Full length article

CNN-based first quantization estimation of double compressed JPEG images☆

Sebastiano Battiato [a], Oliver Giudice [b], Francesco Guarnera [a,*], Giovanni Puglisi [c]

[a] Department of Mathematics and Computer Science, University of Catania, Viale Andrea Doria 6, Catania 95125, Italy
[b] Banca d'Italia, Rome, Italy
[c] Department of Mathematics and Computer Science, University of Cagliari, Via Ospedale 72, Cagliari 09124, Italy

ARTICLE INFO

ABSTRACT

Multiple JPEG compressions leave artifacts in digital images: residual traces that could be exploited in forensics investigations to recover information about the device employed for acquisition or image editing software. In this paper, a novel First Quantization Estimation (FQE) algorithm based on convolutional neural networks (CNNs) is proposed. In particular, a solution based on an ensemble of CNNs was developed in conjunction with specific regularization strategies exploiting assumptions about neighboring element values of the quantization matrix to be inferred. Mostly designed to work in the aligned case, the solution was tested in challenging scenarios involving different input patch sizes, quantization matrices (both standard and custom) and datasets (i.e., RAISE and UCID collections). Comparisons with state-of-the-art solutions confirmed the effectiveness of the presented solution demonstrating for the first time to cover the widest combinations of parameters of double JPEG compressions.

1. Introduction

JPEG is the most commonly used file format for digital images. Indeed JPEG is by far the most common compression engine and has been investigated for digital investigation purposes. When facing a JPEG image, the first problem that could be addressed is image history reconstruction [1,2]: this could provide information about image authenticity and the acquisition device that generated it. After the acquisition (and likely a first JPEG compression), the image usually goes through a Social Network or an Instant Messaging platform [3] that in most cases, applies further JPEG compression. To recover information about the acquisition device [4–6] the forensic analysis of JPEG images specializes in several different tasks: (i) the Double Quantization Detection (DQD) to detect if an image has been JPEG compressed at least twice and (ii) the Quantization Step Estimation (QSE) [7,8]. Forensics community has been spending considerable effort on DQD, facing the problem in different contexts [9–12] and employing different strategies [13–15]. In a digital forensics investigation, DQD is often followed by FQE which has the goal to estimate the quantization matrix employed in the first JPEG compression.

Many approaches have recently been proposed to address this problem in various scenarios. Initially, several methods were designed to estimate the first quantization matrix in presence of file format changes

(e.g., from JPEG to Bitmap) when information about the compression matrix is no longer available in embedded metadata (Fan and De Queiroz [2], Li et al. [16]). Later, scenarios involving multiple JPEG compressions (usually two, with some exceptions such as [17]) were investigated. Bianchi et al. [18–20] designed an FQE method based on Expectation Maximization algorithm while Galvan et al. [21] introduced a technique based on the analysis and filtering of Discrete Cosine Transform (DCT) histograms. Similar ideas have also been exploited in [22–24]. Recently, Thai et al. [7,25] exploited intuitive insights usually employed in steganography to design a robust FQE approach.

Due to the large amount of data that can be properly collected and analyzed, machine learning algorithms have also been introduced in recent FQE methods. Lukáš and Fridrich proposed a first approach based on neural network in [26]. Recently, Convolutional Neural Networks (CNNs) have also been exploited to design FQE solutions. In particular, state-of-the-art CNN results have been achieved by Niu et al. [27] and Tondi et al. [28] for both aligned and non-aligned FQE scenarios. These deep learning solutions have been designed to work directly on input patches; although, end-to-end learning can be exploited also in the FQE scenario, this design choice limits the usability of the provided models to the specific patch size used during the training phase. Moreover, in almost all cases the training phase involved only data processed by

---

standard quantization tables producing in output models that suffer heavily of overfitting. To cope with this limit, Battiato et al. [29] proposed a solution based on both statistical analysis and machine learning approach (i.e., nearest neighbor classifier) in the aligned scenario obtaining effective results also in presence of custom quantization matrices also avoiding problem of overfitting.

In this paper we have considerably improved state-of-the-art results by employing a CNN-based solution coupled with a-priori investigation in terms of data exploitation and collection. It is worth noting that the proposed solution, to the best of our knowledge, is the first FQE technique based on deep learning specifically designed to work without any limits in terms of involved quantization matrices employed in the first and second compression steps and able to manage a wide range of input patch sizes without requiring any modification of the related network architecture. Moreover, a novel regularization approach was designed to cope with challenging conditions (i.e., homogeneous patches, specific relations among quantization factors) that are complex to deal with considering information contained in a single DCT histogram. Experiments in different scenarios, involving several datasets confirmed that the proposed strategy significantly outperforms previous studies by a large margin. The main contributions of the proposed solution can be then summarized as follows:

- A novel CNN-based solution designed to work with DCT histograms distributions as input and robust with respect to different challenging conditions in the FQE aligned scenario (i.e., custom quantization tables and patch size).
- A regularization approach, developed measuring real quantization table statistics, able to boost overall performances whenever information available in the single input histogram is partially missing.
- An exhaustive study with many comparisons of state-of-the-art FQE techniques measured on various combination of first and second quantization factors, taking into account quantization parameters of commercial software.

Data used for training and test together with the employed CNN architectural model will be available for research purposes in a public repository.

The remainder of this paper is organized as follows. Section 2 reports the JPEG notation employed in the paper, Section 3 describes the proposed method with preliminary analysis and related parameters setting in Section 4. Experimental results and comparisons are reported in Section 5 whereas Section 6 concludes the paper.

2. JPEG notation

Given a raw image I, JPEG compression [30] can be defined as a function f_Q such that $I' = f_Q(I)$, where I' is the JPEG compressed image, Q is the quantization matrix (8×8) containing the quantization factors $q_i \in \mathbb{N}$ with $i \in \{1, 2, \dots, 64\}$. As first step, $f_Q(I)$ converts I from the RGB to YCbCr color space, and then divides the input image in 8×8 non-overlapping blocks applying also the integer DCT (Discrete Cosine Transform). Finally, each 8×8 block is divided, pixel by pixel, by Q, rounded and then encoded by classic entropy based engine.

In this paper, only luminance (i.e., Y channel) was considered. Let us also define $I'' = f_{Q_2}(f_{Q_1}(I))$ a JPEG double compressed image, with Q_1 and Q_2 denote the quantization matrices employed for the first and the second compression respectively. Moreover, we refer to QF as the standard quantization matrix associated to a specific quality factor [30] while QF_i to further specify the ith JPEG compression (e.g. $i = 1, 2, \dots$) in which the matrix was employed. We denote h_i the empirical distributions built from the ith DCT coefficients extracted from the 8×8 blocks of I''. Finally, we define the k quantization factors, in zig-zag order, of Q_1 as $q1_1, q1_2, \dots, q1_k$, while we denote as $q1$ and $q2$ the quantization factors employed in the first and in the second compression respectively.

3. Proposed method

The proposed method for the estimation of first quantization matrix can be summarized as an ensemble of CNNs specifically designed for the task. Being a machine learning approach, in the following Subsections we will describe in details: (i) the features, (ii) the employed datasets, (iii) the neural network architecture with all the design information needed for reproducibility and finally the developed regularization strategies.

3.1. Features

The main aim of the proposed solution is to exploit the information contained in Discrete Cosine Transform (DCT) distributions computed from a double compressed image to estimate the first k quantization factors employed in the first compression. Preliminary results achieved by Battiato et al. [29], by working on DCT histograms, suggested us to further investigate allowing us to better manage and exploit the considerable amount of involved data. In particular, many state-of-the-art approaches based on machine learning [27,28], usually train their models considering a dataset built with a fixed quantization matrix in the last compression (often a standard matrix related to a specific QF). However, this design choice strongly limits, in some way, the effectiveness of the provided models. In real applications, it is very likely to find double compressed images with Q_2 different than the one used in those studies. This would force the investigator to build a new dataset and to perform a new training phase on it finally obtaining a new model with the desired Q_2 (moreover it should be demonstrated to work effectively).

To better exploit the information contained in the input data (DCT histograms), limiting also the number of models to be trained, for each $q2$ value, only two sets of empirical distributions h_i related to DC and AC coefficients were considered in the proposed approach. All histograms related to the AC terms, due to the similarity of their distributions [31], were then collected together. In our tests, $q1_{max}$ denotes the maximum quantization factor value to be predicted. For example, in the matrix reported in Figs. 1(c) and 1(d) considering $k = 15$ in zig-zag order, $q1_{max}$ values are 5 and 21 respectively (highlighted in the figures).

To sum up, $2 \cdot q1_{max}$ models allowed us to deal with double compressed images with a generic second compression matrix Q_2 whose quantization factors are lower than or equal to $q1_{max}$ in the first k positions (zig-zag order). Let consider that strategies involving all possible k quantization factors in a unified way actually have to take into account $q1_{max}{}^k$ combinations. Considering parameter values usually employed in state-of-the-art solutions, $k = 15$ and $q1_{max} = 22$, in order to cope with the set of second quantization matrices, 22^{15} models have to be trained, whereas employing our strategy we reduced such amount to a finite short number equal to 44. The proposed strategy allows to deal with a large number of double quantization parameters maintaining a just feasible workload, in terms of computational effort.

