MULTIPLE JPEG COMPRESSION DETECTION THROUGH TASK-DRIVEN NON-NEGATIVE MATRIX FACTORIZATION

Sara Mandelli, Nicolò Bonettini, Paolo Bestagini, Vincenzo Lipari, Stefano Tubaro

Dipartimento di Elettronica, Informazione e Bioingegneria, Politecnico di Milano, Italy

ABSTRACT

Due to the increasingly unbridled practice of sharing visual content on the web, tracing back past history of uploaded images is getting far from being an easy task. Nonetheless, forensic analysts might be interested in probing digital history of content published on the web to assess its authenticity. In this vein, a possible indicator of image integrity is the number of JPEG compressions a picture underwent. As a matter of fact, JPEG compression is typically operated first at image inception time directly on the acquisition device. Then, it is customary re-applied every time an image is manipulated or shared through social media. For this reason, the more the applied JPEG compressions, the more the likelihood that an image underwent some editing. In this work, we propose an algorithm to detect multiple JPEG compressions, specifically up to four coding cycles. This approach leverages the Task-driven Non-negative Matrix Factorization (TNMF) model, fed with histograms of the Discrete Cosine Transform (DCT) of the image under analysis. Experimental results show the effectiveness of the method if compared with the state-of-the-art, confirming this strategy as a viable solution for detecting multiple JPEG compressions.

Index Terms— Image forensics, multiple JPEG compression, data-driven, non-negative matrix factorization

1. INTRODUCTION

Acquisition, editing and diffusion of images over the Internet are nowadays widespread operations. Images available online are likely to be the result of a multi-processing chain, engendering concerns about their authenticity and integrity [1]. Therefore, multimedia forensics community has been focusing on detecting potential processing traces on images with the goal of restore faith in digital pictures [2, 3]. In particular, due to the wide diffusion of JPEG image coding scheme, there is wide literature devoted to investigate and exploit traces left by JPEG compression both in forensic [4–6] and counter-forensic [7–9] scenarios. Indeed, most of the images available online are compressed according to JPEG standard, which leaves on each picture peculiar traces that can be exploited for forensic investigations.

Many of the forensic techniques proposed so far limit their goal to the detection of double compression, i.e., they aim at discovering if an image has been compressed once or twice [10–12]. Indeed, double compressed images are more likely been processed than single compressed ones. In particular, several proposed approaches are based on the investigation of quantized Discrete Cosine Transform (DCT) coefficient statistics, which are characteristically shaped by JPEG compression procedure. Some works embrace the analysis of histograms of DCT coefficients [13–15], or DCT First Significant Digit [16].

Recently, research on image forensics has started also to analyze chains of processing operations in order to model more realistic scenarios [17]. Nevertheless, the issue of identifying multiple JPEG-compressions is, by far, a less investigated problem. However, in many practical situations, pictures under analysis might be compressed several times. Think for example to the increasingly widespread habit of sharing visual content on social media: the average user typically shot a picture with a smartphone (first compression), share it through a messaging app (second compression), and the receiver may re-share it or post it on a social platform (third compression). As a consequence, this operation necessarily results in an "at-least-three-compression" chain for the image under analysis.

Given the strong likelihood of digital images to undergo more than two compression stages, finding a method able to estimate the number of endured JPEG compressions is of paramount importance for the reconstruction of processing history of the investigated content. In this vein, the method proposed in [18] aims at identifying up to three JPEG compressions through a testing scheme based on the statistical analysis of Benford-Fourier coefficients [19]. The problem of detecting up to four JPEG compressions is addressed in [20] by exploiting the First Significant Digit of DCT coefficient in absolute values. The algorithm is based on Support Vector Machines (SVMs) and allows to estimate up to four compression cycles.

In this paper, we propose a novel method for detecting up to four JPEG compressions. In particular, we cast multiple JPEG compression as four-class supervised classification problem. To solve it, we exploit Task-driven Dictionary Learning (TDL) model described in [21]. The goal of TDL is to learn a feature dimensionality reduction strategy based on sparse data representation, which minimizes classification loss by jointly optimizing the dictionary used for dimensionality reduction and the classifier. More specifically, we propose to feed histograms of DCT coefficients as features to TDL model. Given the non-negativity of these features (i.e., histograms bin counts cannot be negative), this paper illustrates a more specific formulation of TDL, namely Task-driven Non-negative Matrix Factorization (TNMF) [22], which has proven to be even more effective than deep learning paradigms situations characterized by non-negative features [23].

