Tracing images back to their social network of origin: a CNN-based approach

Irene Amerini*, Tiberio Uricchio* and Roberto Caldelli*†
*Media Integration and Communication Center (MICC), University of Florence, Florence, Italy
†National Inter-University Consortium for Telecommunications (CNIT), Parma, Italy
{irene.amerini,tiberio.uricchio,roberto.caldelli}@unifi.it

***Abstract*—Recovering information about the history of a digital content, such as an image or a video, can be strategic to address an investigation from the early stages. Storage devices, smartphones and PCs, belonging to a suspect, are usually confiscated as soon as a warrant is issued. Any multimedia content found is analyzed in depth, in order to trace back its provenance and, if possible, its original source. This is particularly important when dealing with social networks, where most of the user-generated photos and videos are uploaded and shared daily. Being able to discern if images are downloaded from a social network or directly captured by a digital camera, can be crucial in leading consecutive investigations. In this paper, we propose a novel method based on convolutional neural networks (CNN) to determine the image provenance, whether it originates from a social network, a messaging application or directly from a photo-camera. By considering only the visual content, the method works irrespective of an eventual manipulation of metadata performed by an attacker. We have tested the proposed technique on three publicly available datasets of images downloaded from seven popular social networks, obtaining state-of-the-art results.**

I. Introduction

The pervasiveness of new technologies, such as smartphones, Internet and Social Networks (SN) made digital images and videos the primary source of visual information in nowadays society. Unfortunately, such multimedia data are often used to commit crimes by means of new modalities and aggressive behaviors. Attacks to personal reputation, cyberbulling, violence instigation and psychological harassments perpetrated online through social networks or messaging applications, like *Facebook*, *Whatsapp* and *Telegram*, represent very critical social issues. Gathering information about the record of an image or a video, could be strategic to address an investigation already from the early stages. In fact, by analyzing the multimedia material contained within storage devices, smartphones and PCs confiscated to a suspect, establishing their provenience can be crucial in leading the successive investigations. It is also desirable to identify the "history" of an image, tracing back the processes applied to a digital document, up to the data acquisition. This is particularly relevant when dealing with social networks where most of the user-generated photos and videos are uploaded and shared especially through mobile devices. To address such problems, new technologies able to analyze images and videos downloaded from social networks or forwarded via instant messaging apps are required.

Fig. 1: Overview of the approach. Images are processed through a CNN based technique to establish their provenance.

In this paper we propose a novel method to detect the most recent origin of an image. Given an image, it is able to detect if it has been acquired directly from the camera of a specific smartphone or if it has been downloaded from a particular social network or chat. Such a methodology is based on the idea of identifying the distinctive and permanent traces inevitably imprinted in each digital content during the processing applied to the data by the social platform. By opportunely exploiting such distinctive features it is possible to understand the provenance of a certain photo. The proposed technique achieves its task without resorting at any side information such as file size, values of image resolution and so on. Neither metadata (EXIF) are considered at all, although usually represent a not-negligible source of information, because they are not so reliable and, above all, they are often deleted from many social networks after the uploading of an image. The presented method extracts features in the image frequency domain to successively train an ad-hoc Convolutional Neural Network (CNN) in order to identify the origin of the to-be-checked image among disparate social networks and instant messaging apps (see Figure 1). Various experimental tests have been carried in different operative situations and with diverse image datasets; obtained results are provided and discussed to witness the good performances achieved demonstrating that the use of CNN really improves the outcome of this task.
The paper is organized as follows: Section II overviews some of the relevant related works and Section III introduces the proposed system describing its sequential phases. After that, Section IV presents some of the main experimental results

while Section V draws final conclusions and suggests some possible future developments.