3.2. Employed dataset

A well known critical aspect in the design of a machine learning algorithm is the choice of a proper dataset to be employed in the training phase. Although several datasets are available in literature, in this paper, RAISE [32] was considered. The choice of RAISE allowed us to obtain heterogeneous images with different resolutions. RAISE is composed of 8156 high-resolution uncompressed images captured in different scenes (indoor, outdoor etc.) employing different cameras.

The double compression phase was carried out extracting different $d \times d$ central patches from raw images and compressing the images with a proper combinations of constant matrices M_i (Figs. 1(a) and 1(b)) with $i \in \{1, 2, \dots, q1_{max}\}$. In our tests, the value of $q1_{max}$ was set to 22 whereas 4 different patch sizes with $d \in \{64, 128, 256, 512\}$

a. M_i with $i = 4$ b. M_i with $i = 22$ c. Standard matrix $QF = 90$ d. Photoshop matrix $QF = 5$
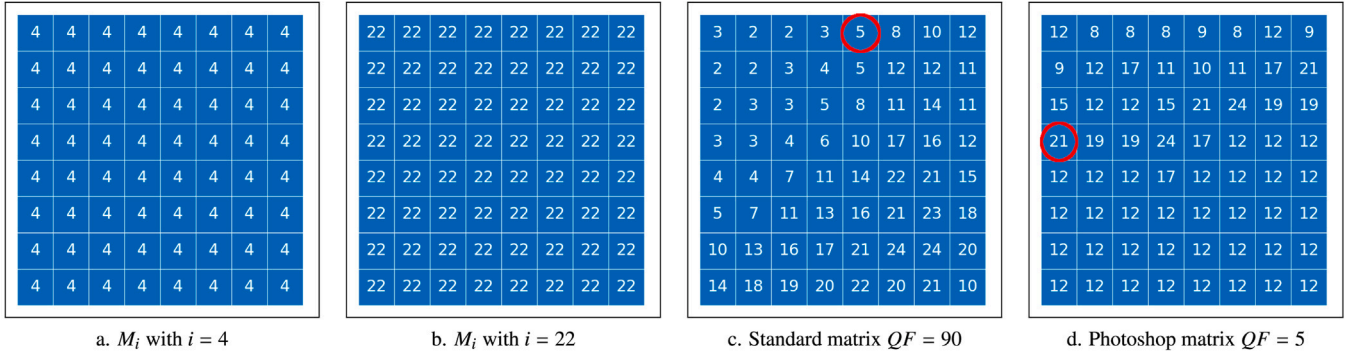
Fig. 1. Example of the constant matrix M_i with $i = 4$ (a) and $i = 22$ (b), standard quantization matrix with $QF = 90$ (c) and custom quantization matrix extracted from Photoshop with quality 5.
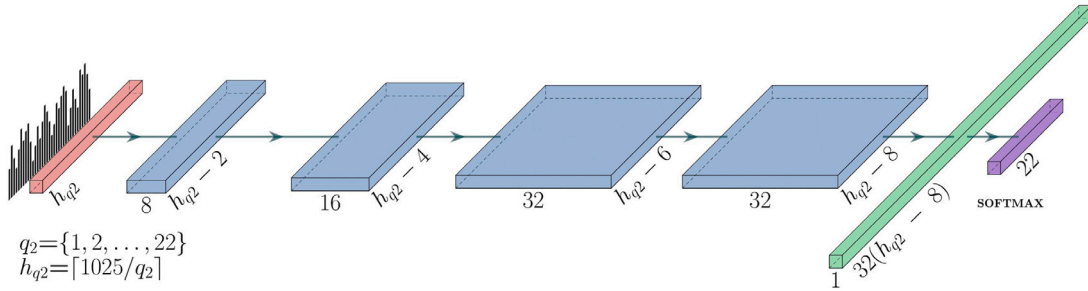

$q2 = \{1, 2, \ldots, 22\}$
$h_{q2} = \lceil 1025/q2 \rceil$

Fig. 2. Parametric architecture representing the trained CNNs. The first layer represents the distribution of ith DCT coefficient, compressed the second time with $q2$. The input distribution is then reduced in h_{q2} bins. The following four layers are 2D convolutions with a filter 1×3, batch normalization and ReLu activation function. The last two layers consist of a fully connected and a softmax layer with 22 elements.

were considered. Hence, the final number of JPEG double compressed images exploited to train the proposed solution was $8156 \times 22 \times 22 \times 4 = 15,790,016$. Moreover, starting from a double compressed image, $k = 15$ DCT distributions (1 DC and 14 AC) were extracted. Each distribution is then characterized by 4 parameters: patch size d, first quantization factor $q1$, second quantization factor $q2$ and specific DCT position k. Finally, to better organize this huge amount of data (i.e., $8156 \times 22 \times 22 \times 4 \times 15$), all the distributions were clustered according to the parameters d, $q2$ and DCT coefficient (DC, AC) generating $4 \times 22 \times 2 = 176$ different sets. It has to be noted that other datasets (BOSSBASE [33] and UCID [34]) were also employed in this paper to avoid overfitting in the parameter setting (Section 4.1) and to test the performance of the proposed solution also considering different configurations in terms of image resolution (Section 5). BOSSBASE [33] is a dataset composed of 1000 512×512 grayscale images, created in 2010 for a scientific challenge with the goal to figure out which images contained a hidden message and which images do not, while UCID [34] is a dataset which has over 1300 medium resolution uncompressed images often employed for forensic purposes.

3.3. Network architecture

The aforementioned choices about employed dataset and histogram generation and organization, are important starting point to the design of a machine learning approach able to work in real scenarios with custom quantization matrices. Deep learning techniques can considerable improve the overall accuracy whilst maintaining robustness and generalization properties.

Given a double JPEG compressed image I'' the main aim of the proposed solution is the estimation of the k first quantization factors employed in the first compression. It is worth noting that Q_2 values can be read directly from the JPEG file: the second quantization factors $q2_i$ with $i \in \{1, 2, \ldots, k\}$ are already available. Furthermore, as already pointed out in Section 3.2, for each $q2$, two different DCT coefficient

types (DC or AC), have to be taken into account, due to their difference in terms of statistical distribution. We then trained one DC-CNN and one AC-CNN for each possible value of $q2 \in \{1, 2, \ldots, q1_{max}\}$. Each CNN has the architecture synthetically sketched in Fig. 2. The input of these networks are the normalized DCT histograms with $h_{q2} = \lceil 1025/q2 \rceil$ bins. The size of the following layers is functions of h_{q2}. Fig. 2 summarizes graphically the neural network architecture layers that consist of 2D convolutions carried out with 1×3 filters, a batch normalization and a ReLu activation whereas the last two layers are fully connected layers with the standard softmax function with $q1_{max} = 22$ values as output layer. As regards the training phase is concerned, we employed Stochastic Gradient Descent (SGD) as optimizer with a starting learning rate 10^{-3} and momentum 9^{-1}, while the categorical cross-entropy was the loss function employed during a 15-epochs training ran with batches of 512 images. Moreover a decay step on learning rate value was carried out, with the drop value described in Eq. (1).

$$lr^e = lr^0 (d_r^{\lfloor (1+e)/ed \rfloor}) \qquad (1)$$

where e is the epoch, lr^e is the learning rate of epoch e, lr^0 is the starting learning rate, $d_r = 0.2$ is the drop value and $ed = 3$ is the number of epochs for every drop.

3.4. Regularization

Sometimes the amount of information contained in an input histogram could be not enough to estimate the related first quantization factor. This lack of information could depend both on the input data (e.g., homogeneous regions) and specific $q1$, $q2$ combinations (e.g., multiples). For example, considering $q2 = 5$, first quantization factor values equal to 1 and 5 are difficult to discriminate. To limit these issues, assumptions about neighboring element values in the quantization matrix can be exploited. To verify these assumptions empirically, an analysis on a dataset (Park et al. [35]) of actual quantization matrices was performed. Specifically, the dataset consists of 1170 different matrices:
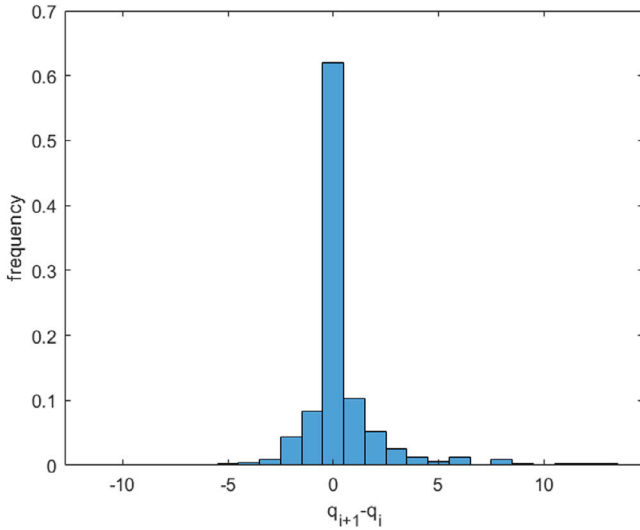
Fig. 3. Empirical distribution of differences of consecutive quantization factors (zig-zag order) built from standard and custom matrices collected in [35] with $q1_i \leq q1_{max} = 22$ and $i \in 1, 2, \ldots, 15$.