The paper is structured as follows. Section 2 introduces the problem formulation and some background. Section 3 reports a detailed algorithm description. Section 4 shows the achieved results compared with state-of-the-art multiple JPEG compression techniques. Finally, Section 5 concludes the paper.

This material is based on research sponsored by DARPA and Air Force Research Laboratory (AFRL) under agreement number FA8750-16-2-0173. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of DARPA and Air Force Research Laboratory (AFRL) or the U.S. Government.

2. PROBLEM STATEMENT

During standard JPEG compression, input images are partitioned into 8×8 non-overlapped pixel blocks. Discrete Cosine Transform (DCT) is computed for each one of them, and transform coefficients are quantized into integer-valued levels depending on the selected quantization matrix and quality factor (QF). Quantized values are then converted into a binary stream, exploiting lossless coding. In decoding phase, binary stream is decompressed, coded blocks are reconstructed by applying inverse DCT on rescaled coefficients and the image is re-built in the pixel domain [24].

Due to quantization, it is well-known in the literature that histograms of DCT coefficients show a typical comb-like shape, and spacing between consecutive peaks is related to the adopted quantization step size. Moreover, if an image is encoded many times with different quality factors, the resulting quantization levels are modified accordingly [25].

In this paper we propose a method for detecting multiple-JPEG compressions. This means, given an image, detect how many times (up to four) it has been JPEG compressed. To do so, we leverage perturbations of DCT histograms that capture traces of multiple compressions, and train a supervised classifier to discriminate between images compressed different amount of times.

More in details, multiple-JPEG detection is performed using Task-driven Non-negative Matrix Factorization (TNMF) algorithm [21, 22, 26]. This method allows to reduce feature dimensionality, which helps avoiding data redundancy, by jointly estimating a dictionary for reduced data representation and a multinomial classifier for multiple-JPEG detection. The rationale behind this choice is that, by optimizing feature dimensionality reduction method, we should be able to obtain better performance than methods that exploit first significant digits as DCT histogram reduction methods [20].

3. PROPOSED METHOD

The proposed pipeline for multiple-JPEG compression classification is depicted in Fig. 1. During training: (i) a feature vector is extracted from training images; (ii) TNMF algorithm is used in order to jointly learn a dictionary for feature reduction and estimate parameters of a supervised classifier. When the system is trained, a new image can be tested. To this purpose: (i) a feature vector is extracted from the image; (ii) features are projected into a reduced dimensionality space using the learned dictionary; (iii) classifier is used to detect the number of compressions. In the following, we provide a detailed analysis of each step of the algorithm and a simplified example of how TNMF dimensionality reduction works.

Feature extraction. In order to start our analysis, we first extract a set of selected features from each image. We opted for the feature extraction pipeline presented in [15], which exploits block-wise DCT histograms of the image. Multiple compression stages are well known to strongly condition the histograms of DCT coefficients, hence justifying our approach. In particular, the set of investigated coefficients includes only the first 9 AC spatial frequencies taken in zig-zag order. Our choice comes from the more regular trend of lower frequencies coefficients and from the reduced statistics of higher components, which are often quantized to zero [20]. For what concerns the histograms, we select only the first 21 central bins for each DCT band, ending up with a feature vector $\mathbf{x} \in \mathbb{R}_+^n$ of n elements per image, with $n = 189$. Notice that $\mathbf{x}$ assumes non-negative values only.

TNMF Training. Since we aim at classifying multiple compressed images, we propose to exploit multinomial logistic regression, in a one-vs-rest implementation. This means that the multi-class classifier is actually composed by four binary classifiers (e.g., one compression vs. others, two compressions vs. others, etc.), and results of these are merged. For each class label, we train the classifier by minimizing the logistic loss function, defined as $l_s = l_s(y_i, \mathbf{A})$, where y_i is the image label and $\mathbf{A} = \{\mathbf{w}, \mathbf{c}\}$ is the parameter configuration related to that class [27].

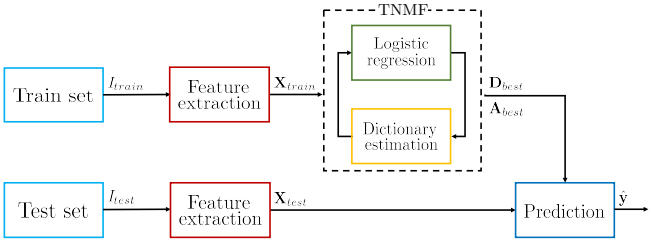


Fig. 1: Schematic representation of proposed pipeline.

