An Adaptive Neural Network for Unsupervised Mosaic Consistency Analysis in Image Forensics

Quentin Bammey, Rafael Grompone von Gioi, Jean-Michel Morel

An Adaptive Neural Network for Unsupervised Mosaic Consistency Analysis in Image Forensics

Quentin Bammey Rafael Grompone von Gioi Jean-Michel Morel
Université Paris-Saclay, ENS Paris-Saclay, CNRS, Centre Borelli
{quentin.bammey, grompone, morel}@ens-paris-saclay.fr

Abstract

Automatically finding suspicious regions in a potentially forged image by splicing, inpainting or copy-move remains a widely open problem. Blind detection neural networks trained on benchmark data are flourishing. Yet, these methods do not provide an explanation of their detections. The more traditional methods try to provide such evidence by pointing out local inconsistencies in the image noise, JPEG compression, chromatic aberration, or in the mosaic. In this paper we develop a blind method that can train directly on unlabelled and potentially forged images to point out local mosaic inconsistencies. To this aim we designed a CNN structure inspired from demosaicing algorithms and directed at classifying image blocks by their position in the image modulo (2×2). Creating a diversified benchmark database using varied demosaicing methods, we explore the efficiency of the method and its ability to adapt quickly to any new data.

1. Introduction

Detecting image forgeries is a problem with critical applications ranging from the control of fake news in online media and social networks [59] to the avoidance of scientific misconduct involving image manipulation[1]. Images are easy to modify in a visually realistic way, but those modifications can be difficult to detect automatically.

The most common image forgery techniques are copy-move, both internal and external (splicing), inpainting and enhancement, which may include a modification of the hue, contrast, brightness, etc., of an image to hide objects or change their meaning [22, 75]. The settings in which these images are created and distributed may further alter the image and hinder certain detection methods. For instance, uncompressed images have characteristic demosaicing and noise signatures which are nearly erased by a strong compression. On the other hand, detection in tampered JPEG images may be based on the inconsistency of JPEG encoding caused by splicing [56, 33, 21, 43, 6, 7, 42]. Yet, this detection method is so efficient that research on counterforensics has been very active and has proposed efficient ways to reinstate a coherent JPEG encoding after forgery [62, 63, 66, 20].

There are two concurrent paradigms for forgery detection techniques. The first way consists in developing many different methods, that address separately the varied forgeries and inconsistencies created by these forgeries. Error Level Analysis (ELA) [34] fits in this category and creates a heatmap by recompressing the image and visualising the difference. As we just mentioned, many methods look for inconsistencies in JPEG encoding; many other try to detect noise discrepancies [36, 58, 14, 27, 48, 49, 5, 51, 48, 49, 5, 51, 41, 67, 35, 16, 54, 47, 44, 76, 74] or attempt to directly detect internal copy-move operations [68, 71, 70, 61, 1, 23]. The variety of setups before and after forgery makes exhaustiveness difficult, yet results obtained by such specific methods are self-explanatory. However, with few exceptions such as the recent development of Siamese Networks, which we briefly describe in Section 2, most of these methods are created manually, which can limit their performances, especially when forged images are created with a combination of methods rather than just one move.

Another possibility is to consider forgery detection as a unique learning problem and develop a structure – usually a neural network – to classify and/or localise forgeries independently of the setup and forgery type. For instance in [72] a heat map is computed, in [3] the network segments the image into forged and non-forged parts. See also [10] and [4]. While exhaustiveness is theoretically possible with these methods, it is actually limited by the database itself: they learn how to detect forgeries as seen in a training database, and can thus fail when confronted with images whose forgeries were made differently.

In this article, we choose to focus on the detection of demosaicing artefacts to detect forged regions on an image. Most cameras cannot directly capture colour. In order to

[1] https://en.wikipedia.org/wiki/List_of_scientific_misconduct_incidents

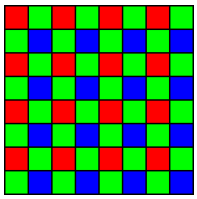


Figure 1: The Bayer matrix is the most common CFA. Each pixel is represented as the colour in which the camera samples it.

create colour images, they instead use a filter, named colour filter array (CFA) or mosaic, before the light rays reach the camera's sensor. As a consequence, each pixel is sampled in only one colour, and the other colours must be interpolated from neighbouring pixels sampled in other colours. These interpolation algorithms leave artefacts that can be detected to know in which colour each pixel was sampled.

The most commonly used CFA is by far the Bayer matrix, shown in Fig. 1, which samples two pixels of green for one pixel of red and another of blue. Although other CFA exist, their use is marginal. As a consequence, we only consider the Bayer matrix in our article.

When an image is forged by copying part of this image or of another image onto it, there is a $\frac{3}{4}$ probability that the mosaic of the forged region will not be aligned to that of the main image. As a consequence, locally detecting the position of the mosaic in images can lead to finding discrepancies caused by forgeries.

While detecting the presence of demosaicing can be done reliably with current state-of-the-art methods, the interpretation of these artefacts is still a challenge. Most methods make assumptions of linearity of the interpolation or even assume the colour channels to be independently demosaiced. These assumptions are invalid with most commonly-used demosaicing methods, and even state-of-the-art mosaic detection algorithms thus tend to yield a large number of false positives.

Many different demosaicing algorithms exist, furthermore most of those used in commercial cameras are undisclosed. Learning-based methods must thus take into account the impossibility to learn on all existing algorithms.

In this paper, we overcome the above limitations by using an unsupervised convolutional neural network that learns to detect changes in the underlying pattern of mosaic artefacts. This network can be trained on unlabelled authentic images to detect forgeries in new images. Similarly to zero-shot learning, it can also train directly on a database of potentially forged images to adapt to JPEG compression.

The contributions of our article are three-fold. We create a new convolutional neural network (CNN) structure tailored to the specific task of mosaic artefacts detection, and that beats state-of-the-art mosaic detection methods. It can be trained in a fully unsupervised manner, and can even be directly retrained on a set of images to adapt to their specific conditions. To do that, we propose a new use for pixelwise convolutions in neural networks. Their main use in the literature has been to reduce the dimensionality of a network before performing heavier spatial operations, such as in [64]. We argue that they can also be used stacked to each other, to process the causality relations between previously-computed spatial features as, for the same price as spatial convolutions, they can have more and bigger layers; furthermore, they do not add any more spatial dependency to the results. Finally, working over the Dresden image dataset [28], we create a new dataset aimed specifically at benchmarking forgery detection via demosaicing artefacts.