1070 custom and 100 standard JPEG quantization tables. Considering only the matrices with $q1_i \leq q1_{max} = 22$ and $i \in 1, 2, \ldots, 15$, 919 tables (both custom and standard) were selected and the empirical distribution of differences between consecutive quantization factors in zig-zag order were built. As shown in Fig. 3, neighboring elements in the quantization matrices (zig-zag order) are usually associated to similar values (i.e., their difference is close to zero).

Considering then a set of n consecutive first quantization factors to be estimated, a cost function C can be designed as the weighted average of a data term (C_{data}) and a regularization term (C_{reg}):

$$C = w \cdot C_{data} + (1 - w) \cdot C_{reg} \qquad (2)$$

where $w \in [0, 1]$, C_{data} is a cost term related to the goodness of the estimation of first quantization factors under analysis, and C_{reg} is a regularization term that tries to minimize differences among neighboring $q1$ values.

An equation similar to (2) has been already proposed in [29] where data used to compute the C_{data} term were obtained employing an algorithm based on nearest neighbors. However, the main limit of the regularization approach proposed in [29] was the strategy adopted to compute the cost function C. That solution actually considered all the combinations of the n consecutive quantization factors. Such as example, if $q1_{max}$ is 22, 22^n combinations, and then evaluations of the cost function C, have to be performed ($n = 3$ in [29]).

To increase the number of consecutive elements to be considered, an analysis of the output of the proposed CNN was performed. Also in presence of challenging conditions (e.g., multiples), the softmax output could provide useful information that can be exploited to limit the set of $q1$ to be taken into account. Specifically, the softmax output is a vector made up of $q1_{max}$ values that can be interpreted as probabilities (they are all positives and sum to one). In Fig. 4 is reported the softmax output computed by the proposed CNN with an histogram obtained from a double compressed 128×128 patch with $q1 = 5$ and $q2 = 5$. Although the estimation provided by the network is wrong (i.e., the first quantization factor reporting the maximum value is $q1 = 1$), the score associated to $q1 = 5$ (i.e., the correct value) is comparable with the best one. It is worth noting that the probability associated to the event $q1 = 1$ is 0.484 whereas the joint probability related to $q1 = 1$ or $q1 = 5$ is 0.959. A set of first quantization factors can be then selected to achieve a satisfactory probability.
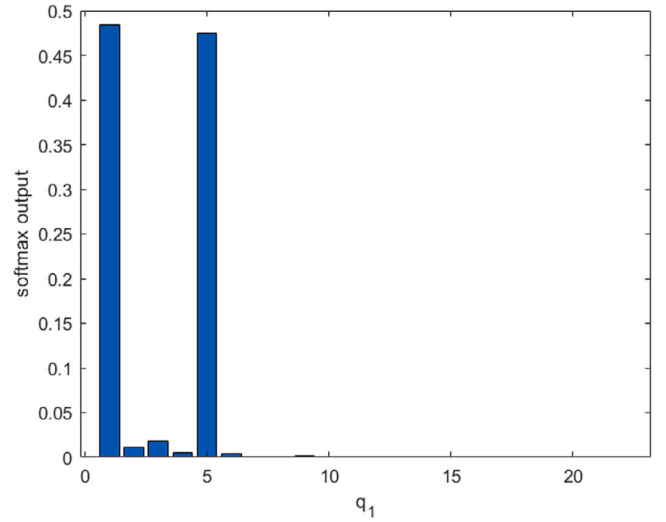

Fig. 4. An example of softmax output provided by the proposed CNN considering an AC histogram obtained with $q1 = 5$ and $q2 = 5$.
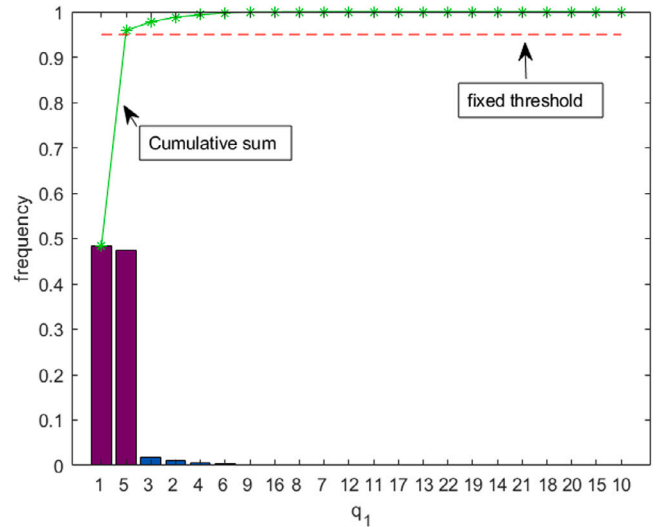

Fig. 5. An example of $q1$ candidate selection from softmax output values provided by the proposed CNN considering the same input of Fig. 4. Softmax outputs p_i are sorted in descending order, the cumulative sum is computed (in green) and compared with a threshold th (in red). The first set of quantization factors whose cumulative sum is higher than th are considered as candidates (1 and 5).

This behavior can be then exploited in different ways depending on the specific scenario. The softmax output, interpreted as a probability, could be used as a simple index of reliability of the estimate performed. Such as example, considering a threshold value of $th = 0.9$, one or multiple elements can be selected in the estimation of the quantization matrix.

Another way to exploit this amount of knowledge is to help reducing the number of first quantization factors to be considered as candidates for the final estimation. Fixed a threshold th in the range $[0, 1]$, and denoted as p_i the output provided by the softmax function with respect to the event $q1 = i$, the smallest set of first quantization factors whose summation of related p_i is higher than th is selected. The quantization factors belonging to this set can be easily collected sorting probabilities p_i in decreasing order and computing the cumulative sum. For instance, considering the softmax outputs depicted in Fig. 4 and $th = 0.95$ only two $q1$ (1 and 5) are selected (see Fig. 5).
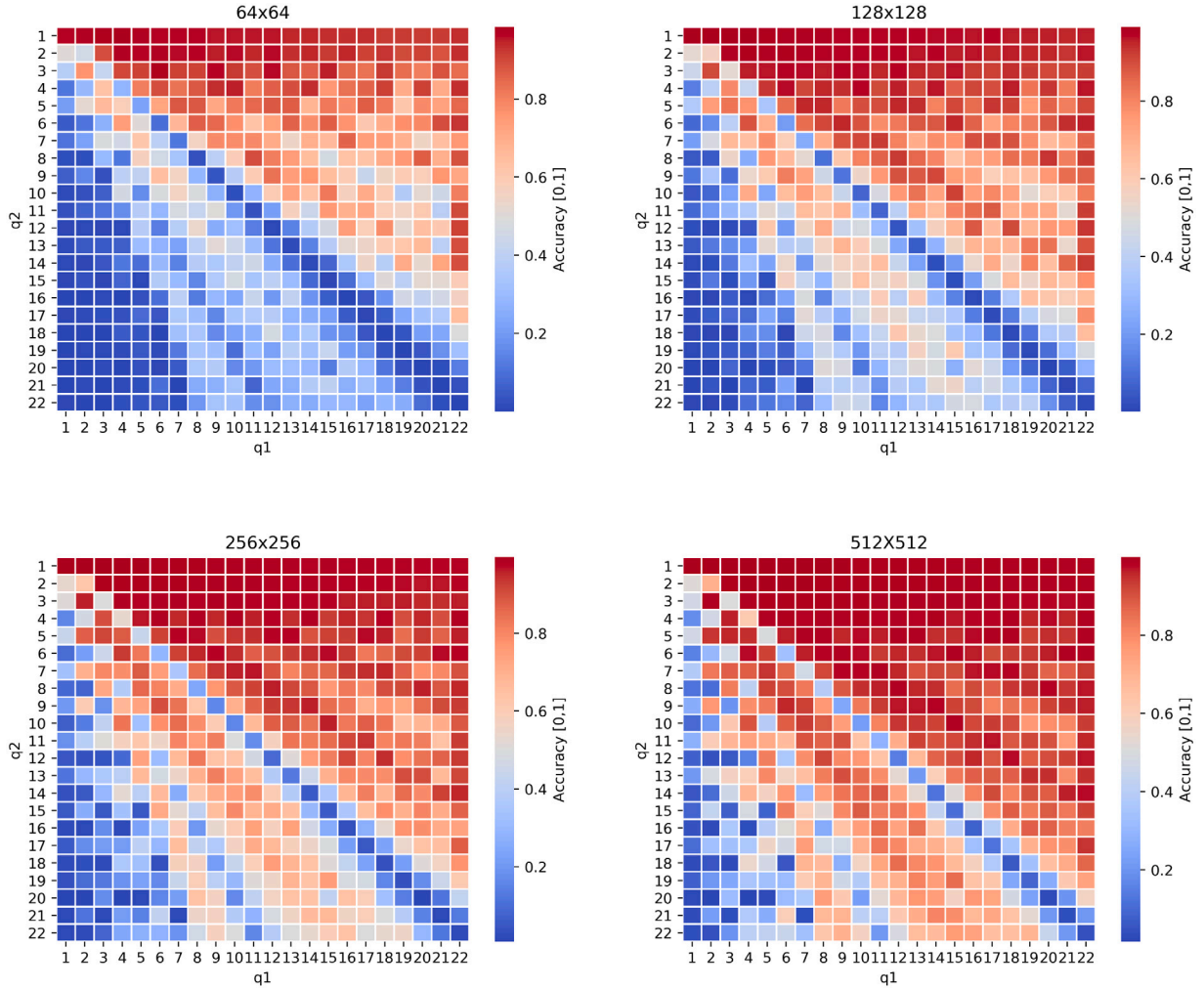
Fig. 6. Accuracies of the trained CNNs at varying of patch size and $q1$, $q2$ combinations. Average values were computed with respect to the first 15 DCT coefficients.