In particular, we propose to exploit Task-driven Non-negative Matrix Factorization (TNMF), which is capable of finding sparse data representations by learning a dictionary suited to the specific task of classification [21]. TNMF model allows to learn the task-driven dictionary and the classifier parameters in a joint iterative fashion. More specifically, we estimate a classifier which is optimized with respect to standard logistic regressor, thanks to a particular representation of input data: DCT features extracted from images are projected on a dictionary that is actually tailored to our multinomial classification task. The algorithm works iteratively, alternating the updating of classifier and dictionary, until a fixed number of iterations is achieved. It follows an exhaustive illustration of the method.

a) Feature reduction. At each iteration t, TNMF starts with feature reduction. First of all, we select as dictionary the one from the previous iteration, hence $\mathbf{D}_t = \mathbf{D}_{t-1}$, with $\mathbf{D}_t \in \mathbb{R}_+^{n \times p}$. Given a training vector $\mathbf{x}_i$ generated from image I_i, TNMF model considers the optimal projections of data points on the dictionary, with the constraints that all the elements of $\mathbf{x}_i$, $\mathbf{D}_t$ and the obtained projections are non-negative. Notice that p corresponds to the desired size of reduced features, n is the input feature size, thus $p < n$.
Typically, the problem is formulated as follows:

$$\mathbf{h}_{i,t}(\mathbf{D}_t) = \arg\min_{\mathbf{h} \in \mathbb{R}_+^p} \|\mathbf{x}_i - \mathbf{D}_t\,\mathbf{h}\|_2^2 + \lambda_1 ||\mathbf{h}||_1 + \lambda_2 ||\mathbf{h}||_2^2 \,. \quad (1)$$

$\mathbf{h}_{i,t}(\mathbf{D}_t) \in \mathbb{R}_+^p$ is the estimated projection, λ_1 and λ_2 are regularization penalty terms, in order to impose sparsity (ℓ_1 norm) and to obtain a strongly convex problem (ℓ_2 norm) hence guaranteeing a unique solution. Eq. (1) can be solved with standard techniques available in the literature as shall be cleared in the experimental results section.

b) Classifier updating. Once we have defined the optimal projections, we can use them to update the classifier, associating each vector $\mathbf{h}_{i,t}$ to its related class label y_i. This operation is performed by the minimization of the expected value of loss function l_s over the entire training set:

$$\mathbf{w}_t,\, \mathbf{c}_t = \arg\min_{\mathbf{w},\mathbf{c}} \mathbb{E}_{y_i,\mathbf{x}_i}[l_s(y_i, \mathbf{w}, \mathbf{c}, \mathbf{h}_{i,t}(\mathbf{D}_t))] + \nu ||\mathbf{w}||_2^2 \,, \quad (2)$$

where ν is the penalty term of the regularizer, introduced to prevent overfitting in the classifier. In the proposed framework, minimization is solved by means of the L-BFGS iterative algorithm [28]. In other words, this step consists in training the multi-class logistic regressor exploiting projected training data samples $\mathbf{h}_{i,t}$ and their labels.

c) Dictionary updating. At the end of each iteration t, the dictionary must be updated considering the trained logistic regressor at step (b), thus obtaining $\mathbf{D}_t$ to be used during next iteration $t+1$. This is done through the minimization of loss function (2) with respect to the dictionary $\mathbf{D}_t$. In particular, we update the dictionary by means of stochastic gradient descent, evaluating the function in each training sample $\mathbf{x}_i$ and minimizing in an iterative manner. In order to perform this task, we have to re-evaluate the sparse representation of each training data sample, $\mathbf{h}_{i,t}(\mathbf{D}_t)$, which depends on the dictionary $\mathbf{D}_t$ estimated from already analyzed samples ($\mathbf{x}_j$, $j < i$). To be more specific, the minimization is performed in two sequential steps:

(i) We exploit stochastic gradient descent by calculating the gradient with respect to $\mathbf{h}_{i,t}(\mathbf{D}_t)$. Since we work with sparse representations of data, we compute the active set $\mathcal{S}$ by selecting only the indexes $\in \{1, ..., p\}$ for which vector $\mathbf{h}_{i,t}(\mathbf{D}_t) \neq 0$. For the sake of notation, we introduce the variable $\mathbf{g}_{i,t}$, defined as:

$$\mathbf{g}_{i,t} = ([\mathbf{D}_t^\top]_\mathcal{S}[\mathbf{D}_t]_\mathcal{S} + \lambda_2 \mathbf{I}_{|\mathcal{S}|})^{-1}[\nabla_{\mathbf{h}_{i,t}(\mathbf{D}_t)} l_s(y, \mathbf{A}_t, \mathbf{h}_{i,t}(\mathbf{D}_t))]_\mathcal{S}$$

where symbol $[\cdot]_\mathcal{S}$ represents the projection on the active set, and $|\mathcal{S}|$ is the cardinality of $\mathcal{S}$.

