A FORENSIC REPRESENTATION TO DETECT NON-TRIVIAL IMAGE DUPLICATES, AND HOW IT APPLIES TO SEMANTIC SEGMENTATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Manipulation and re-use of images in scientific publications is a recurring problem, at present lacking a scalable solution. Existing tools for detecting image duplication are mostly manual or semi-automated, despite the fact that generating data for a learning-based approach is straightforward, as we here illustrate. This paper addresses the problem of determining if, given two images, one is a manipulated version of the other by means of certain geometric and statistical manipulations, e.g. copy, rotation, translation, scale, perspective transform, histogram adjustment, partial erasing, and compression artifacts. We propose a solution based on a 3-branch Siamese Convolutional Neural Network. The ConvNet model is trained to map images into a 128-dimensional space, where the Euclidean distance between duplicate (respectively, unique) images is no greater (respectively, greater) than 1. Our results suggest that such an approach can serve as tool to improve surveillance of the published and in-peer-review literature for image manipulation. We also show that as a byproduct the network learns useful representations for semantic segmentation, with performance comparable to that of domain-specific models.

1 INTRODUCTION

Duplicative data reporting in the biomedical literature is more prevalent than most people realize (Bik et al. (2016)). One common form of data duplication, regardless of intent, is the re-use of scientific images, across multiple publications or even within the same publication. In some cases, images are altered before being re-used (Bik et al. (2016)). Changing orientation, perspective or image statistics, introducing skew or crop, and deleting or inserting data into the original image plane are all ways in which image data may be altered prior to inappropriate introduction, or re-introduction, into the reporting of experimental outcomes (Rossner & Yamada (2004); Cromey (2009); Blatt & Martin (2013)). While the scientific community has affirmatively recognized the need for preventing the incorporation of duplicative or flawed image data into the scientific record, a consistent approach to screening and identifying problematic image data has yet to be established (Rossner (2006; 2008)).

Cases of image data duplication and/or manipulation have often been detected by fellow scientists[1] or by editorial staff during the manuscript review process. Efforts to move towards automation include tools developed to isolate regions of manipulation within images already flagged as suspicious (Koppers et al. (2017)). However, current methods for identifying duplicative and/or manipulated images largely rely on individual visual identification with accompanying application of qualitative similarity measures[2]. Given the rate at which the scientific literature is expanding, it is not feasible for all cases of potential image manipulation to be detected by human eyes. Thus, there is a continued need for automated tools to detect potential duplications, even in the presence of manipulation, to allow for more focused, thorough evaluation of this smaller errant image candidate pool. Such a tool would be invaluable to scientists and research staff on many levels, from figure screening as a step in improving raw data maintenance and manuscript preparation at the laboratory level (Rossner

[1] E.g.: http://retractionwatch.com/,https://pubpeer.com/, https://en.wikipedia.org/wiki/Clare_Francis_(science_critic)

[2] E.g.: https://ori.hhs.gov/forensic-tools

& Yamada (2004)), to the routine screening by journal editorial staff of submitted manuscripts prior to the peer-review process (Rossner (2006); Gilbert (2009)).

The general problem of detecting similar images has been well studied in the field of computer vision (e.g. Wang & Simoncelli (2005); Wang et al. (2014); Veit et al. (2017)). The one application that stands out is determining if two given faces are of the same person, where recent breakthroughs in deep Convolutional Neural Networks have allowed rapid progress (Schroff et al. (2015)).

In this paper, we apply modern methods in metric learning to address the problem of detecting image manipulation and re-use in scientific work. Specifically, we train a ConvNet to learn an image embedding such that images with the same original content, albeit altered through a common set of image manipulations, appear close to each other in the embedding space. We train this model on a large corpus of simulated image manipulations, and test on a small set of manipulated images from known instances of image duplication/manipulation[3]. To our knowledge, this is the first application of deep learning to the detection of image *re-use* in the scientific literature, although there have been works on the area of detecting image manipulation (e.g. Bayar & Stamm (2016)).