Both the code and the dataset can be found on https://github.com/qbammey/adaptive_cfa_forensics.

2. Related works

Identification of demosaicing artefacts for forgery detection is not a new subject. A pioneer paper on this field is provided by [57]. They propose to work independently on the different colour channels, and use an expectation-maximisation (EM) algorithm to jointly compute a linear estimation of the demosaicing algorithm and find the probability of each pixel being interpolated or originally sampled. They then apply the Fast Fourier Transform (FFT) on the pseudo-probability map to detect changes in the magnitude and phase at the 2-periodicity peaks, which can correspond to changes in the CFA artefacts.

[29, 2] improve on [57] by replacing the EM algorithm with a direct linear estimation of the algorithm in all four possible positions. [29] uses the Discrete Cosine Transform (DCT) instead of the FFT in order to see changes of the mosaic, which can correspond to copy-move forgeries, as a change in the sign of the DCT, which is easier to see than a change of phase in the FFT. [2] notes that in a scenario where many different methods are needed to detect the variety of forgeries, it becomes especially important to strictly control the number of false positives for each of them. They propose a simple method to detect the presence of a significant CFA pattern, by pooling the error map into blocks, each of which votes for one of four grids. In the absence of demosaicing, the votes should be uniformly distributed between the four grids. They thus look at the number of votes for each position, and threshold the detection on the rate at which a detection at least as significant would happen in the absence of demosaicing. All three methods make two strong assumptions:

- they assume that demosaicing is done independently in each colour channel, and
- they assume that a linear estimation can sufficiently represent the demosaicing algorithm.

Even though these two assumptions might have been at least partially true in the green channel for most demosaicing algorithms commonly used in 2005, when [57] was first published, it is far from being the case nowadays.

Another important method is provided by [40]. They propose to directly detect the mosaic used in the image, and to do that, they mosaic the image in all four possible positions, and redemosaic it using a simple algorithm such as bilinear interpolation. The reasoning is that the demosaicing should produce an image closer to the original when remosaicked and demosaiced in the correct position. They thus compare the residual maps to detect which of the possible mosaics has been used. Claiming demosaicing artefacts can usually be seen more clearly in the green channel, they first decide on the position of the green sampled pixels. They then use the most significant of the red and blue channels to decide on the remaining to position. This order of decision has been used in most of the literature since then. Their use of the bilinear algorithm limits them in the same way as [57, 29] because of the linearity and colour independence of the bilinear algorithm, which is not shared by most modern demosaicing algorithms. However, their method does not depend on the choice of the algorithm, and they can thus provide very good results in the rare case where the demosaicing algorithm of a studied image is known.

In order to detach themselves from one specific algorithm, [11] notes that pixels are more likely to feature locally extremal values in the channel in which they are sampled, and on the contrary to take intermediary values where interpolated. As a consequence, they count the number of intermediate values in all four positions to decide which position is the correct one, using the decision pipeline introduced in [40]. The assumption that pixels are more likely to take extremal values in their sampled channel is usually verified with most algorithms, which leads this method to yield good classification scores. However, the probability bias can sometimes be reversed when algorithms make extensive use of other channels' high frequencies, which can lead to some regions of the image being detected in a wrong position with a strong confidence.

[60] is the first method that tries to alleviate the colour channels independence assumption. Instead of working separately in each channel, they compute the difference of the green channel separately with both the red and blue channels. Using the variance of those differences, they decide on the correct position using a similar pipeline as above. Although the colour independence is hard-coded, the colour difference is used in many current algorithms. Using this instead of the raw channels thus provide a first step toward a correct understanding of demosaicing artefacts.

[45] is currently, to the best of our knowledge, the only method offering to use a neural network for mosaic detection. They notice that most forgery detection methods involve first computing a residual error map, as in [40], or a similar feature map, as in [57, 29, 11, 60], and then interpreting it, for instance with the FFT in [57]. They first compute an error map based on the green channel, then use a CNN to interpret the error map, and distinguish forgeries from post-processing steps such as JPEG compression. However, distinguishing demosaicing artefacts from JPEG or resampling artefacts can already be seen with simple methods such as [57]'s FFT or [2]'s *a contrario* approach. With most current methods, the first source of indistinguishable errors in the feature map come not from post-processing applied to the image – which can hinder CFA detection rather than create false detections without being visible with manual method – but from a lack of fit between the detection method and the image's demosaicing. As a consequence, we believe that a CFA detection method would benefit from the use of a neural network more in the computation of the feature map than in the interpretation thereof. They do claim high accuracy results. Unfortunately, their tests were made on raw images from the Dresden database [28], which they thus demosaic themselves, without indicating which algorithm has been used, and no code is provided to verify the results.

Neural networks have also gained popularity in image forensics in the form of Siamese networks [8]. The goal of these networks is to compare two samples. Features for both samples are processed with a first network with shared weight, and a second network is applied to the residual between the two samples to decide on their similarity. This approach has already been successfully applied in several areas of forensics, including camera source detection [50], and prediction of the probability of two patches sharing the same EXIF data [32] for splicing detection. While we could use Siamese networks to compare the CFA pattern of different patches, Siamese networks are especially powerful to compare patches when classification is cumbersome – for instance because of a high number of classes, some of which may not be present in the training data –, in which case it can become more practical to directly compare patches without explicitly classifying them. On the other hand, the mosaic of an image belongs to one of four classes. This means that Siamese network do not necessarily offer an advantage to CFA grid detection, and we can probably use directly a classifying network, which is less complex as we do not need to compare all pairs of patches.

3. Proposed method

A standard approach to finding copy-move forgeries through demosaicing artefacts would be to first detect the image's initial mosaic, and then detect if parts of the image actually have a different mosaic. Our manual attempts to detect the original mosaic were not successful. Indeed, cri-

teria to do this heavily depend on the demosaicing method. Instead, we designed a convolutional neural network (CNN) to train on blocks of the image and directly predict their position in the image modulo $(2, 2)$. The only cue to this relative position are the periodic artefacts, such as CFA, resampling and JPEG artefacts. Hence, a change of the mosaic can lead to forged blocks being detected at incorrect positions modulo (2,2) and thus flagged as forged. Because the target output is only the relative position of blocks on the image, all that is required to train the network is a set of demosaiced images, without additional labels.