Further details about data and regularization terms and parameter settings (e.g., w and n) are provided in Section 4.1.

The following Algorithm 1 summarizes the estimation of the first k quantization factors employing the trained networks.

Algorithm 1 The Proposed FQE Method

Input: double compressed image I''
Output: $\{q1_1, q1_2, \ldots, q1_k\}$
    Initialization : k
1: **for** $i = 1$ to k **do**
2:     h_i : (empirical) distribution of i-th DCT coefficient
3:     $q2_i$: quantization factor of Q_2 for i-th DCT
4:     **if** ($i = 1$) **then**
5:         net : $DCq2_i$ trained network
6:     **else**
7:         net : $ACq2_i$ trained network
8:     **end if**
9:     $q1_i$: predicted by net
10: **end for**
11: regularize($\{q1_1, q1_2, \ldots, q1_k\}$)
12: **return** $\{q1_1, q1_2, \ldots, q1_k\}$

---

4. Overall analysis

As already pointed out in Section 3, the proposed solution differently than previous works was specifically designed to work with a

wide set of Q_2 matrices. For each $q2$, two CNNs related to DC and AC terms were trained by employing the parametric architecture depicted in Fig. 2. Although the designed CNN, taking as input DCT histograms, is not strictly limited to be used with a specific patch size, the accuracy with respect to different input parameters was also evaluated. More specifically, for each patch size $d \in \{64, 128, 256, 512\}$, $2 \cdot 22$ CNNs were trained with empirical histograms from double compressed JPEG images (see Section 3.2). Each dataset, one per patch size, was split into 80% training, 10% validation and 10% test and exploited to train 4 sets of $2 \cdot 22$ CNNs. Each group of CNNs is then trained with histograms obtained from input patches of the same size. It has to be noted that histograms containing no information [36] have been removed and were not considered in our tests. Results of the proposed CNNs at varying of patch size and $q1$, $q2$ combinations are reported in Fig. 6 where average values are computed with respect to the first 15 DCT coefficients. The obtained accuracies strictly depend on the amount of information contained in the input histogram (higher at increasing of patch size) and on the combination of $q1$ and $q2$ values (e.g., multiples). It is worth noting that the reported results were computed considering both training and test set related to input patches of the same size. To further study the performance of the proposed solution, additional tests were performed considering a scenario with a mismatch between train and test set patch size. Table 1 shows the average accuracy, computed with respect to $q1 \in \{1, 2, \ldots, 22\}$, $q2 \in \{1, 2, \ldots, 22\}$ and the first 15 DCT terms, achieved by each couple dataset/CNN. As expected, it is evident that, for each dataset, the best result corresponds to the CNN trained with images of the same size. Note that scenarios involving both

Table 1
Accuracies of the CNNs trained with a specific patch size with respect to all the four generated test datasets ($d \in \{64, 128, 256, 512\}$). Average values were computed with respect to $q1 \in \{1, 2, \dots, 22\}$, $q2 \in \{1, 2, \dots, 22\}$ and the first 15 DCT terms. Each test set is a subset (10%) of the related one described in Section 3.2 built employing constant matrices for first and second compressions and images from RAISE [32] collection.

Test set	CNN trained with			
	64×64	128×128	256×256	512×512
64×64	**0.44**	0.43	0.41	0.38
128×128	0.53	**0.54**	0.53	0.50
256×256	0.57	0.62	**0.63**	0.62
512×512	0.58	0.65	0.70	**0.71**

Table 2
Accuracies of the CNN ensembles with respect to all the four generated test datasets ($d \in \{64, 128, 256, 512\}$). Average values were computed with respect to $q1 \in \{1, 2, \dots, 22\}$, $q2 \in \{1, 2, \dots, 22\}$ and the first 15 DCT terms.

Test set	CNN		
	$ensemble_2$	$ensemble_4$	$ensemble_8$
64×64	**0.4447**	0.4443	0.4434
128×128	**0.5425**	0.5420	0.5411
256×256	**0.6302**	0.6295	0.6282
512×512	**0.7070**	0.7059	0.7046

standard and custom quantization matrices actually select a subset of the possible $q1$, $q2$ combinations. Reported average accuracies are then not the same of the ones shown in Section 5.

As already pointed out in Section 3.1, the proposed solution, taking as input DCT histograms, does not strictly depend on a specific patch size. In order to improve the overall effectiveness and usability of the method, a single set of networks can be then trained with multiple patch sizes avoiding then the selection of a specific group of CNNs for each patch size. This improvement can be also really useful whenever input patch size actually differs from the one employed to train the models (e.g., 96×96, 384×512, etc.).

A novel dataset was then built by simply merging the collections employed before with $d \in \{64, 128, 256, 512\}$. However, the data to be handled requires a large amount of memory resources. In order to train the proposed method as the patch size varies, while exploiting all available data, a solution based on ensemble of CNNs was considered. Specifically, the merged dataset was split into 10 subdatasets, 8 employed for training, 1 for validation and 1 for test. Three CNN ensembles were considered with the following strategy: $ensemble_m$ represents an ensemble of m CNNs trained with $8/m$ training subdatasets $m \in \{2, 4, 8\}$. As reported in Table 2 all the proposed ensembles, differently than CNNs trained with fixed patch size, achieve satisfactory accuracy in all the considered test sets $d \in \{64, 128, 256, 512\}$. Although all the considered models achieve comparable accuracy ($ensemble_2$ slightly better than remaining ones), the solution with $m = 2$ does not increase considerably the execution time with respect to the networks trained with a fixed patch size.

4.1. Regularization (parameter settings)

To better justify the design choices related to the regularization approach described in Section 3.4 several tests were performed. Specifically, four double compressed datasets obtained cropping central patches with $d \in \{64, 128, 256, 512\}$, from 1000 images selected from BOSSBASE collection [33] were built. This collection was considered in the parameter setting to limit the overfitting with respect to the dataset employed to train the CNNs (i.e., RAISE [32]). To cope with real scenarios, double compression is performed employing custom tables from [35] by considering only the matrices with $q1_i \leq q1_{max} = 22$ and $i \in 1, 2, \dots, 15$. Two different C_{data} terms were considered:

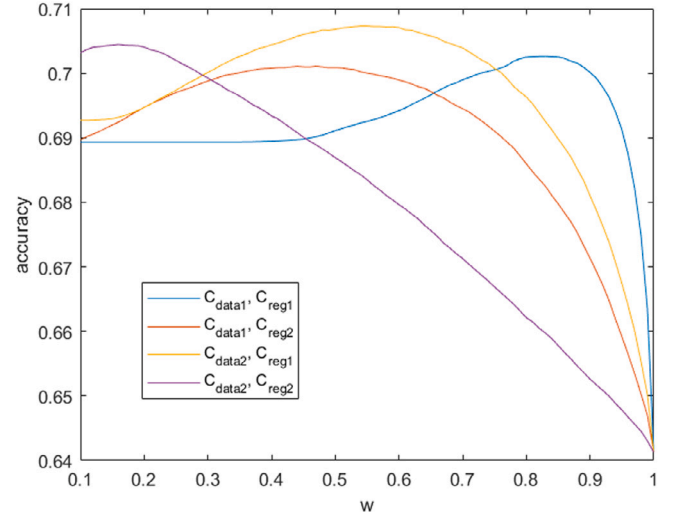$$C_{data1} = 1 - \frac{1}{n} \sum_{j=i-\lfloor n/2 \rfloor}^{i+\lfloor n/2 \rfloor} p_j \tag{3}$$

Fig. 7. Average accuracy of the proposed regularization solution considering $n = 3$ neighbors and all the combinations of C_{data} and C_{reg} formulas.

$$C_{data2} = -\frac{1}{n} \sum_{j=i-\lfloor n/2 \rfloor}^{i+\lfloor n/2 \rfloor} \log p_j \tag{4}$$

where i indicates the position (zig-zag order) of the DCT term under analysis, n the number of considered neighbors and p_j the probability (i.e., softmax output) provided by the proposed CNN at position j related to $q1_j$.