(ii) It follows a projection of $\mathbf{g}_{i,t}$ over the dictionary space, leading to this updating formulation:

$$\mathbf{D}_t = \mathbf{D}_t \; - \; \rho_t(-\mathbf{D}_t\, \mathbf{g}_{i,t}\, \mathbf{h}_{i,t}^\top(\mathbf{D}_t) \; + \; (\mathbf{x}_i - \mathbf{D}_t\, \mathbf{h}_{i,t}(\mathbf{D}_t))\, \mathbf{g}_{i,t})\,.$$

In order to impose the non-negativity of each dictionary element, we select $\epsilon = 10^{-7}$ as floor value in case of negative entries of $\mathbf{D}_t$. The learning rate ρ_t is chosen with the same heuristic criterion proposed in [21]: we select it as $\min(\rho, \rho \cdot n_{iter}/(10t))$, being ρ a parameter to set and n_{iter} the number of iterations. Given the conspicuous theoretical baggage of TNMF, we skip all the formal derivations, addressing the interested reader to [21] for a thorough explanation.

Following the typical framework of dictionary learning problems, we adopt a validation strategy for selecting the best dictionary and classifier. More specifically, we split our data in training and validation set, evaluating the classification accuracy on validation set at each iteration t, and electing as best dictionary the matrix $\mathbf{D}_t$ which returns the best accuracy. We report below the pseudo-code of TNMF method.

TNMF Testing. Once we estimate the best combination of dictionary and classifier, we are ready for test phase. Given any new image I_{test}, we compute DCT histograms to obtain feature vector $\mathbf{x}_{test}$. By considering the best validation dictionary, $\mathbf{D}_{best}$, we apply (1) to project $\mathbf{x}_{test}$ and obtain the reduced feature vector $\mathbf{h}_{test}$. Finally, we feed $\mathbf{h}_{test}$ to the logistic regressor using the best validation configuration $\mathbf{A}_{best}$ in order to perform label prediction, as in a typical classification problem.

TNMF Training ($\mathbf{y}, \mathbf{X}, p, n_{iter}, \lambda_1, \lambda_2, \nu$) $\rightarrow \mathbf{D}_{best}, \mathbf{A}_{best}$

Initialize dictionary $\mathbf{D}_0$
Split training and validation sets:
$\mathbf{y}_{train}$, $\mathbf{y}_{val}$, $\mathbf{X}_{train}$, $\mathbf{X}_{val} \leftarrow \mathbf{y}$, $\mathbf{X}$
for $t = 1, \ldots, n_{iter}$
  Initialize dictionary: $\mathbf{D}_t = \mathbf{D}_{t-1}$
  Feature reduction: $\mathbf{h}_t(\mathbf{D}_t) \leftarrow \mathbf{D}_t$, $\mathbf{X}_{train}$
  Classifier updating: $\mathbf{A}_t \leftarrow \mathbf{h}_t(\mathbf{D}_t)$, $\mathbf{y}_{train}$
  Validation accuracy: acc $\leftarrow \mathbf{D}_t$, $\mathbf{A}_t$, $\mathbf{y}_{val}$, $\mathbf{X}_{val}$
  $\mathbf{D}_{best}$, $\mathbf{A}_{best} \leftarrow \text{acc}_{\max}\{\mathbf{D}_t, \mathbf{A}_t\}$
  for $i = 1, \ldots, N$
    Single feature extraction: $\mathbf{h}_{i,t}(\mathbf{D}_t) \leftarrow \mathbf{D}_t$, $\mathbf{x}_{train_i}$
    Active set: $\mathcal{S} \leftarrow$ indexes $\in \{1, ..., p\} : \mathbf{h}_{i,t}(\mathbf{D}_t) \neq 0$
    Update learning rate: $\rho_t \leftarrow \min(\rho, \rho\frac{t_0}{t})$
    Update dictionary:
    $\mathbf{D}_t = \mathbf{D}_t - \rho_t(-\mathbf{D}_t\, \mathbf{g}_{i,t}\, \mathbf{h}_{i,t}^\top(\mathbf{D}_t) + ...$
      $+ (\mathbf{x}_i - \mathbf{D}_t\, \mathbf{h}_{i,t}(\mathbf{D}_t))\, \mathbf{g}_{i,t})$
    Impose non-negativity: $\mathbf{D}_t(\mathbf{D}_t < 0) = \epsilon$
  end
end