In a standard unsupervised scenario, the CNN can be trained with many authentic images and then used on new images to detect forgeries on them. However, if we have to detect forgeries on a large database, and if we can assume that the images in the database are similar in terms of demosaicing and post-processing – and in particular JPEG compression –, then we can retrain the CNN, performing unsupervised transfer learning directly on the test data. As the forged regions generally occupy a small part of the images, and only a small proportion of the images under study are forged, the risk that the CNN will overfit on the forged regions will be small.

The network consists of several parts, all of which serving different purposes. It only uses 31,504 trainable parameters. In the initial training phase, overfitting can occur both on the image contents and on the specific algorithms used for demosaicing. Although the former can easily be avoided by using more images for training, avoiding overfitting on the algorithms is harder. The small size of the network thus helps to avoid overfitting during training. It is even more useful when retraining on the same images to be studied, as overfitting on those images is much harder to avoid, and can make the network miss forgeries.

3.1. Spatial network

The first layers extract spatial features from the images. Due to the nature of demosaicing, we make use of two specific types of convolutions.

Most demosaicing algorithms try to avoid interpolating against a strong gradient [25], which would lead to visual artefacts. As a consequence, they often interpolate in one direction along edges. To mimic this, the first layers perform 10 horizontal, 10 vertical and 5 full convolutions, which are concatenated at the end of each layer.

In a mosaiced image, only one in four pixels is red and one in four is blue. As visualised in Fig. 2, this means that at the location of a sampled pixel, the closest sampled neighbours are all located at 2 pixels distance horizontally and/or vertically of the current position. We can take advantage of this by using dilated convolutions, which will only involve pixels belonging to the same mosaic.

We first use a sequence of two layers of 10 horizontal 1×3, 10 vertical 1×3 and 5 full 3×3 convolutions. In parallel, we perform 10 horizontal, 10 vertical and 5 full convolutions, which are all 2-dilated. The outputs of both parts are concatenated with a skip-connection from the input image. To this output is applied a similar sequence of two layers of 10 horizontal, 10 vertical and 5 full convolutions, in parallel with 10 horizontal, 10 vertical and 5 full convolutions with a dilation of 2. The spatial output is the concatenation of the output of the second and fourth non-dilated convolutions, and of the two dilated convolutions.

All layers in this part of the network are separated by a leaky rectified linear unit [46]. A diagram of this structure can be found in Fig. 3.

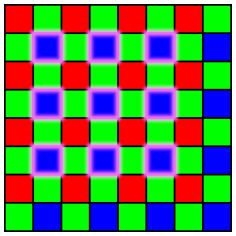


Figure 2: If we use a 3×3 convolution with a dilation of 2, the convolution at the central pixel sampled in blue only involves pixels sampled in the same colour. More generally, a 2-dilated convolution will look at pixels that all belong to the same colour channel.

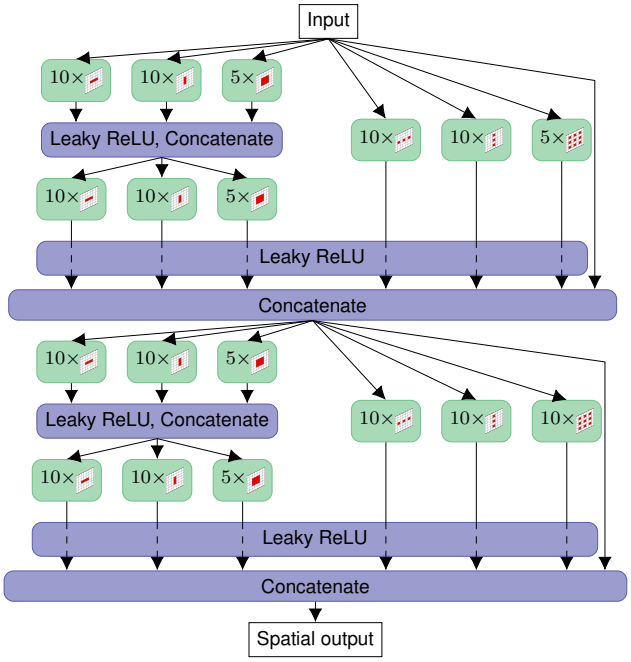


Figure 3: Spatial part of the network, containing 17,160 trainable parameters

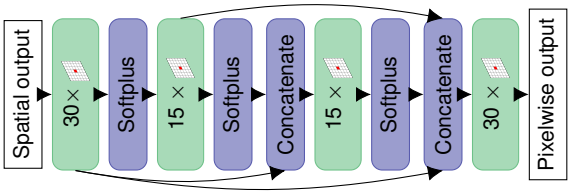


Figure 4: Pixelwise 1×1 convolutional part of the network, containing 6105 parameters

3.2. Pixelwise Causal network

Summarising, the network uses values that are up to four pixels away both horizontally and vertically from each pixel (the receptive field is thus 9×9). We consider this spatial span sufficient. Indeed, most demosaicing algorithms do not look farther to demosaic a given pixel. However, some algorithms still feature complex transfers between the different colour channels, especially in the high frequencies. As a consequence, the second part of our network consists of pixelwise (1×1) convolutions, which enable us to capture complex causal relations without adding more spatial dependencies to the convolutions. Although pixelwise convolutions are often used in the literature, their primary use is often to reduce data dimensionality. The Inception network [64], uses pixelwise convolutions before large convolutions to reduce dimensionality. Other networks use depthwise separable convolutions, where standard convolutions are replaced with one depthwise convolution followed by a pixelwise convolution [12, 31].

In our network, however, we do not stack them to reduce dimensionality, but to perform complex operations after the spatial features have been computed. Linking pointwise convolutions with each other enables us to represent complex relations at a low computational cost, with few parameters and without incrementing spatial dependency.

This part of the network consists of four layers of respectively 30, 15, 15 and 30 1×1 convolutions. The output of the first convolution is skip-connected to the third and fourth convolutions, and the output of the second convolution is skip-connected to the fourth convolution. As a consequence, the last convolution takes the results of all previous pointwise layers into consideration to prepare features for the next step.

All the layers in this part of the network are separated by Softplus activation [18]. A diagram of the structure can be seen in Fig. 4.

3.3. Blocks preparation

Although relative positions could be detected at the pixel level, grouping the pixels into blocks can lead to more reliable predictions. However, the blocks must be created carefully in order to avoid any bias.

