. Barni, M. Fontani, and B. Tondi, “A universal technique to hide traces of histogram-based image manipulations,” in *Proceedings of the on Multimedia and security*. ACM, 2012, pp. 97–104.

[8] C. Chen, Y. Q. Shi, and W. Su, “A machine learning based scheme for double JPEG compression detection,” in *19th International Conference on Pattern Recognition, 2008. ICPR 2008.* IEEE, 2008, pp. 1–4.

[9] A. De Rosa, M. Fontani, M. Massai, A. Piva, and M. Barni, “Second-order statistics analysis to cope with contrast enhancement counter-forensics,” *IEEE Signal Processing Letters*, vol. 22, no. 8, pp. 1132–1136, 2015.

[10] X. Pan, X. Zhang, and S. Lyu, “Exposing image forgery with blind noise estimation,” in *Proceedings of the thirteenth ACM multimedia workshop on Multimedia and security*. ACM, 2011, pp. 15–20.

[11] H. Li, W. Luo, X. Qiu, and J. Huang, “Identification of various image operations using residual-based features,” *IEEE Transactions on Circuits and Systems for Video Technology*, 2016.

[12] N. Singh and A. Gupta, “Analysis of contrast enhancement forensics in compressed and uncompressed images,” in *Signal Processing and Communication (ICSC), 2016 International Conference on*. IEEE, 2016, pp. 303–307.

[13] M. Barni, E. Nowroozi, and B. Tondi, “Higher-order, adversary-aware, double jpeg-detection via selected training on attacked samples,” in *2017 25th European Signal Processing Conference (EUSIPCO)*, Aug 2017, pp. 281–285.

[14] P. Bestagini, A. Allam, S. Milani, M. Tagliasacchi, and S. Tubaro, “Video codec identification,” in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*. IEEE, 2012, pp. 2257–2260.

[15] J. Fridrich and J. Kodovsky, “Rich models for steganalysis of digital images,” *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 3, pp. 868–882, 2012.

[16] L. Verdoliva, D. Cozzolino, and G. Poggi, “A feature-based approach for image tampering detection and localization,” in *Information Forensics and Security (WIFS), 2014 IEEE International Workshop on*. IEEE, 2014, pp. 149–154.

[17] M. Goljan, J. Fridrich, and R. Cogranne, “Rich model for steganalysis of color images,” in *Information Forensics and Security (WIFS), 2014 IEEE International Workshop on*. IEEE, 2014, pp. 185–190.

[18] S. M. Pizer, E. P. Amburn, J. D. Austin, R. Cromartie, A. Geselowitz, T. Greer, B. ter Haar Romeny, J. B. Zimmerman, and K. Zuiderveld, “Adaptive histogram equalization and its variations,” *Computer vision, graphics, and image processing*, vol. 39, no. 3, pp. 355–368, 1987.

[19] J. Kodovsky, J. Fridrich, and V. Holub, “Ensemble classifiers for steganalysis of digital media,” *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 2, pp. 432–444, 2012.

[20] T. Pevny, P. Bas, and J. Fridrich, “Steganalysis by subtractive pixel adjacency matrix,” *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 2, pp. 215–224, June 2010.

[21] K. Zuiderveld, “Graphics gems iv,” P. S. Heckbert, Ed. San Diego, CA, USA: Academic Press Professional, Inc., 1994, ch. Contrast Limited Adaptive Histogram Equalization, pp. 474–485. [Online]. Available: http://dl.acm.org/citation.cfm?id=180895.180940

[22] 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.

[23] D.-T. Dang-Nguyen, C. Pasquini, V. Conotter, and G. Boato, “Raise: A raw images dataset for digital image forensics,” in *Proceedings of the 6th ACM Multimedia Systems Conference*, ser. MMSys ’15. New York, NY, USA: ACM, 2015, pp. 219–224. [Online]. Available: http://doi.acm.org/10.1145/2713168.2713194

[24] D. Cozzolino, D. Gragnaniello, and L. Verdoliva, “Image forgery detection through residual-based local descriptors and block-matching,” in *Image Processing (ICIP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 5297–5301.