II. Related works

The extensive use of CNN and Deep Learning in many areas such as image classification and annotation, object detection and so on [1], [2], [3], [4], has motivated and led the multimedia forensics community to comprehend if such technological solution is suitable both to detect image manipulations and to exploit camera identification. In order to detect image forgery, the work presented in [5] proposes to use the histograms of Discrete Cosine Transform (DCT) coefficients as input to a CNN to detect single or double JPEG compressions while the paper in [6] introduces a pre-processing module, before training a CNN, by using various high pass filters for the computation of residual maps in spatial rich model. After that, the net is fed with positive patches extracted from the borders of tampered images while the negative ones are randomly picked from authentic images. A multi-domain based CNN approach is proposed in [7] again to solve the image forgery detection task. The work explores the combined usage of a CNN trained on spatial domain patches (RGB) with another one which is provided by DCT histograms as input.
Regarding source identification, CNNs are mainly used for the camera model identification purpose. In particular, the authors in [8] proposed a pre-processing layer, consisting of a high pass filter which is applied to the input image before exploiting the CNN for the detection of camera models. Trying to solve the same task as before, the authors in [9] have realized a CNN to extract features given as input to a battery of SVMs (Support Vector Machine) for the classification phase.
Source identification of social network images is a very new and hot topic, very few state of the art works exist and CNN approach has not been used for this task so far. In [10] a drafting procedure to distinguish among different social networks is presented by using resizing, compression, renaming and metadata alterations left by the upload/download system platforms using a K-NN classifier. Furthermore, in [11], a classification method among *Facebook*, *Flickr* and *Twitter* images is exploited by adopting only pixel-based information deriving from DCT histograms of JPEG images; social network identification is achieved by means of bagged decision tree classifier. The method is evaluated on two different datasets that are also employed in the proposed paper (see experimental results within Section IV).

III. Proposed method

The task of social media images classification is defined as follows. Given an image x, we define a function $f(x)$ that outputs the social network from where x is originated. $f(\cdot)$ operates a categorization task where the output is one of several SNs. The main underlying assumption is that each social network has a different process when they handle images [11]. When an image is uploaded on a social network, it undergoes a specific processing which typically includes JPEG compression but also optionally resizing and some filtering to adapt the quality of it. Even if the actual process and their parameters (such as the quality of compression) are not known, some distinctive features on the images are left. As a result, they can be detected by a classifier. Similarly as in [11], our hypothesis is that such features are mainly related to JPEG compression parameters. We then naturally choose to employ DCT-based features, since they are strongly affected by JPEG compression. Moreover, they are successful in similar tasks such as detecting double compression [12], [13] and they were also employed in recent deep learning approaches like in [5] and [7].
Based on this, a frequency domain based CNN has been devised taking as input a statistical representation of the DCT coefficients and directly outputs the class of the social network that originated the image. Assuming that we have K social networks, the network will thus have K classes in the network output. The next two subsections will be dedicated to present the DCT-based features (Section III-A) and the architecture of the convolutional neural network (Section III-B) respectively.

A. DCT-based features

Considering that a CNN needs an input of a fixed size, we decided to use the defined amount of histograms of image DCT coefficients [7]. Being DCT mainly affected by the content and size of the considered image [5] and in order to be independent with respect to the image resolution, each picture is subdivided in non-overlapping patches instead of processing the entire image as one input only. Each patch is then fed to the network and the outputs are finally refined to get the final class. In particular, given an $N \times N$ image patch, DCT coefficients are first extracted and for each 8×8 block in a patch, the first 9 spatial frequencies in zig-zag scan order beside the DC coefficient are selected. For each spatial frequency (i, j) (i.e. mode), the histogram $h_{(i,j)}$ representing the occurrences of the quantized DCT coefficients is built. The term $h_{(i,j)}(m)$ will indicate the occurrences of the values m in the histogram of mode (i, j) of the DCT coefficients with $m = (-50; 0; +50)$. So, the network takes a vector of 909 elements (101 histogram bins $\times$ 9 DCT frequencies) as input.

B. CNN architecture and final class prediction

The proposed CNN model is based on similar ideas taken from the image classification literature, like in [1] and in [7] and its architecture is illustrated in Figure 2. Each $N \times N$ input patch is pre-processed to compute, as described before, the feature vector (size 909×1) which is then fed to the network to obtain one of the K social network classes. We employ two blocks comprised of one-dimensional convolutional block followed by max-pooling layers to reduce dimensionality and computational requirements of the approach. Then, three fully connected layers are employed to calculate the final output of the network. Each convolutional block is defined as:

$$f(x) = g(W * x + b) \tag{1}$$

where $*$ is the convolutional operator, W are 1-D weights of the layer, b is the bias and g is a non-linear activation function.