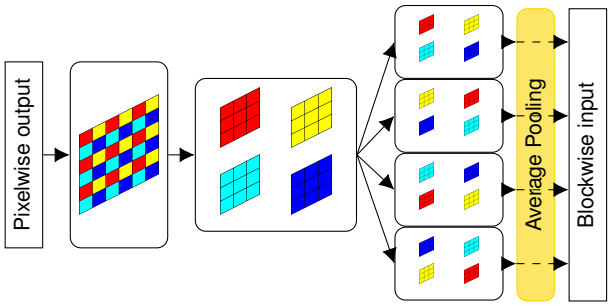


Figure 5: Processing the image into blocks

Given an input image I of shape $(2Y, 2X, C)$, where C is the number of channels ($C = 30$ after our pixelwise network) and $2Y$ and $2X$ represent the spatial dimensions, we start by splitting the four modulo $(2, 2)$ positions of this image. We thus create four images I_{00}, I_{01}, I_{10} and I_{11}, each of shape (Y, X, C) and defined by

$$I_{\delta_x \delta_y}[y, x, c] = I[2y + \delta_y, 2x + \delta_x, c]. \tag{1}$$

We then concatenate these four images in different ways into four new images J_{00}, J_{01}, J_{10} and J_{11}, each of shape $(Y, X, 4C)$ and defined as follows:

$$\begin{aligned} J_{\delta_x \delta_y}[y, x, 4c] &= I_{\delta_x \delta_y}[y, x, c] \\ J_{\delta_x \delta_y}[y, x, 4c + 1] &= I_{(1-\delta_x)\delta_y}[y, x, c] \\ J_{\delta_x \delta_y}[y, x, 4c + 2] &= I_{\delta_x (1-\delta_y)}[y, x, c] \\ J_{\delta_x \delta_y}[y, x, 4c + 3] &= I_{(1-\delta_x)(1-\delta_y)}[y, x, c] \end{aligned} . \tag{2}$$

These four images are merely channel-wise permutations of one another, which enables the network to keep balance between the four patterns.

Finally, each of these images is decomposed in blocks. Because all spatial and pixelwise features have already been computed in the previous parts, we can directly view the decomposition in blocks as one big average pooling, so that each block is spatially represented by one pixel. We thus get four output images B_{00}, B_{01}, B_{10} and B_{11}, each of shape $(\frac{Y}{16}, \frac{X}{16}, 4C)$. Each image is thus spatially 32×32 times smaller than the original image.

Thanks to this permutation, the detection problem is slightly shifted: Pixels in $J_{\delta_x \delta_y}$ are shifted so that all blocks of $B_{\delta_x \delta_y}$ should be detected at the same relative position modulo $(2, 2)$, $\delta_x \delta_y$. This process is explained in Fig. 5.

3.4. Blockwise Causal network

Because blocks are represented through average pooling, each block is spatially represented by one pixel. As a consequence, creating new pointwise convolutions amounts to processing the data independently – but with shared weights – in each block.

Furthermore, the four values $B_{\delta_x \delta_y}[y, x, 4c + i]$ for $i \in (0, 1, 2, 3)$ represent the same feature, averaged indepen-

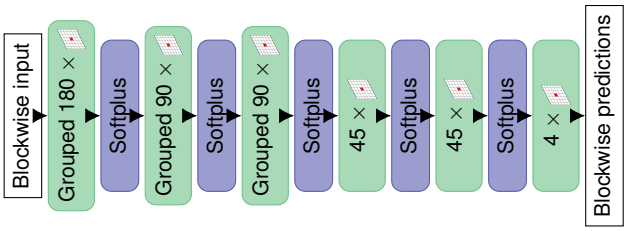


Figure 6: Blockwise part of the network, containing 8,239 trainable parameters

dently in each of the four possible mosaics $\delta_x\delta_y$. To compare these features separately before merging them, we start by stacking three layers of respectively 180, 90 and 90 grouped pixelwise convolution, where the output in one channel at one given block–position is made using only the values of the same feature, in the four mosaics and at the same position. Finally, we merge these features together with two additional layers, each of 45 full-depth pointwise convolutions. Like in the pixelwise network, the layers are separated by Softplus activation [18]. The structure of the blockwise causal network is shown in Fig. 6.

3.5. Decision and loss module

A final layer of four pointwise convolutions is placed to predict scores for each position. In an authentic image, all blocks from each image $B_{\delta_x\delta_y}$ would be expected to detect their own position as $\delta_x\delta_y$. If training on several images whose main mosaic may be different, we let the network permute the output of the four images either horizontally, vertically, or diagonally, in order to have the lowest of the four global losses before computing the local loss. This enables the loss to take into account the possibility of different images having different main positions.

3.6. Auxiliary prediction for training

Because the spatial and pixelwise networks are used at full resolution – whereas the resolution of images is reduced by a factor 32×32 in the blockwise network –, the first part of the network takes a higher computational toll than the rest. In order to speed up training, we work in a manner similar to [64] and start by training the spatial and pixelwise networks together. We add an additional layer of 4 pointwise convolutions at the end of the pixelwise network, and train it with the cross-entropy loss to detect the position of each pixel modulo $(2, 2)$.

Once the first part of the network is trained, we remove this auxiliary layer and process the output of the training images into blocks, as explained in 3.3. We then train the blockwise network, using the preprocessed output of the pixelwise network.

By training the first part of the network separately, and more importantly using a loss computed at full resolution, we can train it in fewer and faster iterations. Processing the images into blocks, which also requires a significant time, must only be applied once between the two global training steps. Finally, the blockwise part of the network can be trained very quickly, because there is no need to propagate into and from the full-resolution network at each iteration, making each individual iteration quicker.

Training is done first on the spatial (Fig. 3) and pixelwise (Fig. 4) networks, using the aforementioned auxiliary layer. Then, the blockwise network (Fig. 6) is trained alone, using the results of the pixelwise network, processed into blocks as seen in Fig. 5. All training is done with the Cross-Entropy loss and the Adam optimiser [39], with a learning rate of 10^{-3}.

4. Dataset

Several datasets exist to benchmark image forgery detection, most notably Coverage [69], CoMoFoD [65], Casia [17] and [13]. However, these datasets were created for generic copy-move detection. They do not allow for a demosaicing based detection. Indeed, the images of those datasets either do not present any trace of demosaicing, or were all demosaiced with the same algorithm. They are therefore useless for benchmarking CFA-based forgery detection algorithms.

The Dresden Image Database [28] provides 16,961 authentic images taken with 27 different cameras. Among them, 1,491 pictures taken with three different cameras, the Nikon D200, D70 and D70s, are provided unprocessed in a RAW format, which enabled us to perform demosaicing ourselves. Using these images, we created a new forgery detection database aimed specifically at the detection of forgeries by an analysis of CFA demosaicing inconsistencies.