We focus on the domain of biological images, since we have easy access to one such dataset, but naturally the model to be described is agnostic to the image domain. We test the learned forensic representation not only on new/unseen synthetic and real data for the problem of duplicate-detection, but also on a somewhat unrelated area: semantic segmentation. We show that the features learned in the convolution layers of the siamese network can be readily plugged into a pixel classifier, yielding results comparable with those of state-of-the art, domain-specific architectures.

2 Related Work

This work is primarily based on Chopra et al. (2005), Schroff et al. (2015), and Koch et al. (2015).

The classic model for image similarity was proposed in Chopra et al. (2005) in the context of face verification: a *siamese* neural network. This network has two *branches* that share parameters (weights) during training. Each branch is composed of layers of convolutions and non-linearities followed by fully connected layers. The two branches are connected at the bottom by the L_1 norm. During training, pairs of images known to be similar or dissimilar are fed to the network, and the loss function is designed to encourage the network to learn a representation that makes the L_1 distance between the two representations small or large, respectively.

In Schroff et al. (2015), the authors improved upon the standard siamese network model by adding one extra branch, thus training on image triplets instead of pairs. A triplet consists of an anchor, a positive example ("same" or "similar" to the anchor image), and a negative example ("different" from the anchor). A triplet loss was designed to drive similar images to be nearby, and dissimilar images to be far apart, encouraging the embedding space to be locally Euclidean. A clever trick that enables fast convergence is the use of hard negative mining: selecting examples where "different" images are close according to the current metric, and "similar" images are far apart.

In Koch et al. (2015), the authors kept the 2-branch architecture, but used a non-conventional "metric" (possibly assuming negative values) at the connection of the two branches, with a cross-entropy loss function. This approach allows for the model to learn a function that gives a binary output, rather than a distance between images, which has the advantage of not requiring the user to establish a threshold of proximity for images to be the same, as required in Chopra et al. (2005) and Schroff et al. (2015).

For our application, we found the binary output option to be more interesting from a user's perspective, since the threshold for "sameness" can be difficult to set properly. However, experiments with the loss function proposed in Koch et al. (2015) led us to abandon it due to its instability for images that are actually the same. We settled with a modification that enforces a threshold of 1, beyond which images are considered different, and let the network learn the appropriate scaling required for the metric to comply with such separation. We borrow the triplet loss strategy from Schroff et al. (2015), for faster training.

[3]The test images were previously described as problematic and either corrected or retracted from the literature. Sourced from http://retractionwatch.com/ and/or https://pubpeer.com/

3 MODEL

We aim to solve the following problem: given two images, I and J, determine if they are the same or different, where J is considered to be the same as I if it is a manipulated version of I. We sought to find a solution to this problem in the form of a function f, an *image-forensic metric*, that computes a distance between two images, satisfying:

- $f(I, J) \geq 0$;
- $f(I, J) \leq 1$ when J is a manipulated version of I;
- $f(I, J) > 1$ when I and J are different images.

3.1 ARCHITECTURE

We use a *triplet* network architecture (Schroff et al. (2015)), with the 3 branches sharing parameters. Each branch consists of 4 convolution layers, each with ReLU non-linearity, followed by 2 fully-connected layers. We also included a few standard tricks-of-the-trade, such as batch normalization (Ioffe & Szegedy (2015)) and residual learning (He et al. (2016)). The resulting image representation C^i is a vector of dimension 128. A summary of the model is shown in the left panel of Figure 1 (a). We experimented with a considerable number of variations on network depth and hyper-parameters, though we did not perform a thorough or automated search for the optimal architecture.