Moreover, two different C_{reg} terms have been investigated:

$$C_{reg1} = \frac{1}{n-1} \sum_{j=i-\lfloor n/2 \rfloor+1}^{i+\lfloor n/2 \rfloor} |q1_j - q1_{j-1}| \tag{5}$$

$$C_{reg2} = \frac{2}{n-1} \sum_{j=i-\lfloor n/2 \rfloor+1}^{i+\lfloor n/2 \rfloor} \frac{|q1_j - q1_{j-1}|}{q1_j + q1_{j-1}} \tag{6}$$

where $q1_i$ is the first quantization candidate at position i (zig-zag order) under analysis.

In Fig. 7 are reported the average accuracies obtained employing (2) considering all the possible combinations of C_{data} ((3), (4)) and C_{reg} ((5), (6)) with $n = 3$. For each weighting factor w, the average accuracy is computed taking into account the four aforementioned datasets ($d \in 64, 128, 256, 512$) and the DCT coefficients. Best performances are obtained with C_{data2} and C_{reg1}. An additional test was carried out with C_{data2} and C_{reg1} at varying of the number of neighbors $n \in \{3, 5, 7\}$ and weighting factor w. Moreover, to make results comparable with respect to different number of neighbors n, only positions $i = 4, \dots, 12$ are considered in the parameter setting tests. Note that, although the regularization strategy described in Section 3.4 considerably reduces the average number of combinations, worst case scenario has to be avoided. To this aim, the maximum number of allowed combinations per estimation was set to 10^6. As can be easily seen from Fig. 8, $n = 7$ and $w = 0.43$ provide the best results.

5. Experimental results

In order to demonstrate the effectiveness of the proposed method, a series of comparisons with the state-of-the-art solutions were carried out. The device employed to run the experiments was an hardware equipped with a GPU NVIDIA TESLA K80. Both statistical [19,21,22,37] and machine learning approaches [27–29] were selected for comparison. The original code provided by the authors of the aforementioned state-of-the-art solutions was employed. Moreover, to cope with real
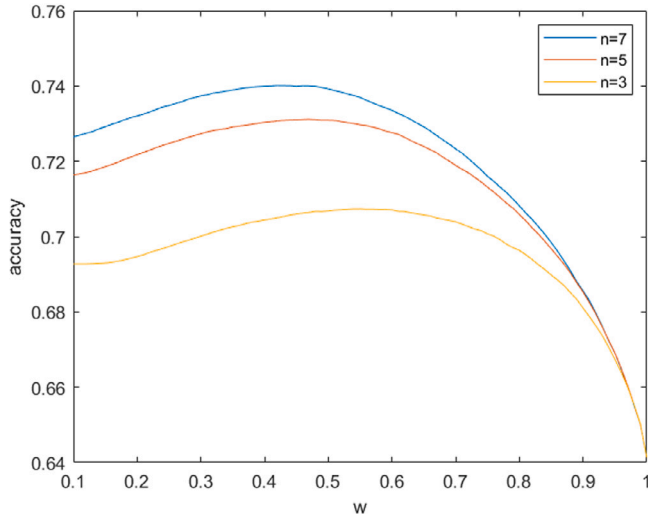
Fig. 8. Average accuracy computed considering C_{data2} and C_{reg1} (best combinations) at varying of n (i.e., number of neighbors). Note that $w = 1$ actually corresponds to results achieved without employing any regularization.

scenarios, both standard and custom matrices have also been considered in our tests. Finally, as already described in state-of-the-art, high frequencies are usually killed by the JPEG compression and after a certain position the values are zero (the so-called 'dead-zone'). An acceptable trade-off employed by the community is the usage of $k = 15$. Tests described in the following sections, for sake of comparison, were then performed with $k = 15$.

5.1. Comparison test

Some recent state-of-the-art solutions have been designed to work with specific patch sizes [27,28]. To properly compare the proposed method with the aforementioned approaches, a first series of tests were then performed considering several scenarios involving 64×64 patches as input. Moreover, the effectiveness of the proposed solution has been demonstrated considering both statistical [19,21,22,37] and machine learning methods [27–29]. Specifically, four double compressed datasets were generated starting from random 64×64 patches cropped from RAISE collection [32] (one patch for each RAISE image):

1. $QF_1 \in \{55, 60, 65, 70, 75, 80, 85, 90, 95, 98\}$, $QF_2 = 90$
2. $QF_1 \in \{55, 60, 65, 70, 75, 80, 85, 90, 95, 98\}$, $QF_2 = 80$
3. $Q_1 \in \{5, 6, 7, 8, 9, 10, 11, 12\}$, $QF_2 = 90$
4. $Q_1 \in \{5, 6, 7, 8, 9, 10, 11, 12\}$, $QF_2 = 80$

where $Q_1 \in \{5, 6, 7, 8, 9, 10, 11, 12\}$ of (3) and (4) are referred to Photoshop's quantization matrices (version 20.0.4).

Datasets (1) and (2) are actually related to a classical scenario involving only standard quantization matrices whereas, datasets (3) and (4), employing also Photoshop's quantization tables, can be considered a more challenging test to verify the robustness of the considered methods with respect to real conditions (i.e., custom quantization matrices). Note that Dalmia et al. [22] has not been taken into account in the comparisons involving dataset (3) and (4) due to the assumptions about the standard matrices of first compression in the provided implementation.

As reported in Table 3, where a classical scenario involving standard quantization matrices is considered, the proposed approach outperforms state-of-the-art solutions in almost all combinations. These results are also confirmed in Figs. 9(a) and 9(b) with performances analyzed at varying of DCT coefficients. Moreover, the robustness of the proposed solution at varying of the employed quantization matrices has been tested with custom tables. As reported in Table 4 and Figs. 9(c) and

Table 3

Accuracies obtained by the proposed approach (with and without regularization) compared to Bianchi et al. [19], Galvan et al. [21], Dalmia et al. [22], Niu et al. [27], Tondi et al. [28], Battiato et al. [37] and Battiato et al. [29] with different combinations of QF_1/QF_2, considering standard quantization tables.

QF_1	$QF_2 = 90$								
	Our	Our reg.	[19]	[21]	[22]	[27]	[28]	[29]	[37]
55	**0.84**	0.80	0.53	0.52	0.45	0.00	0.00	0.77	0.62
60	0.81	**0.86**	0.53	0.56	0.47	0.64	0.53	0.82	0.66
65	**0.84**	**0.84**	0.54	0.57	0.49	0.54	0.81	0.81	0.68
70	0.83	**0.88**	0.43	0.57	0.51	0.66	0.66	0.85	0.70
75	0.89	0.89	0.41	0.63	0.53	0.77	**0.93**	0.85	0.75
80	**0.89**	0.87	0.29	0.61	0.45	0.81	0.67	0.83	0.75
85	0.86	**0.90**	0.14	0.74	0.36	0.81	0.88	0.85	0.72
90	**0.40**	0.12	0.00	0.00	0.00	0.02	0.02	0.24	0.23
95	0.50	0.62	0.11	0.00	0.00	**0.78**	0.68	0.52	0.40
98	0.36	0.76	0.00	0.00	0.00	0.76	**0.91**	0.57	0.41
MEAN	0.72	**0.75**	0.30	0.42	0.33	0.58	0.61	0.71	0.59
QF_1	$QF_2 = 80$								
55	**0.76**	0.61	0.36	0.37	0.37	0.24	0.52	0.58	0.49
60	**0.66**	0.65	0.27	0.37	0.38	0.50	0.31	0.60	0.47
65	0.67	**0.71**	0.19	0.41	0.43	0.31	0.38	0.65	0.52
70	0.72	**0.82**	0.19	0.50	0.49	0.50	0.70	0.75	0.66
75	**0.67**	0.65	0.07	0.56	0.45	0.15	0.58	0.56	0.47
80	**0.13**	0.05	0.00	0.00	0.00	0.00	0.04	0.11	0.04
85	**0.56**	0.53	0.19	0.00	0.00	0.04	0.14	0.34	0.31
90	0.34	0.40	0.06	0.00	0.00	**0.48**	0.37	0.19	0.21
95	0.11	0.35	0.00	0.00	0.00	**0.95**	0.37	0.30	0.19
98	0.09	0.43	0.01	0.00	0.00	0.21	**0.48**	0.42	0.19
MEAN	0.47	**0.52**	0.13	0.22	0.21	0.28	0.39	0.45	0.36

Table 4

Accuracies obtained by the proposed approach (with and without regularization) compared to Bianchi et al. [19], Galvan et al. [21], Niu et al. [27], Tondi et al. [28], Battiato et al. [37] and Battiato et al. [29] employing custom tables for first compression. The column PS refers to the quality (and then custom tables) used by Photoshop.