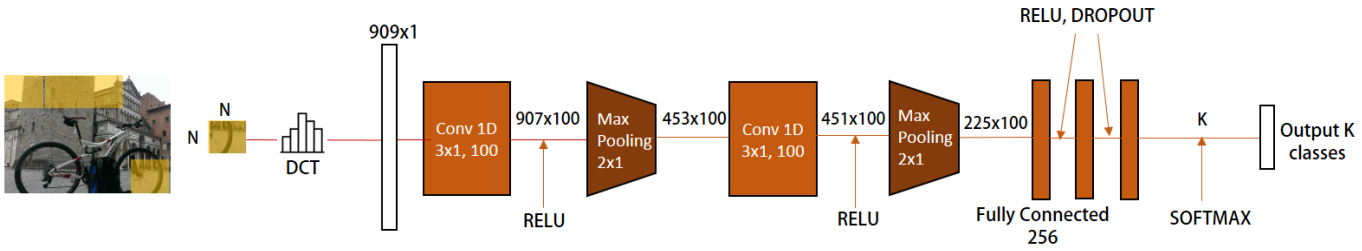

Fig. 2: Architecture of the proposed CNN.

Due to their low complexity and good performance [1], we use the rectified linear units (ReLUs) $g(x) = max(0; x)$ as the activation function. We empirically chose W and b sizes of 100 filters, that proved to have good performance in our preliminary experiments.

The fully connected layers are defined as:

$$f(x) = g(W \cdot x + b) \tag{2}$$

where, similarly as in the convolutional layers, W are the weights of the layer, b is the bias and g is a non-linear activation function (ReLUs in our case). We choose a dimensionality of 256 for the first two fully connected layers and also employ dropout [14] that proved to be a helpful solution during preparatory analysis. The final layer has instead K outputs that are sent to a softmax layer, in order to obtain the final probability of each social network class.

After processing each image patch with the CNN, a set of predictions is obtained at patch level; to propagate the classification at image level, we have taken the SN class with the higher score all over the image.

IV. Experimental results

In this section a description of the image databases employed for the experiments is firstly outlined hereafter and then some of the experimental tests carried out are presented in the successive subsections. Different datasets have been used for the experimental tests: the two sets named *UCID social* and *Public social*[1] and the dataset named *Iplab*[2] (see in [11] and in [10] for details respectively).

The first dataset, *UCID social*, has been created with digital images taken from UCID (Uncompressed Colour Image Database) database [15] which is composed by 1338 images (512x384 pixels) in TIFF format. Starting from this, JPEG compressed images are generated at different quality factors $QF = 50 : 95$ (step 5). JPEG images, made according to this process, are uploaded for each selected social network (*Flickr*, *Facebook* and *Twitter* in this case) and then downloaded in order to be analyzed. Specifically, the *UCID social* is composed by 30000 images (1000 images$\times$10 QFs$\times$3 social networks).

The second dataset, called *Public social*, constitutes a more variable and challenging set; it is composed by 3000 uncontrolled images (different sizes, JPEG quality factors and contents) downloaded from different social networks (*Flickr*, *Facebook* and *Twitter*, 1000 images for each of them).

The third dataset, a selection of the *Iplab* database, contains images coming from 7 different platforms (5 social networks: *Facebook*, *Flickr*, *Google+*, *Instagram*, *Twitter* and 2 instant messaging apps: *WhatsApp* and *Telegram*) and a set of unprocessed JPEG images (directly acquired by a photo-camera) for a total of 8 classes. Each class is composed by 240 images with different sizes and contents (outdoor and indoor) and are acquired with different smartphones at two different resolutions: the higher and lower quality resolution allowed by the device. Images for the 7 different platforms are obtained by means of a procedure of uploading and then downloading on each social network.