3.2 TRIPLET LOSS

Let $C^i(I)$ be the representation at the bottom of branch i for image I, $i = 0, 1, 2$. Our forensic metric is defined as

$$f^{ij}(I, J) = \sum_k \alpha_k |C^i_k(I) - C^j_k(J)| \,, \tag{1}$$

where α_k are parameters to be learned. Now, with the convention that the anchor images feed through branch 0, "same" through branch 1, and "different" through branch 2, we define

$$\sigma_s(I, J) = \sigma(1 - f^{01}(I, J)) \,, \tag{2}$$
$$\sigma_d(I, J) = \sigma(1 - f^{12}(I, J)) \,, \tag{3}$$

where $\sigma()$ is the sigmoid function. Our triplet loss is then

$$L(B) = - \sum_{(A,S,D)\in B} [\log(\sigma_s(A, S)) + \log(1 - \sigma_d(A, D))] \,, \tag{4}$$

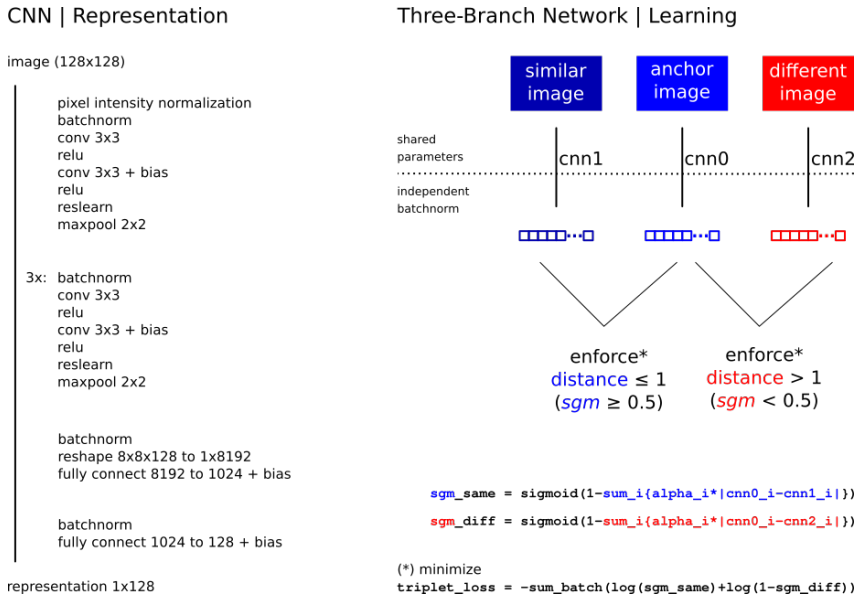
where B is a batch of triplets (A, S, D), i.e., anchor, same, different.

This loss forces $\sigma_s \geq \frac{1}{2}$ (thus $f^{01} \leq 1$), and $\sigma_d < \frac{1}{2}$ (thus $f^{12} > 1$), therefore imposing a virtual threshold of 1 as criteria for similarity as measured by f^{ij}.

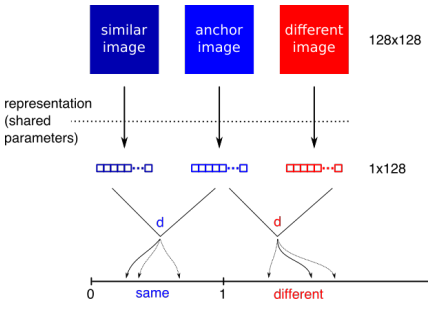
3.3 TRAINING PROCEDURE

Positive examples of image manipulation corresponding to data confirmed by institutional or regulatory bodies as problematic, which may include retracted and or corrected data, are not publicly available at the scale that would be required to train a high capacity ConvNet. Thus, we approached this problem by simulating examples of image manipulation to generate a large training set, and testing on a small set of real-world examples of inappropriately duplicated images in peer-reviewed publications[4]).

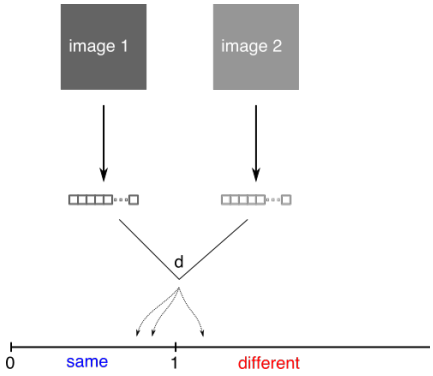
[4]These were obtained through PubPeer (`https://pubpeer.com`) and/or Retraction Watch (`http://retractionwatch.com`).