To create the database, we cropped randomly each of the 1,491 images into smaller 648×648 pictures. We demosaiced them with one of eleven publicly available demosaicing algorithms: Bilinear interpolation, LMMSE [25], Hamilton-Adams [30], RI [37], MLRI [38], ARI [52], GBTF [55], contour stencils [26], adaptive inter-channel correlation [19], Gunturk [24] and self-similarity˜[9].

We then split the resulting set of images in three equal parts. One third of the images were left unmodified. In the second third, we took half of the images and used them to perform a splicing into the other half. Each pair of images had been previously demosaiced with the same algorithm. In the last third, we picked half the images again and used them to falsify the other half. However, we did not enforce pairs of images to be demosaiced with the same algorithm in this set. Note that the source images for the forgeries are not part of the resulting dataset; therefore, there is the same number of authentic and forged images. At least half the forged images were created with a source image demosaiced with the same algorithm as for the target.

Figure 7: Examples of forged images in our database.

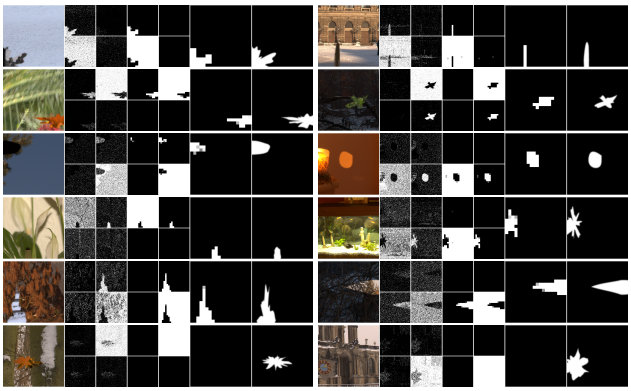


Figure 8: Network's results. For each image, in this order: Forged image, pixelwise predictions for each of the 4 grids (auxiliary network output), blockwise predictions for each of the 4 grids (full network output), detected forged blocks, ground truth. The mosaic of the image and the forgery is aligned for the two images in the last row, which explain why no detection can be made with our method.

To forge an image, we cropped the source image inside a random mask and pasted it onto the forged image. The masks were created as areas surrounded by random Bezier curves. They were enforced to contain at least one 64×64 square block, and to cover less than 10% of the image.

Examples of forged images found in our database can be seen in Fig. 7.

5. Experiments

We trained our network with a small database of 19 images, downsampled four times to remove any demosaicing trace. Each image has a size of at most 774×518 pixel, and was demosaiced by three different algorithms: bilinear interpolation, LMMSE [25] and Hamilton-Adams [30]. We trained the first part of the network for 1500 iterations and the second part for 500 iterations. Examples of detections can be seen in Fig. 8.

We also adapted the pretrained network to the database by retraining it directly on it for the 1000 iterations on the first part of the network and 500 on the second part. This training was done without knowledge of which image is forged or authentic.

We compare our results with intermediate value mosaic detection [11], variance of colour difference [60], as well as with ManTraNet [73], a state-of-the-art forgery detection method that directly trains a neural network to detect various forgeries on standard datasets. Results are measured with the ROC curve on the number of detected forged blocks.

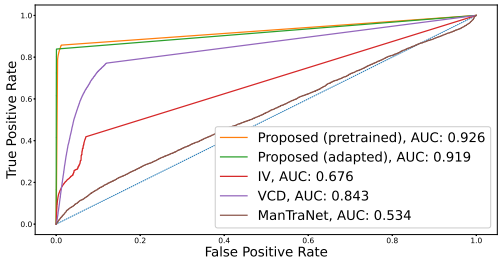
By nature of demosaicing, a region forged by copy-move has a $\frac{1}{4}$ probability of having its mosaic aligned with the main one, and in such case it cannot be detected by its CFA position. In our databases, aligned forgeries account for 26.7% of the total number of blocks. The results of our algorithms on the whole dataset is shown on Fig. 9b. Such results are closer to what could be detected in practical applications. However, because forgeries with aligned mosaic are not detectable by mosaic detection algorithms, we present other results with a modified ground truth, in which we consider a block as forged only if its mosaic is different than the position of the original image. These scores are thus given relative to what could theoretically be detected with perfect knowledge of the mosaics. Results under this definition of the ground truth can be seen in Fig. 9a.l The database features three algorithms that were also used for pretraining the network: bilinear interpolation, LMMSE [25] and Hamilton-Adams demosaicing [30]. In order to ensure fairness in the comparison, we remove all images demosaiced with, or containing a forged region demosaiced with, one of these three algorithms. The results are presented in Fig. 9c. We can see that the results are similar to those on the whole database, which shows that the network generalised well to new algorithms.

Finally, we test the robustness of our models to JPEG compression by compressing all the images at a quality of 95. The results are presented in Fig. 9d. [60] does no better than random guessing, with an AUC score of 0.52 in the global evaluation and 0.49 in the local evaluation, and both [11] and our pretrained network do little better. On the other hand, the adaptive network, by adapting directly to the database and thus learning to detect the CFA position over JPEG artefacts, was able to perform much better.

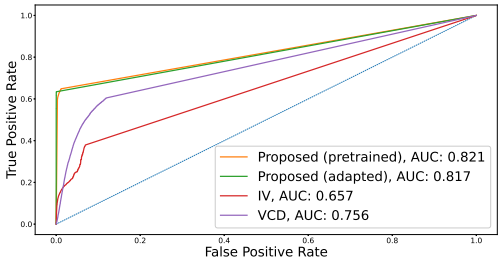
6. Discussion

We have shown that a small convolutional neural network could be used to accurately detect and interpret CFA artefacts, and subsequently use them to detect forgeries in images. Even without new training, this network can adapt well to images demosaiced with unseen and more complex algorithms than those studied during training. Our neural network is small and can process images almost as quickly as methods presented in the literature, while offering detections of superior quality.