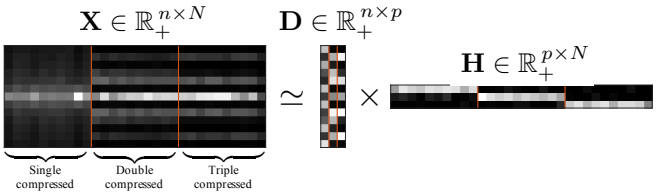


Fig. 2: Simple TNMF example: $\mathbf{X}$ is the matrix of data, specifically N is the total amount of training samples. We can exploit TNMF to approximate $\mathbf{X}$ as the product of the dictionary $\mathbf{D}$ and the matrix $\mathbf{H}$ containing in its columns the optimal projections of $\mathbf{X}$ on $\mathbf{D}$.

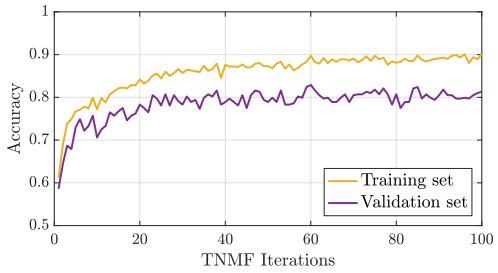
TNMF Training: a simplified example. For the sake of simplicity and data visualization, let us consider a simplified problem consisting of a small dataset of images compressed up to three times. From each image we extract feature $\mathbf{x}$ only considering the 7-th DCT frequency, picking the first central 13 bins. Selecting as reduced feature size $p = 3$, we leverage TNMF training algorithm to find a dictionary for representing our data. In particular, as we are working in a simplified scenario aiming at a ternary classification (discriminating up to three JPEG compressions), a good feature reduction method should enable ternary classification in the reduced space. Fig. 2 depicts the results of dictionary learning through TNMF: we are actually able to estimate a dictionary, associating reduced features to original input data (i.e., classification result is clear just by looking at projected features). Notice that matrix $\mathbf{X}$ illustrates quite well the effects of multiple quantization steps: the more the compression stages, the lower the density of the histogram bins.

4. EXPERIMENTAL RESULTS

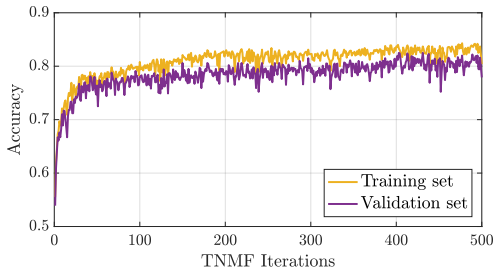
Datasets generation. Following the procedure depicted in [20], we build three datasets starting from 1338 images from UCID [29] (384×512 pixels). Given a final quality factor $\text{QF}_f \in \{75, 80, 90\}$, we compress each grayscale image up to 4 times. The intermediate QF at compression step $i < f$ is randomly chosen in the interval $\big[\text{QF}_{i+1} - 12,\ \text{QF}_{i+1} - 5\big] \cup \big[\text{QF}_{i+1} + 5,\ \text{QF}_{i+1} + 12\big]$ to ensure that QF_i differs from QF_{i+1}. We refer to these datasets as $\mathcal{D}_{75}^U$, $\mathcal{D}_{80}^U$, $\mathcal{D}_{90}^U$, each of them with $4 \times 1338 = 5352$ images. In order to test our approach on a larger scale, we build three further datasets starting with 4000 grayscale images from RAISE database [30]. Due to the large dimensions of these images, we previously center-crop them to 512×512 pixels and then apply the aforementioned compression pipeline. We obtain $\mathcal{D}_{75}^R$, $\mathcal{D}_{80}^R$, $\mathcal{D}_{90}^R$, each of them with $4 \times 4000 = 16000$ images.