Each of the considered dataset has been subdivided in training set (80%), validation set (10%) and test set (10%) in order to keep separate the bunches of images involved in the different phases. The neural network, described in the previous section, learns on different patches (as described in Section III-B, non-overlapping and of dimension 64$\times$64 pixels) depending on the databases for each of the K classes and is optimized by using AdaDelta method [16]. The training phase is stopped when the loss function on the validation set reaches its minimum that usually happens after ten/twenty epochs.

A. Three classes patch-level evaluation

In this subsection experiments dedicated to investigate the social network of provenance of test images are presented at patch level; moreover, in this case, the classifier is trained to recognize only three social networks (classes $K = 3$): *Flickr*, *Facebook* and *Twitter*. Results for each of the three social networks are reported evaluating the performances on all the three datasets (obviously, in the case of *Iplab* only the images coming from *Facebook*, *Flickr* and *Twitter* are selected for this experiment). The CNN assigns a class for each patch 64$\times$64 thus also demonstrating the ability of prediction in the case of very small images. Table I shows the confusion matrix obtained for the test set of the *UCID social* dataset while in Table II and III confusion matrices for the other two datasets (*Public social* and *Iplab*) are reported respectively. It is evident that the system provides good performances with

[1] http://lci.micc.unifi.it/labd/2015/01/trustworthiness-and-social-forensic/
[2] http://iplab.dmi.unict.it/DigitalForensics/social_image_forensics/

a percentage of correct classification of about 90%. Results presented witness that the classification capacity of the method is satisfactory both in a controlled scenario like *UCID social* (avg. 98.41%) and in open scenarios represented by the *Public social* (avg. 87.60%) and *Iplab* (avg. 90.89%) datasets.

TABLE I: *UCID social* dataset: classification among *Facebook*, *Flickr* and *Twitter*.

Classification (%) vs SNs	Facebook	Flickr	Twitter
Facebook	96.15	0.19	3.66
Flickr	0.03	99.79	0.18
Twitter	0.59	0.11	99.30

TABLE II: *PUBLIC social* dataset: classification among *Facebook*, *Flickr* and *Twitter*.

Classification (%) vs SNs	Facebook	Flickr	Twitter
Facebook	84.23	4.60	11.17
Flickr	4.21	88.80	6.99
Twitter	7.83	2.39	89.78

TABLE III: *IPLAB* dataset: classification among *Facebook*, *Flickr* and *Twitter*.

Classification (%) vs SNs	Facebook	Flickr	Twitter
Facebook	94.00	6.00	0.00
Flickr	1.76	92.13	6.11
Twitter	0.00	13.47	86.53

B. Three classes image-level evaluation

Similarly to what has been done before, we considered the three datasets with respect to three social networks but this time a full frame evaluation (image-level) is taken into account, in order to make a comparison with a pixel-based technique proposed in [11]. As indicated in Section III-B, the predicted class for an image I is obtained by majority voting on the number of patches assigned to the different classes.
In Figure 3, the comparison on the three datasets between the proposed CNN-based method and the technique presented in [11] is displayed. Performances on the test set on the full frame images are evaluated in terms of *True Positive Rate* ($TPR = \frac{TP}{TP+FN}$) for sake of readability. The proposed method performs slightly better with respect to the other one when a controlled scenario (*UCID social*) is taken into account (Figure 3 left side). On the contrary, when the other two datasets are considered, the CNN-based method is able to better generalized and the performances increase with respect to [11] as evidenced in Figure 3 (central and right side). Furthermore, in Tables IV, V and VI an extended comparison of results is shown in terms of correct classification and misclassification rates. Correct classification percentages are averagely around 95% with the proposed method while using [11] they are at about 88%; the percentages of incorrect classification are generically reduced using the CNN approach. It can be pointed out that, as expected, according to the criterion chosen for full-frame decision propagation, the performances improve compared to the patch-level case.