PS	$QF_2 = 90$							
	Our	Our reg.	[19]	[21]	[27]	[28]	[29]	[37]
5	**0.86**	0.84	0.56	0.58	0.05	0.10	0.78	0.69
6	**0.91**	0.88	0.46	0.60	0.07	0.13	0.82	0.75
7	**0.86**	0.85	0.41	0.58	0.07	0.09	0.83	0.70
8	**0.86**	**0.86**	0.25	0.65	0.10	0.13	0.81	0.74
9	0.67	**0.76**	0.02	0.47	0.02	0.05	0.61	0.47
10	0.60	**0.72**	0.19	0.00	0.25	0.50	0.50	0.43
11	0.40	0.69	0.04	0.00	0.69	**0.75**	0.52	0.40
12	0.36	0.76	0.04	0.00	0.75	**0.85**	0.57	0.41
MEAN	0.69	**0.80**	0.25	0.36	0.25	0.30	0.68	0.58
PS	$QF_2 = 80$							
5	**0.72**	**0.72**	0.26	0.46	0.07	0.08	0.68	0.57
6	0.53	**0.68**	0.05	0.41	0.02	0.06	0.54	0.42
7	0.72	**0.74**	0.15	0.48	0.08	0.10	0.68	0.56
8	0.33	**0.35**	0.03	0.03	0.01	0.04	0.22	0.18
9	0.49	**0.54**	0.19	0.00	0.07	0.13	0.28	0.28
10	0.23	**0.47**	0.00	0.00	0.40	0.16	0.20	0.17
11	0.09	**0.39**	0.01	0.00	0.24	0.38	0.38	0.23
12	0.09	0.43	0.01	0.00	0.21	**0.44**	0.42	0.26
MEAN	0.40	**0.54**	0.09	0.18	0.17	0.14	0.42	0.33

9(d), the gain in terms of achieved accuracy of the proposed approach with respect to the other CNN-based methods [27,28] increases considerably. Differently than the proposed approach, end-to-end CNN solutions employed in [27,28] suffer in presence of quantization matrices that have not been considered in their training process.

5.2. Generalizing test

Most of state-of-the-art methods have been designed and tested considering standard quantization tables. However, as reported in [35]
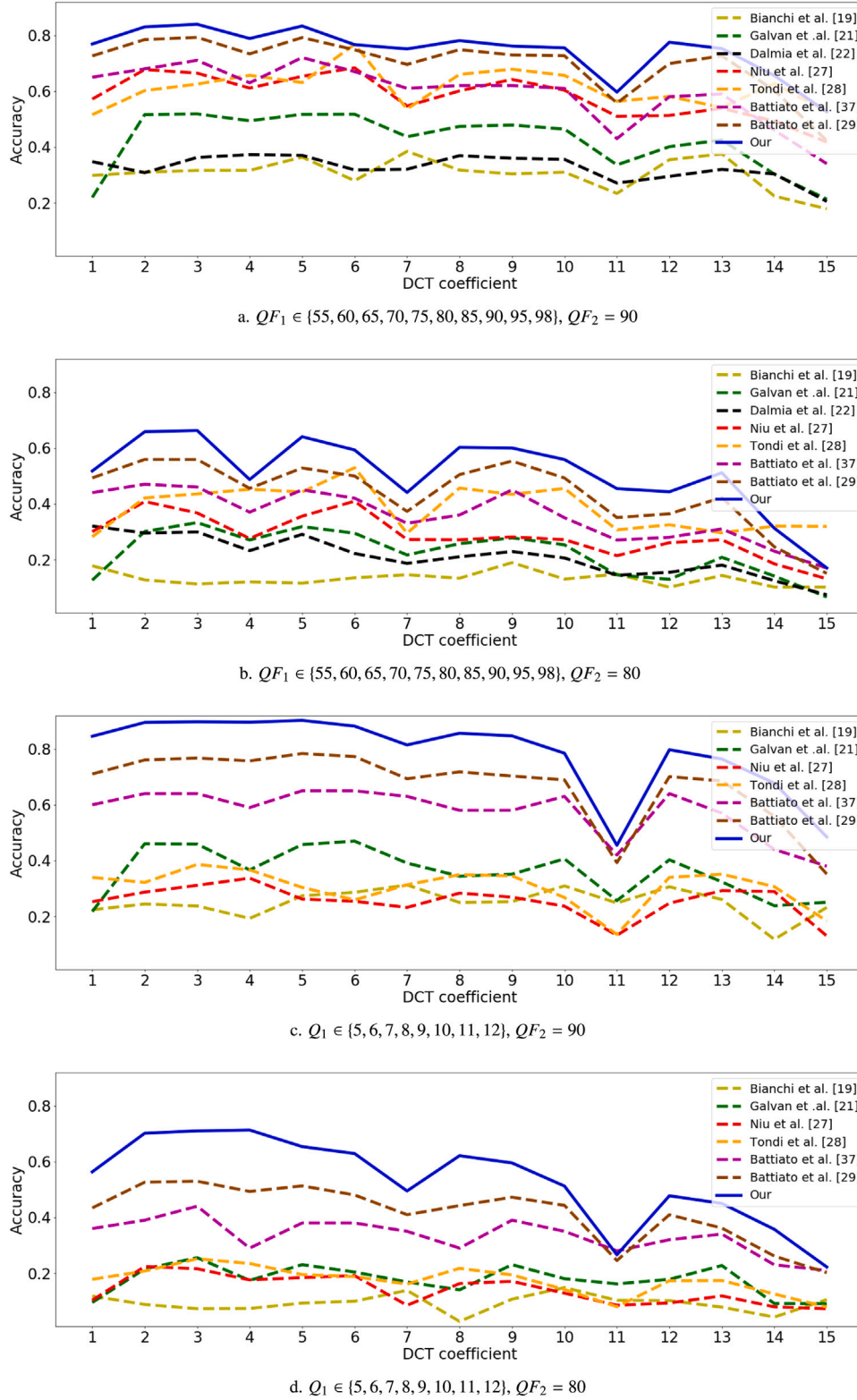
a. $QF_1 \in \{55, 60, 65, 70, 75, 80, 85, 90, 95, 98\}$, $QF_2 = 90$

b. $QF_1 \in \{55, 60, 65, 70, 75, 80, 85, 90, 95, 98\}$, $QF_2 = 80$

c. $Q_1 \in \{5, 6, 7, 8, 9, 10, 11, 12\}$, $QF_2 = 90$

d. $Q_1 \in \{5, 6, 7, 8, 9, 10, 11, 12\}$, $QF_2 = 80$

Fig. 9. Accuracies of the same methods described in Tables 3 and 4 at varying of the quantization factors $q1_i$ to be predicted. The values are averaged over all the QF_1/Q_1.

analyzing JPEG images downloaded from Internet, custom quantization tables are often employed in the compression pipeline. To further demonstrate the robustness of the proposed solution with respect to the employed quantization matrices an additional test was performed. Specifically, Park et al. in [35] collected 1170 quantization matrices

(100 standard and 1070 custom) retrieved by real scenarios. This collection was exploited to carry out further tests: the matrices with $q1_i \leq q1_{max} = 22$ and $i \in \{1, 2, \ldots, 15\}$ were selected, sorted by the average of the first 15 quantization factors and then split into three sets (called Low, Mid, High) of 291 elements. Due to the double compression, 9
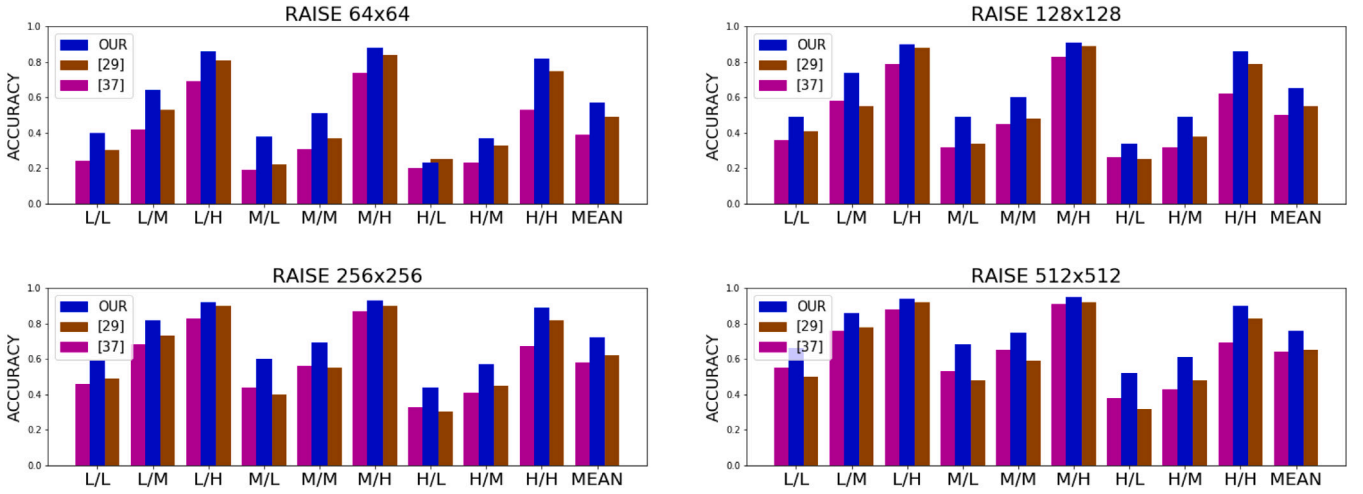
Fig. 10. Comparison between the proposed solution, Battiato et al. [37] and Battiato et al. [29] considering custom tables from [35] and patches from RAISE dataset at varying of $d \in \{64, 128, 256, 512\}$. L, M, and H represent respectively the sets of matrices Low, Mid and High described in Section 5.2.