TNMF parameters. We follow a common train-validation-test approach, using 70% of each dataset images for training and the remaining for testing. Training set is further divided in training and validation, following a 90%-10% partition. In particular, as recommended in [21], we initialize the dictionary $\mathbf{D}_0$ by the unsupervised formulation of the problem, leveraging the SPAMS toolbox for computations [31]. The size of $\mathbf{D}_0$ has been chosen as trade off between result quality and computational cost: we set p as the 30% of the DCT length, hence drawing a dictionary $\in \mathbb{R}_+^{189 \times 57}$.

Moreover, in order to improve the convergence speed of the



(a) UCID



(b) RAISE

Fig. 3: Classification accuracy of TNMF algorithm. (a) UCID Dataset. (b) RAISE Dataset.

training phase, the proposed method works with a minibatch strategy for the stochastic gradient descent. This basically takes into consideration $n_{batch} > 1$ training samples at each iteration of the inner loop, instead of a single one [31].

Due to the large amount of parameters of TNMF algorithm, we select some specific values for tuning (i.e., iterations $\in \{50, 100, 200, 500\}$, batch size $\in \{200, 400\}$, $\rho \in \{0.001, 0.005, 0.01\}$, $\lambda_1 \in \{0.01, 0.1, 0.5\}$) and perform a grid search in order to obtain the best accuracy on the validation set. In particular, being TNMF an iterative algorithm, number of iterations has a severe impact on validation accuracy, thus we explore different values until convergence.

For what concerns the remaining parameters, we set $\lambda_2 = 0$ drawing the idea from [21], even though $\lambda_2 > 0$ would be necessary for the differentiability of (1). This has proven to get satisfactory results in most experiments. The penalty weight in (2) is left untouched with respect to the standard formulation of logistic regressor, hence $\nu = 1$.

Comparison with state-of-the-art. At first glance, we notice that the algorithm needs more iterations on RAISE-derived datasets than on the UCID ones. This is probably due to the huge difference in terms of dataset size. In this vein, Fig. 3 depicts the temporal evolution of TNMF accuracy, evaluated for training and validation sets on $\mathcal{D}^U_{75}$ and $\mathcal{D}^R_{75}$. Notice that, if on $\mathcal{D}^U_{75}$ we achieve convergence in at most 100 iterations, $\mathcal{D}^R_{75}$ requires more than 200 iterations. For the sake of brevity, we are showing results for the specific combination of parameters which yields the best validation accuracy, and we will stick to this approach from now on. Nonetheless, we performed a comprehensive investigation on test results for all the parameter configurations, obtaining puny variations among them (standard deviation of test accuracy < 0.01).

Tables 1 and 2 show results of the test phase, in terms of mean accuracies and confusion matrices. Specifically, we compare our strategy to [20] (i.e., the only algorithm that deals with four JPEG compressions to the best of our knowledge) and to standard logistic regression without the feature reduction step. This last experiment is used to study the actual positive effect of dictionary projection.

Table 1: Mean test accuracies over 4 classes for proposed TNMF method, classifier in [20], and logistic regressor (LR) without feature reduction. Best results in bold.

UCID	TNMF	[20]	LR	**RAISE**	TNMF	[20]	LR
$\mathcal{D}^U_{75}$	**0.78**	0.73	0.64	$\mathcal{D}^R_{75}$	**0.80**	0.76	0.64
$\mathcal{D}^U_{80}$	**0.82**	0.75	0.65	$\mathcal{D}^R_{80}$	**0.81**	0.75	0.64
$\mathcal{D}^U_{90}$	**0.87**	0.80	0.75	$\mathcal{D}^R_{90}$	**0.87**	0.83	0.74

Table 2: Confusion matrices for $\mathcal{D}^R_{80}, \mathcal{D}^R_{90}$. Top: proposed method. Bottom: method in [20]. The highest accuracy among the two methods for a given compression step and dataset is highlighted in yellow.