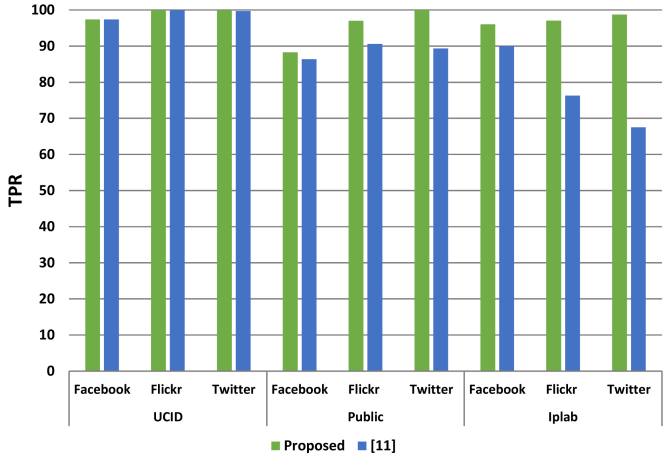

Fig. 3: Comparison on three datasets in terms of TPR between the proposed CNN-based method (*green*) and the one in [11] (*blue*).

TABLE IV: Classification among *Facebook*, *Twitter* and *Flickr* (*UCID social* dataset). Full frame evaluation of the proposed method (Prop.) compared with [11].

	Facebook		Flickr		Twitter	
	Prop.	[11]	Prop.	[11]	Prop.	[11]
Facebook	97.37	97.42	0.00	0.00	2.63	2.58
Flickr	0.00	0.00	100	100	0.00	0.00
Twitter	0.00	0.33	0.00	0.00	100	99.67

TABLE V: Classification among *Facebook*, *Twitter* and *Flickr* (*Public social* dataset). Full frame evaluation of the proposed method (Prop.) compared with [11].

	Facebook		Flickr		Twitter	
	Prop.	[11]	Prop.	[11]	Prop.	[11]
Facebook	88.24	86.34	0.00	3.08	11.76	10.58
Flickr	0.99	5.58	97.03	90.59	1.98	3.83
Twitter	0.00	10.00	0.00	0.67	100	89.33

TABLE VI: Classification among *Facebook*, *Twitter* and *Flickr* (*Iplab* dataset). Full frame evaluation of the proposed method (Prop.) compared with [11].

	Facebook		Flickr		Twitter	
	Prop.	[11]	Prop.	[11]	Prop.	[11]
Facebook	96.01	90.00	3.99	8.75	0.00	1.25
Flickr	1.68	8.75	97.06	76.25	1.26	15.00
Twitter	0.00	10.00	1.26	22.50	98.74	67.50

TABLE VII: *Iplab* dataset: classification among five social networks (*Facebook*, *Flickr*, *Google+*, *Instragram*, *Twitter*), two instant messaging apps (*WhatsApp*, *Telegram*) and one unprocessed group of JPEG images (*Original*)

Classification (%) vs SNs	Facebook	Flickr	Google+	Instagram	Original	Telegram	Twitter	WhatsApp
Facebook	**87.12**	3.84	0.17	1.65	0.70	6.32	0.17	0.04
Flickr	0.10	**85.72**	0.04	0.10	3.72	7.42	2.63	0.27
Google+	0.04	0.84	**84.54**	0.00	13.48	1.01	0.06	0.03
Instagram	0.06	0.71	0.14	**97.71**	0.99	0.25	0.00	0.14
Original	0.00	0.27	0.44	0.01	**99.03**	0.24	0.00	0.01
Telegram	0.01	0.15	0.01	0.00	1.56	**98.25**	0.02	0.00
Twitter	0.04	4.26	0.00	0.00	1.06	0.43	**94.21**	0.00
WhatsApp	0.0	0.15	0.02	0.00	1.03	0.02	0.00	**98.78**

C. Eight classes patch-level evaluation

Finally, we evaluate the proposed methodology over a selection of the *Iplab* dataset composed by images coming from *Facebook*, *Flickr*, *Google+*, *Instragram*, *Telegram*, *Twitter*, *WhatsApp* and by an unprocessed group of JPEG images (named *Original*).

The CNN is now trained on $K = 8$ classes of non-overlapping image patches of size $N = 64$. Results, over a test-set of 30132 image patches, are presented in Table VII. The method is able to classify the different social networks very well demonstrating the robustness of the proposed approach with a percentage of classification over 90% averagely.