(a)



(b)



(c)

Figure 1: (a) Model details. (b) Model training diagram. (c) Model testing diagram.

To guide the generation of simulated data, we first evaluated the most common forms of image manipulation within our test set. We identified the following operations: identity, rotation, translation, scale, or perspective transform; local or global histogram adjustment; partial erasing; jpeg-like compression artifacts. In addition, we also accounted for operations that are common in the preparation of images for scientific manuscripts, such as the insertion of text or drawings. Examples of some of these deformations are shown in Figure 2.

We started by gathering data mainly from our own dataset of biomedical images, but also from public sources such as the NYU Mouse Embryo Tracking Database[5] and the Broad Bioimage Benchmark Collection[6]. There are about 20 classes of images, from various cell types and model organisms, including C. elegans, adipocytes, mouse nuclei, and mouse embryo cells. The data was cropped (with no overlap) in patches of 256×256 pixels, totaling 5215 images that were used to train the

[5] http://celltracking.bio.nyu.edu/

[6] https://data.broadinstitute.org/bbbc/

method. A separate set of images from 10 different biological subjects was created for testing, as detailed in Section 4.

At each training step two distinct batches of n images are sampled from the entire training set. The first batch is reserved for the "anchor" branch of the 3-branch siamese net, and the second for the "different" branch. For each anchor image, a corresponding image for the "similar" branch is obtained by on-the-fly deformation of the anchor. Deformations vary in degree (how much) and number (how many), according to the following pseudo-code, where `rand()` is a sample from the uniform distribution in $[0, 1]$, `randreflection()` is random reflection, `randpptf()` a random perspective transform, `randtform()` a random similarity transform (rotation, scale, translation), `crop()` is a 128×128 centered crop, `randjpegcompress()` is a jpeg compression with random loss, `randgammaadj()` is a random gamma adjustment, and `randlocaledit()` is a random local edit (change in pixel intensity).

```
deformation(im):
    r = rand()
    if r < 0.9:
        im1 = randreflection(im)     if rand() < 0.5 else im
        im2 = randpptf(im1)          if rand() < 0.5 else im1
        im3 = randtform(im2)         if rand() < 0.5 else im2
    else:
        im3 = im
    im4 = crop(im3)
    if r < 0.9:
        im5 = randjpegcompress(im4) if rand() < 0.5 else im4
        im6 = randgammaadj(im5)     if rand() < 0.5 else im5
        im7 = randlocaledit(im6)    if rand() < 0.5 else im6
    else:
        im7 = im4
    return im7
```

The "anchor" and "different" images on the triplet are also center-cropped to 128×128 to be of the same size as the "different" image, which needs cropping to eliminate border effects introduced by the deformations. Random clutter is added (with certain probability) to all images in the triplet – it can be either random text or a random rectangle. Precise parameters for these deformations will be available upon release of the source code. Some examples of deformations are shown in Figure 2.