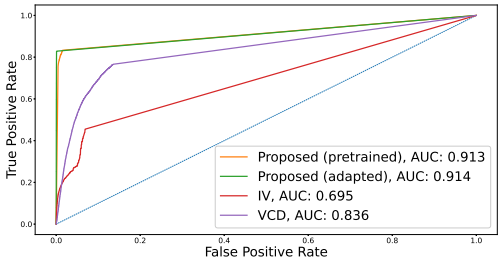
The forgeries in our database are very basic, since they are only made to evaluate CFA detection. Despite this, state-of-the-art generic methods such as ManTraNet yield detections that are little better than random, and worse than



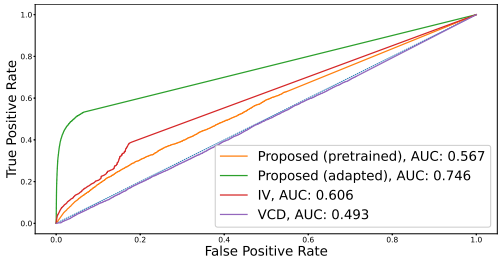
(a) Only misaligned forgeries are considered.



(b) All forgeries are considered.



(c) Only misaligned forgeries, algorithms on which our network was pretrained are excluded.



(d) Only misaligned forgeries, images are compressed at a JPEG quality of 95.

Figure 9: ROC curves comparing the detections of our methods to ManTraNet [73], Intermediate Values (IV) detection [11] and Variance of Colour Differance (VCD) detection [60].

simple manual algorithms such as [11, 60]. This shows that detection methods that focus on specific artefacts, such as demosaicing detection, JPEG compression [53] or camera noise [15], still have a big role to play.

Our network was trained on few images, which were not taken from the evaluation dataset, and with only three algorithms. This enabled us to show, in Fig. 9c, that we could get strong results even on images demosaiced with algorithms on which the network was not trained. In order to test and show its capacity to generalise to new images and algorithms, we only trained it with 19 images from a different dataset than the evaluation images, and three algorithms. A full instance of this network, trained with all known and available algorithm, would probably yield even better results.

Unfortunately, the pretrained model is not sufficient to process mosaic artefacts in compressed images. This is to be expected, as JPEG compression erases high frequencies even at a high quality, which is also where demosaicing algorithms leave artefacts. However, adapting the network to the new compressed data by retraining it directly on the studied data enabled it to retrieve demosaicing traces over JPEG compression.

We believe that our method shows sufficient results for use in demosaiced images without postprocessing. The next step is to consider common post-processing effects, including but not limited to JPEG compression, added noise, or colour change. Further work will study the robustness of the network to various post-processing setups, and try to improve adaptation of the network to post-processed images.

Adapting a pre-trained network to the testing data by retraining it on said data is of course something that must be done carefully. A network that is too big can easily overfit if too few samples are available, and see its quality lower in comparison to the pre-trained network. More experiments must thus be done to fully understand what can be done this way. Namely, two big questions arise: How much similar data is needed, and how similar does it need to be, in order to be able to improve the quality of a neural network by retraining it on this data? More importantly, can we prevent the network from overfitting when retrained on small amounts of data?

Since overfitting is the only reason why a network could worsen by trying to adapt to new data, it is likely that preventing or limiting it would make it possible to improve a network by retraining it on new data. We have shown that it was possible to drastically improve the performance of the network by retraining on the full database. Would it still be possible if we only have several images, or, in the most extreme but also the most frequent case, only one image? Preliminary experiments suggest that it would be possible providing the image is big enough, but more work is necessary to determine this.

This is a difficult challenge, however learning to make a network fit to new unlabelled data could greatly help make it more robust for practical applications.

Work funded by the French Ministère des armées – Direction Générale de l'Armement, and by grant ANR-16-DEFA-0004 Signature d'Images – ANR/DGA DEFALS challenge.
Centre Borelli is also a member of Université Paris Cité, SSA and

INSERM.

References

[1] Irene Amerini, Lamberto Ballan, Roberto Caldelli, Alberto Del Bimbo, and Giuseppe Serra. A sift-based forensic method for copy–move attack detection and transformation recovery. *IEEE Transactions on Information Forensics and Security*, 6(3):1099–1110, 2011. 1

[2] Q. Bammey, R. Grompone von Gioi, and J. Morel. Automatic detection of demosaicing image artifacts and its use in tampering detection. In *2018 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, pages 424–429, April 2018. 2, 3

[3] Jawadul H Bappy, Amit K Roy-Chowdhury, Jason Bunk, Lakshmanan Nataraj, and BS Manjunath. Exploiting spatial structure for localizing manipulated image regions. In *Proceedings of the IEEE international conference on computer vision*, pages 4970–4979, 2017. 1

[4] Jawadul H Bappy, Cody Simons, Lakshmanan Nataraj, BS Manjunath, and Amit K Roy-Chowdhury. Hybrid lstm and encoder-decoder architecture for detection of image forgeries. *IEEE Transactions on Image Processing*, 2019. 1

[5] Belhassen Bayar and Matthew C Stamm. Constrained convolutional neural networks: A new approach towards general purpose image manipulation detection. *IEEE Transactions on Information Forensics and Security*, 13(11):2691–2706, 2018. 1

[6] Tiziano Bianchi, Alessia De Rosa, and Alessandro Piva. Improved dct coefficient analysis for forgery localization in jpeg images. In *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2444–2447. IEEE, 2011. 1

[7] Tiziano Bianchi and Alessandro Piva. Image forgery localization via block-grained analysis of jpeg artifacts. *IEEE Transactions on Information Forensics and Security*, 7(3):1003–1017, 2012. 1

[8] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. Signature verification using a" siamese" time delay neural network. In *Advances in neural information processing systems*, pages 737–744, 1994. 3

[9] A. Buades, B. Coll, J. M. Morel, and C. Sbert. Self-similarity Driven Demosaicking. *Image Processing On Line*, 1:51–56, 2011. 6

[10] Jason Bunk, Jawadul H Bappy, Tajuddin Manhar Mohammed, Lakshmanan Nataraj, Arjuna Flenner, BS Manjunath, Shivkumar Chandrasekaran, Amit K Roy-Chowdhury, and Lawrence Peterson. Detection and localization of image forgeries using resampling features and deep learning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 1881–1889. IEEE, 2017. 1

[11] Chang-Hee Choi, Jung-Ho Choi, and Heung-Kyu Lee. Cfa pattern identification of digital cameras using intermediate value counting. In *Proceedings of the Thirteenth ACM Multimedia Workshop on Multimedia and Security*, MM&Sec '11, pages 21–26, New York, NY, USA, 2011. ACM. 3, 7, 8

[12] F. Chollet. Xception: Deep learning with depthwise separable convolutions. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1800–1807, July 2017. 5

[13] V. Christlein, C. Riess, J. Jordan, C. Riess, and E. Angelopoulou. An evaluation of popular copy-move forgery detection approaches. *IEEE Transactions on Information Forensics and Security*, 7(6):1841–1854, Dec 2012. 6