$\mathcal{D}^R_{80}$	1	2	3	4	$\mathcal{D}^R_{90}$	1	2	3	4
1	**0.980**	0.004	0.015	0.001	1	**0.997**	0.002	0.001	0.000
2	0.009	**0.850**	0.068	0.073	2	0.005	**0.952**	0.022	0.021
3	0.079	0.119	**0.711**	0.091	3	0.022	0.048	**0.833**	0.097
4	0.005	0.158	0.117	**0.720**	4	0.000	0.126	0.175	**0.699**

$\mathcal{D}^R_{80}$	1	2	3	4	$\mathcal{D}^R_{90}$	1	2	3	4
1	**0.999**	0.000	0.000	0.001	1	**1.000**	0.000	0.000	0.000
2	0.025	**0.960**	0.012	0.003	2	0.130	**0.863**	0.004	0.003
3	0.182	0.234	**0.523**	0.061	3	0.029	0.088	**0.828**	0.055
4	0.094	0.275	0.109	**0.522**	4	0.020	0.179	0.165	**0.636**

For what concerns the accuracy, our solution is able to go beyond the previously proposed method, since the overall average accuracy (considering all the datasets) is 5.5 percentage points above the mean accuracy of [20]. Regarding the confusion matrices, our results are more accurate than [20] in detection of classes 3 and 4. Indeed, for datasets $\mathcal{D}^R_{75}$ and $\mathcal{D}^R_{80}$ the diagonal terms corresponding to classes 3 and 4 present an average gap of $+0.15$ with respect to state-of-the-art, which achieves acceptable multi-classification outcomes especially when QF_f increases up to 90. Concerning the other classes, our results are comparable to [20] for single compressed images, while the detection of double compressed is slightly overwhelmed, probably dictated by a better accuracy of further compressions.

Robustness. In order to preliminary test the method's resilience to editing operations in between JPEG compressions, we applied our detector trained on $\mathcal{D}^R_{90}$ to images that randomly underwent either blurring or gamma correction in addition to compression. This campaign shows that if a single transformation is performed, accuracy drops approximately by 10%. The main effect of the transformations is to hide the previously applied JPEG compression. This paves the way to thrilling future research scenarios.

5. CONCLUSIONS

In this paper we propose a novel method for detecting multiple JPEG compressions, considering up to five coding steps. Our approach takes advantage of Task-driven Non-negative Matrix Factorization (TNMF) model, both for feature reduction and for classification, through a joint iterative estimation of dictionary and classifier. We extensively test several setups, taking into account different datasets and quality factors. These experiments show that our method outperforms up to date state-of-the-art [20] in terms of classification accuracy. Given the promising results in classification of furthest compression levels, future work will be devoted to raise the bar and go beyond the fourth coding cycle. Moreover, we are looking forward to focusing on the estimation of intermediate quality factors still concerning multiple JPEG compressions.

6. REFERENCES

[1] M. C. Stamm, M. Wu, and K. J. R. Liu, "Information forensics: An overview of the first decade," *IEEE Access*, vol. 1, pp. 167–200, 2013.

[2] Anderson Rocha, Walter Scheirer, Terrance Boult, and Siome Goldenstein, "Vision of the unseen: Current trends and challenges in digital image and video forensics," *ACM Computing Surveys (CSUR)*, vol. 43, pp. 26:1–26:42, 2011.

[3] A. Piva, "An overview on image forensics," *ISRN Signal Processing*, vol. 2013, pp. 22, 2013.

[4] T. Bianchi, A. Piva, and F. Pérez-González, "Near optimal detection of quantized signals and application to JPEG forensics," in *IEEE International Workshop on Information Forensics and Security (WIFS)*, 2013.

[5] M. Carnein, P. Schöttle, and R. Böhme, "Forensics of high-quality JPEG images with color subsampling," in *IEEE International Workshop on Information Forensics and Security (WIFS)*, 2015.

[6] M. Barni, Z. Chen, and B. Tondi, "Adversary-aware, data-driven detection of double JPEG compression: How to make counter-forensics harder," in *IEEE International Workshop on Information Forensics and Security (WIFS)*, 2016.

[7] M. C. Stamm, S. K. Tjoa, W. S. Lin, and K. J. R. Liu, "Undetectable image tampering through jpeg compression anti-forensics," in *IEEE International Conference on Image Processing (ICIP)*, 2010.

[8] C. Pasquini and G. Boato, "JPEG compression anti-forensics based on first significant digit distribution," in *IEEE International Workshop on Multimedia Signal Processing (MMSP)*, 2013.

[9] M. Kirchner and S. Chakraborty, "A second look at first significant digit histogram restoration," in *IEEE International Workshop on Information Forensics and Security (WIFS)*, 2015.

[10] Tiziano Bianchi and Alessandro Piva, "Detection of non-aligned double JPEG compression based on integer periodicity maps," *IEEE Transactions on Information Forensics and Security (TIFS)*, vol. 7, pp. 842–848, 2012.