Fig. 11. Comparison between the proposed solution, Battiato et al. [37] and Battiato et al. [29] considering custom tables from [35] and patches from UCID dataset at varying of $d \in \{64, 128, 256, 512\}$. L, M, and H represent respectively the sets of matrices Low, Mid and High described in Section 5.2.
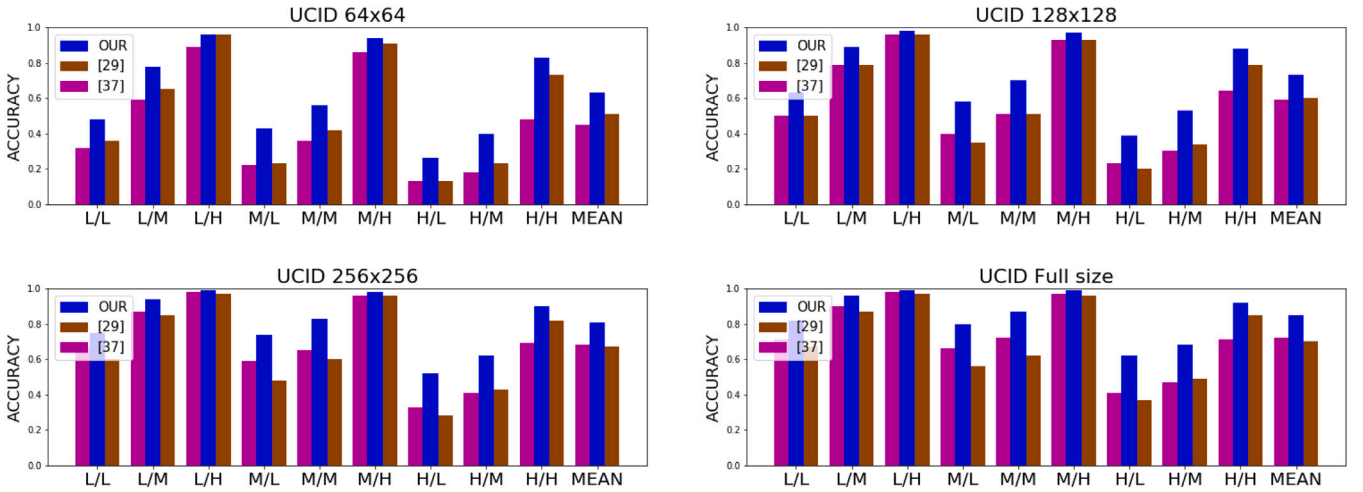
different combinations (every ordered couple among Low, Mid and High) have been then considered. Moreover, to study the performance in the wild conditions 8 different input datasets were created: 4 different patch sizes (64×64, 128×128, 256×256, 512×512) cropped from RAISE [32] and UCID [34]. For each dataset, the quantization tables employed for double compression were randomly selected from the 291 available in the corresponding set (Low, Mid, High). It is worth to note that UCID dataset, due to the different resolution of the original images used to extract patches vs. the collection employed to train the CNNs (i.e., RAISE [32]), allow us to verify the robustness of the proposed solution with respect to the variability of the dataset.

As can be seen from Figs. 10 and 11 the proposed approach achieves satisfactory accuracy even in this challenging scenario. In addition, the results are closely related to the amount of information contained in the input histogram. An higher accuracy is therefore obtained as the patch size increases and with UCID dataset [34].

5.3. Tampering localization

FQE can be simply employed to perform tampering localization. The classical scenario is the following one: a copy-paste of JPEG image (foreground image) is applied on a JPEG image (background image) in order to add or hide some information. To localize the tampered areas a sliding window approach can be applied, getting a map for each DCT coefficient; every window estimation represents a pixel and values related to tampered areas must be different than the other ones. We conducted a test with the following parameters. Starting from a JPEG image compressed with $QF = 60$ (the background image, Fig. 12 in the top) 2 tampering were applied: the first one is a copy-move of a patch extracted from the same background image, compressed with $QF = 90$ and moved in the top-left corner of the image while the second one is an external image JPEG compressed with $QF = 80$ and applied in the bottom-right corner. The image was then compressed again with $QF = 90$ obtaining the tampered image depicted in Fig. 13 (bottom). To localize tampered regions, FQE estimation was done with our method for every patch 64×64 of a sliding window moved 8 pixel each time (in both the directions). For each DCT coefficient, starting from the values provided by the FQE algorithm, a mask can be generated. As can be seen from Fig. 14, tampered regions can be easily detected from the generated masks. Future steps of this work will include the analysis of the FQE in presence of different types of manipulations [38] but also in presence of artifacts introduced by the normal life cycle of an image (e.g., the manipulation introduced by the upload on Social platforms [39]).

Fig. 12. Original image compressed with $QF = 60$.

Fig. 13. Tampered image employed in the test. Several patches have been added to the original image (Fig. 12) and a second compression with $QF = 90$ is performed.

6. Conclusion

The estimation of the first quantization matrix is useful to recover information about the history of the image under analysis mainly for forensics purposes. In this paper, a novel CNN-based estimation solution has been proposed in the aligned scenario. By proper collecting and training a neural architecture by considering 1D-histograms of DCT values (AC and DC terms) the proposed method outperforms state-of-the-art solutions by a large margin. Moreover, experiments results carried out in challenging scenarios, confirmed the robustness of the designed solution with respect to input patch size, quantization matrix (both standard and custom) and employed datasets (RAISE and UCID). A regularization strategy devoted to improve overall results in challenging conditions (i.e., quite homogeneous patches, specific
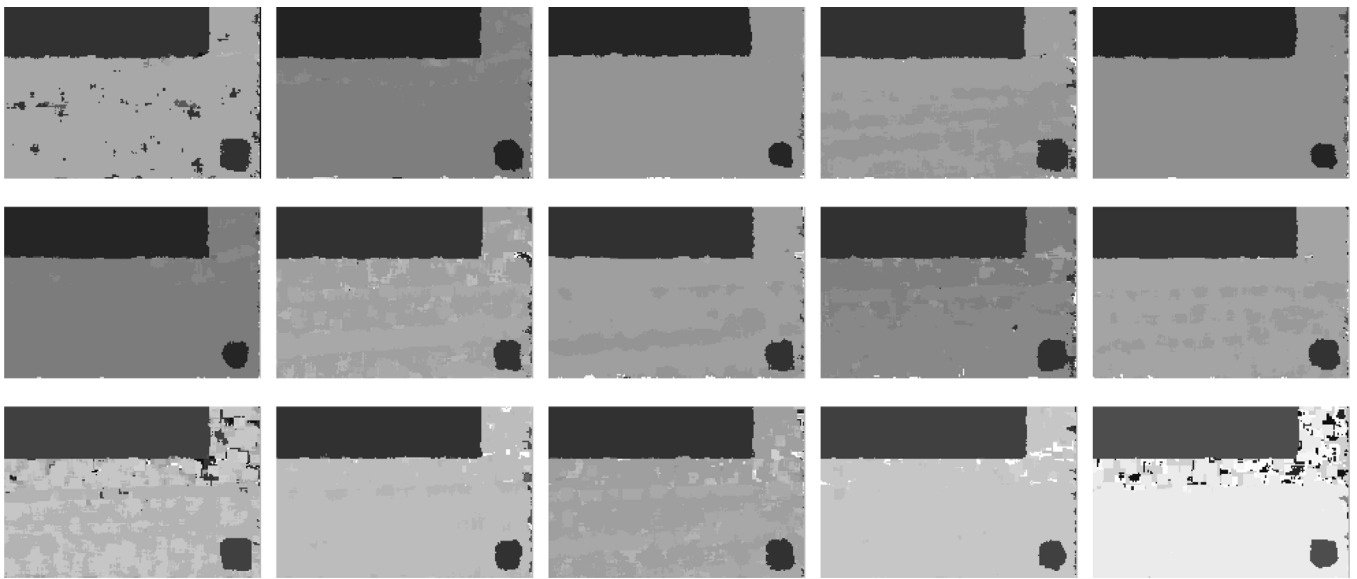
Fig. 14. Masks obtained for each coefficient, from 1 (top-left), to 15 (bottom-right) in zig-zag order.

relations among quantization factors) that are complex to deal with considering only information contained in a single DCT histogram has been also designed. Future works will be devoted to further extend the proposed solution to the non-aligned case and other forensics tasks such as tampering detection/localization.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

References

[1] H. Farid, Digital Image Ballistics from JPEG Quantization: A Followup Study, Tech. Rep. TR2008-638, Department of Computer Science, Dartmouth College, 2008.