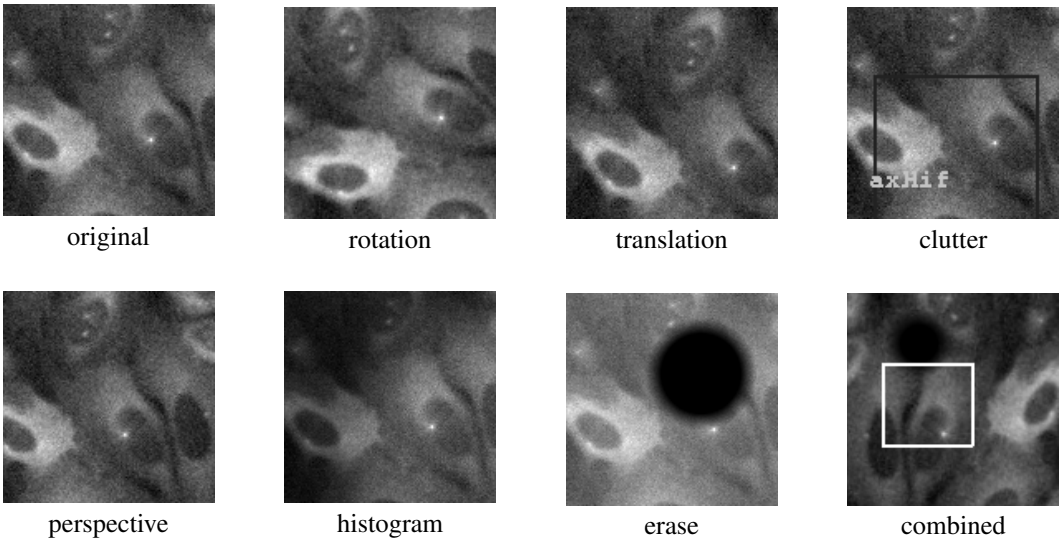


Figure 2: Deformations. "clutter" corresponds to the addition of random text and a random box; "histogram" corresponds to local and global pixel intensity adjustment; "combined" corresponds to a sample run of the algorithm described in Subsection 3.3.

4 RESULTS AND APPLICATIONS

For a batch size of 128, the model trains in about 20k steps with the momentum optimizer (momentum 0.9, learning rate 10^{-5}). Accuracy on the validation set flattens at around 0.96.

Our test set of real manipulations is composed of 108 classes of duplicates. Each class contains two or more manipulations of the same image. To evaluate performance, for each class of duplicates we pick randomly a fixed image from the class, another random image from the same class, and one random image from another (random) class. We then measure the average of correct predictions after running this procedure 10 times through all classes.

Our synthetic test set has the same structure, except more classes (402), and exactly 10 replicates per class. For a given class, 9 deformations are created (using the random algorithm described above) from a single fixed image. None of these images were present in the training set.

Accuracy on these two sets is shown in the following table, where SN is the siamese model as described above, and SN[G] is a similar one where only geometric deformations (e.g. rotation, translation, etc) are applied during training (i.e. no pixel intensity deformations such as gamma adjustment). SN[G] trains much faster in terms of time, since it demands less on-the-fly manipulations.

Model	Synthetic Test Set	Real-World Test Data
SN	0.853731	0.861574
SN[G]	0.748010	0.848148

As expected, SN[G] performs worse than SN on the synthetic set (because the synthetic set has more deformations), though more or less the same on the real-world test set, perhaps indicating the simulated pixel intensity variations do not reflect well those seen on real-world data.

Figure 3 contains additional analysis of the network's predictions on these sets in terms of σ (equation 2,3).

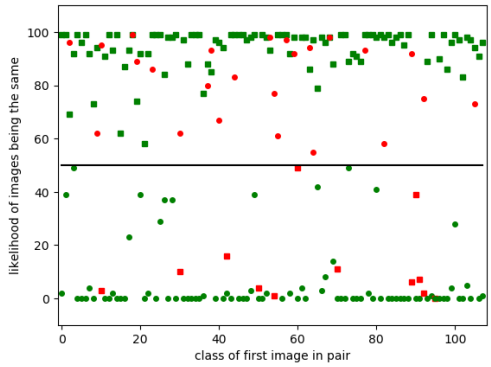
Unfortunately at this time we are unable to publish real-world example images due to copyright issues. Some examples of synthetic images are shown in Figure 3. Implementation (in TensorFlow) and synthetic datasets shall be made publicly available soon.

SEGMENTATION

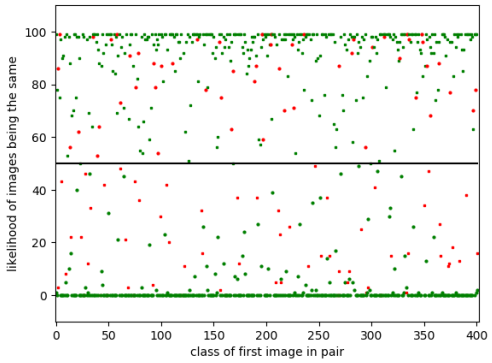
The 128-dimensional vector at the bottom of each branch of the siamese network is a natural representation for image clustering, search, or classification, which has been demonstrated elsewhere in the literature (e.g. in Koch et al. (2015) and Schroff et al. (2015)). Here we look into the features of the convolution layers, and how they can be used for image segmentation. To do so, we upsample and concatenate the feature maps in a way similar to what's done in the popular U-Net architecture (Ronneberger et al. (2015)), and plug the resulting tensor into a random forest classifier. This is illustrated in Figure 4. We then compare the performance of this hybrid classifier with the U-Net itself in two different background/foreground segmentation tasks. One dataset contains 83 training and 21 test images of size 512x512, where the goal is to segment a certain cell type. The other contains 74 training and 19 test images of size 360x360, where the goal is segment nuclei.