[14] Davide Cozzolino, Giovanni Poggi, and Luisa Verdoliva. Splicebuster: A new blind image splicing detector. In *2015 IEEE International Workshop on Information Forensics and Security (WIFS)*, pages 1–6. IEEE, 2015. 1

[15] Davide Cozzolino and Luisa Verdoliva. Noiseprint: a cnn-based camera model fingerprint. *IEEE Transactions on Information Forensics and Security*, 2019. 8

[16] Christophe Destruel, Vincent Itier, Olivier Strauss, and William Puech. Color noise-based feature for splicing detection and localization. In *2018 IEEE 20th International Workshop on Multimedia Signal Processing (MMSP)*, pages 1–6. IEEE, 2018. 1

[17] Jing Dong, Wei Wang, and Tieniu Tan. Casia image tampering detection evaluation database. pages 422–426, 07 2013. 6

[18] Charles Dugas, Y. Bengio, François Bélisle, Claude Nadeau, and Rene Garcia. Incorporating second-order functional knowledge for better option pricing. pages 472–478, 01 2000. 5, 6

[19] J. Duran and A. Buades. Self-similarity and spectral correlation adaptive algorithm for color demosaicking. *IEEE TIP*, 23(9):4031–4040, Sept 2014. 6

[20] Wei Fan, Kai Wang, François Cayre, and Zhang Xiong. A variational approach to jpeg anti-forensics. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 3058–3062. IEEE, 2013. 1

[21] Hany Farid. Exposing digital forgeries from jpeg ghosts. *IEEE transactions on information forensics and security*, 4(1):154–160, 2009. 1

[22] Hany Farid. *Photo Forensics*. The MIT Press, 2016. 1

[23] Anselmo Ferreira, Siovani C Felipussi, Carlos Alfaro, Pablo Fonseca, John E Vargas-Muñoz, Jefersson A dos Santos, and Anderson Rocha. Behavior knowledge space-based fusion for copy–move forgery detection. *IEEE Transactions on Image Processing*, 25(10):4729–4742, 2016. 1

[24] Pascal Getreuer. Gunturk-altunbasak-mersereau alternating projections image demosaicking. *Image Processing on Line*, 1:90–97, 2011. 6

[25] Pascal Getreuer. Zhang-wu directional lmmse image demosaicking. *Image Processing On Line*, 1:117–126, 2011. 4, 6, 7

[26] P. Getreuer. Image Demosaicking with Contour Stencils. *IPOL*, 2:22–34, 2012. 6

[27] Aurobrata Ghosh, Zheng Zhong, Terrance E Boult, and Maneesh Singh. Spliceradar: A learned method for blind image forensics. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019. 1

[28] Thomas Gloe and Rainer Böhme. The 'Dresden Image Database' for benchmarking digital image forensics. In *Proceedings of the 25th Symposium On Applied Computing (ACM SAC 2010)*, volume 2, pages 1585–1591, 2010. 2, 3, 6

[29] Edgar González Fernández, Ana Sandoval Orozco, Luis García Villalba, and Julio Hernandez-Castro. Digital image tamper detection technique based on spectrum analysis of cfa artifacts. *Sensors*, 18(9):2804, Aug 2018. 2, 3

[30] John F Hamilton Jr and James E Adams Jr. Adaptive color plan interpolation in single sensor color electronic camera, May 13 1997. US Patent 5,629,734. 6, 7

[31] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *CoRR*, abs/1704.04861, 2017. 5

[32] Minyoung Huh, Andrew Liu, Andrew Owens, and Alexei A. Efros. Fighting fake news: Image splice detection via learned self-consistency. In *The European Conference on Computer Vision (ECCV)*, September 2018. 3

[33] Chryssanthi Iakovidou, Markos Zampoglou, Symeon Papadopoulos, and Yiannis Kompatsiaris. Content-aware detection of jpeg grid inconsistencies for intuitive image forensics. *Journal of Visual Communication and Image Representation*, 54:155–170, 2018. 1

[34] Daniel Cavalcanti Jeronymo, Yuri Cassio Campbell Borges, and Leandro dos Santos Coelho. Image forgery detection by semi-automatic wavelet soft-thresholding with error level analysis. *Expert Systems with Applications*, 85:348–356, 2017. 1

[35] Thibaut Julliand, Vincent Nozick, and Hugues Talbot. Automated image splicing detection from noise estimation in raw images. 2015. 1

[36] Yongzhen Ke, Qiang Zhang, Weidong Min, and Shuguang Zhang. Detecting image forgery based on noise estimation. *International Journal of Multimedia and Ubiquitous Engineering*, 9(1):325–336, 2014. 1

[37] Daisuke Kiku, Yusuke Monno, Masayuki Tanaka, and Masatoshi Okutomi. Residual interpolation for color image demosaicking. In *2013 IEEE International Conference on Image Processing*, pages 2304–2308. IEEE, 2013. 6

[38] Daisuke Kiku, Yusuke Monno, Masayuki Tanaka, and Masatoshi Okutomi. Minimized-laplacian residual interpolation for color image demosaicking. In *Digital Photography X*, volume 9023, page 90230L. International Society for Optics and Photonics, 2014. 6

[39] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 6

[40] Matthias Kirchner. Efficient estimation of CFA pattern configuration in digital camera images. In *Media Forensics and Security*, 2010. 3

[41] Paweł Korus. Digital image integrity–a survey of protection and verification techniques. *Digital Signal Processing*, 71:1–26, 2017. 1

[42] Weihai Li, Yuan Yuan, and Nenghai Yu. Passive detection of doctored jpeg image via block artifact grid extraction. *Signal Processing*, 89(9):1821–1829, 2009. 1

[43] Zhouchen Lin, Junfeng He, Xiaoou Tang, and Chi-Keung Tang. Fast, automatic and fine-grained tampered jpeg image detection via dct coefficient analysis. *Pattern Recognition*, 42(11):2492–2501, 2009. 1

[44] Bo Liu and Chi-Man Pun. Splicing forgery exposure in digital image by detecting noise discrepancies. *International Journal of Computer and Communication Engineering*, 4(1):33, 2015. 1

[45] Lu Liu, Yao Zhao, Rongrong Ni, and Qi Tian. Copy-move forgery localization using convolutional neural networks and cfa features. *Int. J. Digit. Crime For.*, 10(4):140–155, Oct. 2018. 3

[46] Andrew L. Maas. Rectifier nonlinearities improve neural network acoustic models. 2013. 4