In most of the cases the prediction is over 95% (i.e. *Instagram*, *Original*, *Telegram*, *Twitter* and *WhatsApp*); on the contrary in the case of *Google+* some of the patches are recognized as *Original* that could be a sign of less intrusive modifications on the images of *Google+* platform with respect to other SNs. As before, also in this case, performances improve of about 5% averagely when propagating the decision at image-level; another table has not been inserted to avoid redundancy.

V. Conclusions and future works

This paper has proposed a new methodology based on convolutional neural networks (CNN) to go back to the social network of provenance of a certain image without resorting at its metadata. The presented technique has been tested on three public datasets until seven most common social networks or instant messaging applications. The obtained results demonstrated a good ability of the proposed CNN-based approach to distinguish among different social platforms.

Future works will be devoted to increase the number of the considered social networks, evaluating also different kinds of CNN architectures. Another interesting topic will be to understand the behavior of the proposed method in the case of multiple upload/download, i.e JPEG images that firstly have been uploaded-downloaded on a social network (e.g. *Facebook*), then uploaded-downloaded on another one (e.g. *Instagram*).

References

[1] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proceedings of the Neural Information Processing Systems (NIPS)*, 2012.

[2] R. Girshick, J. Donahue, T. Darrell, and J. Malik, "Rich feature hierarchies for accurate object detection and semantic segmentation," in *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014, pp. 580–587.

[3] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *Proceedings of the International Conference on Learning Representations (ICLR)*, 2015.

[4] T. Uricchio, L. Ballan, L. Seidenari, and A. Del Bimbo, "Automatic image annotation via label transfer in the semantic space," *Pattern Recognition*, vol. 71, pp. "144 – 157", 2017.

[5] Q. Wang and R. Zhang, "Double jpeg compression forensics based on a convolutional neural network," *EURASIP Journal on Information Security*, vol. 2016, no. 1, p. 23, 2016. [Online]. Available: http://dx.doi.org/10.1186/s13635-016-0047-y

[6] Y. Rao and J. Ni, "A deep learning approach to detection of splicing and copy-move forgeries in images," in *Proceedings of the IEEE International Workshop on Information Forensics and Security (WIFS)*, 2016, pp. 1–6.

[7] I. Amerini, T. Uricchio, L. Ballan, and R. Caldelli, "Localization of jpeg double compression through multi-domain convolutional neural networks," *Proc. of IEEE CVPR Workshop on Media Forensics*, 2017.

[8] A. Tuama, F. Comby, and M. Chaumont, "Camera model identification with the use of deep convolutional neural networks," in *Proceedings of the IEEE International Workshop on Information Forensics and Security (WIFS)*, Dec 2016, pp. 1–6.

[9] L. Bondi, L. Baroffio, D. Gera, P. Bestagini, E. J. Delp, and S. Tubaro, "First steps toward camera model identification with convolutional neural networks," *IEEE Signal Processing Letters*, vol. 24, no. 3, pp. 259–263, March 2017.

[10] O. Giudice, A. Paratore, M. Moltisanti, and S. Battiato, "A classification engine for image ballistics of social data," *CoRR*, vol. abs/1610.06347, 2016. [Online]. Available: http://arxiv.org/abs/1610.06347

[11] R. Caldelli, R. Becarelli, and I. Amerini, "Image origin classification based on social network provenance," *IEEE Transactions on Information Forensics and Security*, vol. 12, no. 6, pp. 1299–1308, June 2017.

[12] T. Bianchi and A. Piva, "Image forgery localization via block-grained analysis of JPEG artifacts," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 3, pp. 1003–1017, 2012.

[13] I. Amerini, R. Becarelli, R. Caldelli, and A. Del Mastio, "Splicing forgeries localization through the use of first digit features," in *Proceedings of the IEEE International Workshop on Information Forensics and Security (WIFS)*, 2014, pp. 143–148.

[14] N. Srivastava, G. E. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting." *Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[15] G. Schaefer and M. Stich, "UCID - an uncompressed colour image database," in *Proceedings of the Storage and Retrieval Methods and Applications for Multimedia*, 2004, pp. 472–480.

[16] M. D. Zeiler, "Adadelta: an adaptive learning rate method," *CoRR*, vol. abs/1212.5701, 2012.