Our hybrid siamese net + random forest model (SN+RF) consists of upsampling and concatenating the feature maps of the last 3 convolution layers, implying each pixel has a feature vector of dimension 128+64+32. This model is trained on the original dataset – without augmentation. The U-Net for the cell segmentation problem has 4 downsampling layers, whereas the one for the nuclei segmentation task has 3 (due to smaller image sizes). To train these networks, we augment the training datasets by a factor of 40, with rotations, reflections, and thin-plate spline distortions. The following table summarizes the IoU (intersection over union) and PA (Pixel Accuracy, or portion of correctly classified pixels) results.

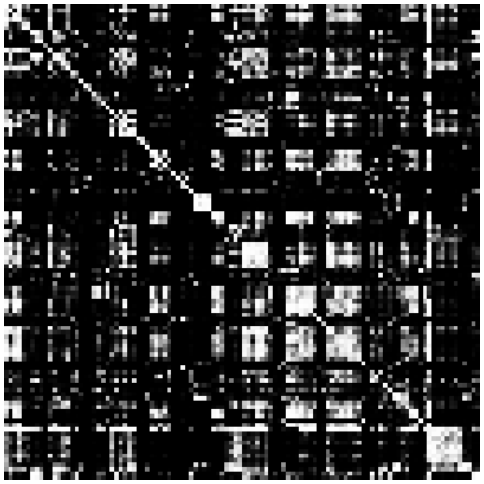
	Cell Seg.		Nuc. Seg.	
Model	IoU	PA	IoU	PA
U-Net	0.41136141	0.96844864	0.61621280	0.88191967
SN+RF	0.41355777	0.96729606	0.61537368	0.88659722
SN[G]+RF	0.40389093	0.96607463	0.61940216	0.88730750



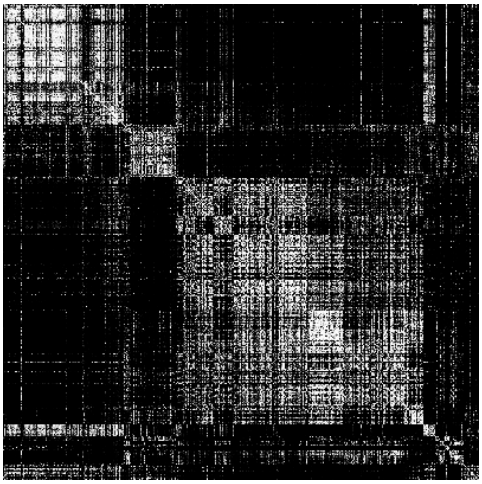
(a)



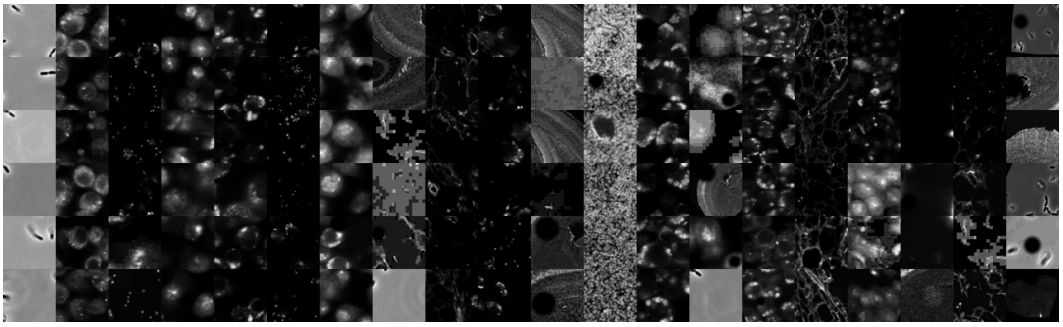
(b)