[2] Z. Fan, R. De Queiroz, Identification of bitmap compression history: JPEG detection and quantizer estimation, IEEE Trans. Image Process. 12 (2) (2003) 230–235.

[3] O. Giudice, A. Paratore, M. Moltisanti, S. Battiato, A classification engine for image ballistics of social data, in: Proc. of International Conference on Image Analysis and Processing, LNCS, Vol. 10485, Springer, 2017, pp. 625–636.

[4] A. Piva, An overview on image forensics, ISRN Signal Process. 2013 (2013) 22.

[5] L. Verdoliva, Media forensics and deepfakes: An overview, IEEE J. Sel. Top. Sign. Proces. 14 (5) (2020) 910–932.

[6] M.C. Stamm, M. Wu, K.J.R. Liu, Information forensics: An overview of the first decade, IEEE Access 1 (2013) 167–200.

[7] T.H. Thai, R. Cogranne, F. Retraint, Statistical model of quantized DCT coefficients: Application in the steganalysis of Jsteg algorithm, IEEE Trans. Image Process. 23 (5) (2014) 1980–1993.

[8] J. Yang, Y. Zhang, G. Zhu, S. Kwong, A clustering-based framework for improving the performance of JPEG quantization step estimation, IEEE Trans. Circuits Syst. Video Technol. (2020).

[9] G. Varghese, A. Kumar, Detection of double JPEG compression on color image using neural network classifier, Int. J. Innov. Res. Sci. Technol. 3 (3) (2016) 175–181.

[10] M. Barni, L. Bondi, N. Bonettini, P. Bestagini, A. Costanzo, M. Maggini, B. Tondi, S. Tubaro, Aligned and non-aligned double JPEG detection using convolutional neural networks, J. Vis. Commun. Image Represent. 49 (Supplement C) (2017) 153–163.

[11] T. Uricchio, L. Ballan, R. Caldelli, I. Amerini, Localization of JPEG double compression through multi-domain convolutional neural networks, in: Proc. of the IEEE Conference on Computer Vision and Pattern Recognition Workshops, 2017, pp. 53–59.

[12] Y. Niu, X. Li, Y. Zhao, R. Ni, Detection of double JPEG compression with the same quantization matrix via convergence analysis, IEEE Trans. Circuits Syst. Video Technol. (2021).

[13] O. Giudice, F. Guarnera, A. Paratore, S. Battiato, 1-D DCT domain analysis for JPEG double compression detection, in: Proc. of International Conference on Image Analysis and Processing, LNCS, Vol. 11752, Springer, 2019, pp. 716–726.

[14] S. Agarwal, H. Farid, Photo forensics from JPEG dimples, in: Proc. of IEEE Workshop on Information Forensics and Security, WIFS, IEEE, 2017, pp. 1–6.

[15] I. Hussain, S. Tan, B. Li, X. Qin, D. Hussain, J. Huang, A novel deep learning framework for double JPEG compression detection of small size blocks, J. Vis. Commun. Image Represent. 80 (2021) 103269.

[16] B. Li, T. Ng, X. Li, S. Tan, J. Huang, Statistical model of JPEG noises and its application in quantization step estimation, IEEE Trans. Image Process. 24 (5) (2015) 1471–1484.

[17] S. Battiato, O. Giudice, F. Guarnera, G. Puglisi, Estimating previous quantization factors on multiple JPEG compressed images, EURASIP J. Inf. Secur. (8) (2021) 1–11.

[18] T. Bianchi, A. De Rosa, A. Piva, Improved DCT coefficient analysis for forgery localization in JPEG images, in: Proc. of IEEE International Conference on Acoustics Speech and Signal Processing, ICASSP, 2011, 2444, 2447.

[19] T. Bianchi, A. Piva, Image forgery localization via block-grained analysis of JPEG artifacts, IEEE Trans. Inf. Forensics Secur. 7 (3) (2012) 1003.

[20] A. Piva, T. Bianchi, Detection of non-aligned double JPEG compression with estimation of primary compression parameters, in: Proc. of 18th IEEE International Conference on Image Processing, ICIP, 2011, pp. 1929–1932.

[21] F. Galvan, G. Puglisi, A.R. Bruna, S. Battiato, First quantization matrix estimation from double compressed JPEG images, IEEE Trans. Inf. Forensics Secur. 9 (8) (2014) 1299–1310.

[22] N. Dalmia, M. Okade, First quantization matrix estimation for double compressed JPEG images utilizing novel DCT histogram selection strategy, in: Proc. of the Tenth Indian Conference on Computer Vision, Graphics and Image Processing, 2016, pp. 1–8.

[23] H. Yao, H. Wei, T. Qiao, C. Qin, JPEG quantization step estimation with coefficient histogram and spectrum analyses, J. Vis. Commun. Image Represent. (2020) 102795.

[24] N. Dalmia, M. Okade, Robust first quantization matrix estimation based on filtering of recompression artifacts for non-aligned double compressed JPEG images, Signal Process., Image Commun. 61 (2018) 9–20.

[25] T.H. Thai, R. Cogranne, Estimation of primary quantization steps in double-compressed JPEG images using a statistical model of discrete cosine transform, IEEE Access 7 (2019) 76203–76216.

[26] J. Lukáš, J. Fridrich, Estimation of primary quantization matrix in double compressed JPEG images, in: Proc. of the Digital Forensic Research Workshop, 2003, pp. 5–8.

[27] Y. Niu, B. Tondi, Y. Zhao, M. Barni, Primary quantization matrix estimation of double compressed JPEG images via CNN, IEEE Signal Process. Lett. 27 (2020) 191–195.

[28] B. Tondi, A. Costanzo, D. Huang, B. Li, Boosting CNN-based primary quantization matrix estimation of double JPEG images via a classification-like architecture, EURASIP J. Inf. Secur. 2021 (1) (2021) 1–14.

[29] S. Battiato, O. Giudice, F. Guarnera, G. Puglisi, First quantization estimation by a robust data exploitation strategy of DCT coefficients, IEEE Access 9 (2021) 73110–73120.

[30] G.K. Wallace, The JPEG still picture compression standard, Commun. ACM 34 (4) (1991) 30–44.

[31] E.Y. Lam, J.W. Goodman, A mathematical analysis of the DCT coefficient distributions for images, IEEE Trans. Image Process. 9 (10) (2000) 1661–1666.

[32] D. Dang-Nguyen, C. Pasquini, V. Conotter, G. Boato, RAISE: A raw images dataset for digital image forensics, in: Proc. of the 6th ACM Multimedia Systems Conference, 2015, pp. 219–224.

[33] P. Bas, T. Filler, T. Pevnỳ, Break our steganographic system: The ins and outs of organizing BOSS, in: Proc. of International Workshop on Information Hiding, Springer, 2011, pp. 59–70.

[34] G. Schaefer, M. Stich, UCID: An uncompressed color image database, in: Storage and Retrieval Methods and Applications for Multimedia 2004, vol. 5307, International Society for Optics and Photonics, 2003, pp. 472–480.

[35] J. Park, D. Cho, W. Ahn, H.-K. Lee, Double JPEG detection in mixed JPEG quality factors using deep convolutional neural network, in: Proceedings of the European Conference on Computer Vision, ECCV, 2018, pp. 636–652.

[36] S. Battiato, O. Giudice, F. Guarnera, G. Puglisi, In-depth DCT coefficient distribution analysis for first quantization estimation, in: Proc. of MultiMedia FORensics in the WILD (MMFWild2020) - Workshop of 25th International Conference on Pattern Recognition, Vol. 12666, ICPR, Springer, 2021, pp. 573–587.

[37] S. Battiato, O. Giudice, F. Guarnera, G. Puglisi, Computational data analysis for first quantization estimation on JPEG double compressed images, in: 25th International Conference on Pattern Recognition, ICPR, IEEE, 2021, pp. 5951–5958.

[38] H. Farid, Image forgery detection, IEEE Signal Process. Mag. 26 (2) (2009) 16–25.

[39] M. Moltisanti, A. Paratore, S. Battiato, L. Saravo, Image manipulation on facebook for forensics evidence, in: Proc. of International Conference on Image Analysis and Processing, Springer, 2015, pp. 506–517.