[47] Babak Mahdian and Stanislav Saic. Using noise inconsistencies for blind image forensics. *Image and Vision Computing*, 27(10):1497–1503, 2009. 1

[48] Owen Mayer, Belhassen Bayar, and Matthew C Stamm. Learning unified deep-features for multiple forensic tasks. In *Proceedings of the 6th ACM Workshop on Information Hiding and Multimedia Security*, pages 79–84. ACM, 2018. 1

[49] Owen Mayer and Matthew C Stamm. Learned forensic source similarity for unknown camera models. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2012–2016. IEEE, 2018. 1

[50] O. Mayer and M. C. Stamm. Learned forensic source similarity for unknown camera models. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 2012–2016, April 2018. 3

[51] O. Mayer and M. C. Stamm. Forensic similarity for digital images. *IEEE Transactions on Information Forensics and Security*, 2019. 1

[52] Yusuke Monno, Daisuke Kiku, Masayuki Tanaka, and Masatoshi Okutomi. Adaptive residual interpolation for color image demosaicking. In *2015 IEEE International Conference on Image Processing (ICIP)*, pages 3861–3865. IEEE, 2015. 6

[53] T. Nikoukhah, J. Anger, T. Ehret, M. Colom, J.M. Morel, and R. Grompone von Gioi. Jpeg grid detection based on the number of dct zeros and its application to automatic and localized forgery detection. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, June 2019. 8

[54] Xunyu Pan, Xing Zhang, and Siwei Lyu. Exposing image splicing with inconsistent local noise variances. In *2012 IEEE International Conference on Computational Photography (ICCP)*, pages 1–10. IEEE, 2012. 1

[55] Ibrahim Pekkucuksen and Yucel Altunbasak. Gradient based threshold free color filter array interpolation. In *2010 IEEE International Conference on Image Processing*, pages 137–140. IEEE, 2010. 6

[56] Tomáš Pevny and Jessica Fridrich. Detection of double-compression in jpeg images for applications in steganography. *Information Forensics and Security, IEEE Transactions on*, 3(2):247–258, 2008. 1

[57] A.C. Popescu and H. Farid. Exposing digital forgeries in color filter array interpolated images. *Trans. Sig. Proc.*, 53(10):3948–3959, Oct. 2005. 2, 3

[58] Alin C Popescu and Hany Farid. Statistical tools for digital forensics. In *Information Hiding*, pages 128–147. Springer, 2004. 1

[59] M Ali Qureshi and M Deriche. A review on copy move image forgery detection techniques. In *2014 IEEE 11th International Multi-Conference on Systems, Signals & Devices (SSD14)*, pages 1–5. IEEE, 2014. 1

[60] Hyun Jun Shin, Jong Ju Jeon, and Il Kyu Eom. Color filter array pattern identification using variance of color difference image. *Journal of Electronic Imaging*, 26(4):1 – 12, 2017. 3, 7, 8

[61] Ewerton Silva, Tiago Carvalho, Anselmo Ferreira, and Anderson Rocha. Going deeper into copy-move forgery detection: Exploring image telltales via multi-scale analysis and voting processes. *Journal of Visual Communication and Image Representation*, 29(0):16 – 32, 2015. 1

[62] Matthew C Stamm, Steven K Tjoa, W Sabrina Lin, and KJ Liu. Undetectable image tampering through jpeg compression anti-forensics. In *Image Processing (ICIP), 2010 17th IEEE International Conference on*, pages 2109–2112. IEEE, 2010. 1

[63] Matthew Christopher Stamm, Steven K Tjoa, W Sabrina Lin, and KJ Ray Liu. Anti-forensics of jpeg compression. In *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, pages 1694–1697. IEEE, 2010. 1

[64] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015. 2, 5, 6

[65] D. Tralic, I. Zupancic, S. Grgic, and M. Grgic. Comofod — new database for copy-move forgery detection. In *Proceedings ELMAR-2013*, pages 49–54, Sep. 2013. 6

[66] Giuseppe Valenzise, Marco Tagliasacchi, and Stefano Tubaro. Revealing the traces of jpeg compression anti-forensics. *Information Forensics and Security, IEEE Transactions on*, 8(2):335–349, 2013. 1

[67] Savita Walia and Mandeep Kaur. Forgery detection using noise inconsistency: A review. *International Journal of Computer Science and Information Technologies*, 5(6):7618–7622, 2014. 1

[68] Nathalie Diane Wandji, Sun Xingming, and Moise Fah Kue. Detection of copy-move forgery in digital images based on dct. *arXiv preprint arXiv:1308.5661*, 2013. 1

[69] Bihan Wen, Ye Zhu, Ramanathan Subramanian, Tian-Tsong Ng, Xuanjing Shen, and Stefan Winkler. Coverage – a novel database for copy-move forgery detection. In *IEEE International Conference on Image processing (ICIP)*, pages 161–165, 2016. 6

[70] Bihan Wen, Ye Zhu, Ramanathan Subramanian, Tian-Tsong Ng, Xuanjing Shen, and Stefan Winkler. Coverage—a novel database for copy-move forgery detection. In *2016 IEEE International Conference on Image Processing (ICIP)*, pages 161–165. IEEE, 2016. 1

[71] Yue Wu, Wael Abd-Almageed, and Prem Natarajan. Busternet: Detecting copy-move image forgery with source/target localization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 168–184, 2018. 1

[72] Yue Wu, Wael Abd-Almageed, and Prem Natarajan. Mantra-net: Manipulation tracing network for detection and localization of image forgerieswith anomalous features. 2019. 1

[73] Yue Wu, Wael AbdAlmageed, and Premkumar Natarajan. Mantra-net: Manipulation tracing network for detection and localization of image forgeries with anomalous features. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019. 7, 8

[74] Heng Yao, Shuozhong Wang, Xinpeng Zhang, Chuan Qin, and Jinwei Wang. Detecting image splicing based on noise level inconsistency. *Multimedia Tools and Applications*, 76(10):12457–12479, 2017. 1

[75] Markos Zampoglou, Symeon Papadopoulos, Yiannis Kompatsiaris, Ruben Bouwmeester, and Jochen Spangenberg. Web and social media image forensics for news professionals. In *SMN@ ICWSM*, 2016. 1

[76] Hui Zeng, Yifeng Zhan, Xiangui Kang, and Xiaodan Lin. Image splicing localization using pca-based noise level estimation. *Multimedia Tools and Applications*, 76(4):4783–4799, 2017. 1