(c)



(d)



(e)

Figure 3: **(a, recpectively b)** $100 \cdot \sigma$ (equation 2,3) for one run of tests on the real-world (resp. synthetic) test set, which can be interpreted as the likelihood of a pair of images being the same. Squares correspond to pairs of "same" images, circles to "different" – thus all squares (resp. circles) should be above (resp. below) the horizontal line of likelihood equal to 50 (those that are, are colored green, those that are not, are colored red). **(c, respectively d)** similarity between classes of duplicates, as measured (via σ) for a pair of images taken one from each different class, for the real-world (resp. synthetic) test set. The large checkerboard patterns in the matrices (specially in (d)) are a reflection of the fact that the datasets are organized in a way where classes from the same category of images are indexed nearby. **(e)** For each fixed image on the top row, the images below it in the same column are their 5 nearest neighbors, the top-most being the closest. The 20 columns were randomly selected from the set of 402 classes in the synthetic test set.

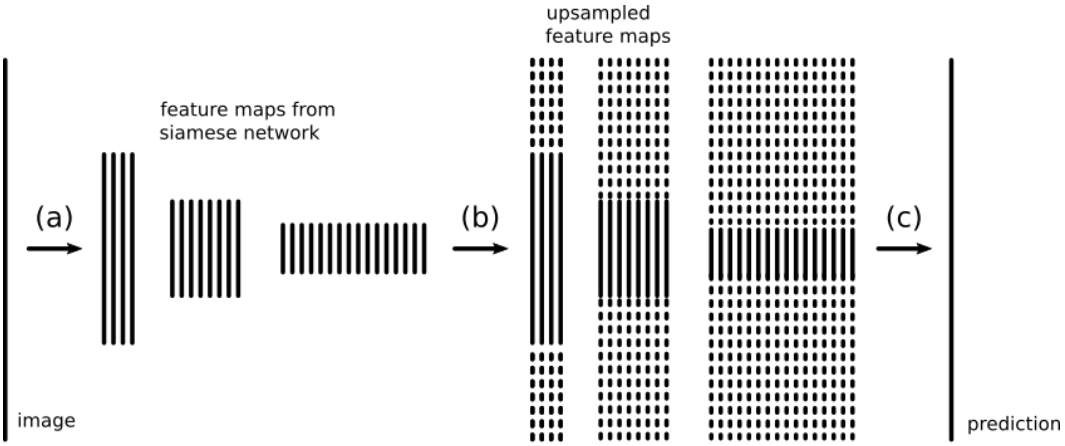


Figure 4: Usage of the representation learned by the siamese network for semantic segmentation. (a) The siamese net generates feature maps in different scales. (b) Each feature map is upsampled independently to match the size of the original image. (c) A random forest algorithm makes a per-pixel prediction using the upsampled features. This hybrid model resembles the U-Net architecture.

The last line corresponds to a siamese net trained with only geometric deformations (e.g. rotations, translations, etc) and no pixel-intensity alterations (e.g. gamma, local erase, etc). This model is much faster to train (less on-the-fly deformations to do), yet with no substantial loss in performance (for nuclei segmentation, actually improved performance).

It's worth mentioning that some of the images from the nuclei segmentation task are used to train the siamese net, whereas none of the images from the cell segmentation task are. This suggests a new way to tackle segmentation problems when the annotated dataset is small. Of course, so called transfer learning is not new, but in such cases weights are typically initialized from networks trained on large amounts of annotated data, whereas in our case all the annotations for the "donnor" network were created automatically.

5 Conclusions and Future Work

We have demonstrated that siamese networks have the potential to improve surveillance of the published and in-peer-review literature for duplicated images. This approach may not prove accurate enough to definitively determine image duplication, but rather could serve to narrow down the pool of images which are subjected to further review.