[11] T.H. Thai, R. Cogranne, F. Retraint, and T.N.C. Doan, "JPEG Quantization Step Estimation and Its Applications to Digital Image Forensics," *IEEE Transactions on Information Forensics and Security*, vol. PP, no. 99, pp. 123–133, 2016.

[12] M. Barni, L. Bondi, N. Bonettini, P. Bestagini, A. Costanzo, M. Maggini, B. Tondi, and S. Tubaro, "Aligned and non-aligned double jpeg detection using convolutional neural networks," *Journal of Visual Communication and Image Representation*, vol. 49, pp. 153 – 163, 2017.

[13] T. Pevny and J. Fridrich, "Detection of double-compression in JPEG images for applications in steganography," *IEEE Transactions on Information Forensics and Security (TIFS)*, vol. 3, pp. 247–258, 2008.

[14] Tiziano Bianchi and Alessandro Piva, "Image forgery localization via block-grained analysis of jpeg artifacts," *IEEE Transactions on Information Forensics and Security (TIFS)*, vol. 7, pp. 1003–1017, 2012.

[15] Qing Wang and Rong Zhang, "Double JPEG compression forensics based on a convolutional neural network," *EURASIP Journal on Information Security*, vol. 2016, no. 1, pp. 23, 2016.

[16] Bin Li, Y.Q. Shi, and Jiwu Huang, "Detecting doubly compressed JPEG images by using mode based first digit features," in *IEEE Workshop on Multimedia Signal Processing (MMSP)*, 2008.

[17] V. Conotter, P. Comesaña, and F. Pérez-González, "Forensic analysis of full-frame linearly filtered JPEG images," in *IEEE International Conference on Image Processing (ICIP)*, 2013.

[18] Cecilia Pasquini, Giulia Boato, and Fernando Pérez-González, "Multiple JPEG compression detection by means of Benford-Fourier coefficients," in *IEEE International Workshop on Information Forensics and Security (WIFS)*, 2014.

[19] Fernando Pérez-González, Greg L Heileman, and Chaouki T Abdallah, "Benford's lawin image processing," in *Image Processing, 2007. ICIP 2007. IEEE International Conference on*. IEEE, 2007, vol. 1, pp. I–405.

[20] S. Milani, M. Tagliasacchi, and S. Tubaro, "Discriminating Multiple JPEG Compressions Using First Digit Features," *APSIPA Transactions on Signal and Information Processing*, vol. 3, no. May, pp. 1–11, 2014.

[21] Julien Mairal, Francis Bach, and Jean Ponce, "Task-driven dictionary learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 4, pp. 791–804, 2012.

[22] Victor Bisot, Romain Serizel, Slim Essid, and Gael Richard, "Supervised nonnegative matrix factorization for acoustic scene classification," *IEEE AASP Challenge on Detection and Classification of Acoustic Scenes and Events (DCASE)*, 2016.

[23] V. Bisot, R. Serizel, S. Essid, and G. Richard, "Feature Learning with Matrix Factorization Applied to Acoustic Scene Classification," working paper or preprint, Sept. 2016.

[24] Rafael C Gonzalez and Richard E Woods, "Image processing," *Digital image processing*, vol. 2, 2007.

[25] A. Popescu and H. Farid, "Statistical tools for digital forensics," in *International Conference on Information Hiding*, 2004.

[26] Romain Serizel, Victor Bisot, Slim Essid, and Gaël Richard, "Supervised group nonnegative matrix factorisation with similarity constraints and applications to speaker identification," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2017.

[27] Christopher M Bishop, "Pattern recognition," *Machine Learning*, vol. 128, pp. 209–210, 2006.

[28] Jorge Nocedal, "Updating quasi-newton matrices with limited storage," *Mathematics of computation*, vol. 35, no. 151, pp. 773–782, 1980.

[29] G. Schaefer and M. Stich, "UCID: an uncompressed color image database," in *Storage and Retrieval Methods and Applications for Multimedia 2004*, M. M. Yeung, R. W. Lienhart, and C.-S. Li, Eds., 2003, vol. 5307, pp. 472–480.

[30] Duc-Tien Dang-Nguyen, Cecilia Pasquini, Valentina Conotter, and Giulia Boato, "RAISE: A raw images dataset for digital image forensics," in *ACM Multimedia Systems Conference*, 2015.

[31] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro, "Online learning for matrix factorization and sparse coding," *Journal of Machine Learning Research*, vol. 11, no. Jan, pp. 19–60, 2010.