We found that most errors in the real-world test set involved histogram/contrast alterations that are difficult to simulate, or scale changes beyond those the network was trained to detect. We will continue to explore synthetic manipulations as a way to improve accuracy of the algorithm.

As indicated by the self-similarity matrices in Figure 3, improvements are needed when different images are from the same category. We will improve the training procedure to sample more of these hard cases.

One of the main roadblocks to this research is the lack of a public, large-scale database of image manipulation cases on which to further test the model. The challenge here is not only of generating one such dataset, but also of securing the proper permissions to release the data, given the legal issues involved. We are continually expanding our dataset and will make it available as soon as possible.

The application to semantic segmentation was discovered somewhat by chance in an attempt to circumvent issues with the U-Net, mainly the need for a large corpus of annotations and difficulty setting hyperparameters. In contrast, the representation provided by the forensic siamese net is quite easy to deploy in conjunction with a random forest classifier.

REFERENCES

Belhassen Bayar and Matthew C. Stamm. A deep learning approach to universal image manipulation detection using a new convolutional layer. In *Proceedings of the 4th ACM Workshop on Information Hiding and Multimedia Security*, IH&MMSec '16, pp. 5–10, New York, NY, USA, 2016. ACM. ISBN 978-1-4503-4290-2. doi: 10.1145/2909827.2930786. URL http://doi.acm.org/10.1145/2909827.2930786.

E. M. Bik, A. Casadevall, and F. C. Fang. The prevalence of inappropriate image duplication in biomedical research publications. *mBio*, 7(3):published online, 2016.

M. Blatt and C. Martin. Manipulation and misconduct in the handling of image data. *The Plant Cell*, 25:3147–3148, 2013.

S. Chopra, R. Hadsell, and Y. LeCun. Learning a similarity metric discriminatively, with application to face verification. In *IEEE Computer Vision and Pattern Recognition*, volume 1, pp. 539–546, 2005.

D. W. Cromey. Avoiding twisted pixels: Ethical guidelines for the appropriate use and manipulation of scientific digital images. *Science and Engineering Ethics*, 16(4):639–667, 2009.

N. Gilbert. Science journals crack down on image manipulation. *Nature*, Oct 9:published online, 2009.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *CoRR*, abs/1502.03167, 2015. URL http://arxiv.org/abs/1502.03167.

G. Koch, R. Zemel, and R. Salakhutdinov. Siamese neural networks for one-shot image recognition. In *ICML Deep Learning Workshop*, volume 2, 2015.

L. Koppers, H. Wormer, and K. Ickstadt. Towards a systematic screening tool for quality assurance and semiautomatic fraud detection for images in the life sciences. *Science and Engineering Ethics*, 23(4):1113–1128, 2017.

Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. *CoRR*, abs/1505.04597, 2015. URL http://arxiv.org/abs/1505.04597.

M. Rossner. How to guard against image fraud. *The Scientist Magazine*, Mar 1:published online, 2006.

M. Rossner. A false sense of security. *Cell Biology*, 183(4):573–574, 2008.

M. Rossner and K. M. Yamada. What's in a picture? the temptation of image manipulation. *Cell Biology*, 166(1):11–15, 2004.

F. Schroff, D. Kalenichenko, and J. Philbin. Facenet: A unified embedding for face recognition and clustering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 815–823, 2015.

Andreas Veit, Serge Belongie, and Theofanis Karaletsos. Conditional similarity networks. In *Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, 2017. URL https://vision.cornell.edu/se3/wp-content/uploads/2017/04/CSN_CVPR-1.pdf.

Jiang Wang, Yang Song, Thomas Leung, Chuck Rosenberg, Jingbin Wang, James Philbin, Bo Chen, and Ying Wu. Learning fine-grained image similarity with deep ranking. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2014.

Zhou Wang and E. P. Simoncelli. Translation insensitive image similarity in complex wavelet domain. In *Proceedings. (ICASSP '05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.*, volume 2, pp. ii/573–ii/576 Vol. 2, March 2005. doi: 10.1109/ICASSP.2005.1415469.