Evaluating RoI Comparison Methods to Improve Copy-Move Forgery Detection

Arnav Ghosh
ag983@cornell.edu
Cornell University

Arun Pidugu
ap639@cornell.edu
Cornell University

Yuzhao Shen
ys525@cornell.edu
Cornell University

Abstract

Detecting and localizing image manipulation is a task made ever more relevant in the age of effective, accessible manipulation technology. We propose two pairwise region comparison methods that improve the copy-move detection performance of an existing two-stream Faster R-CNN forgery detector. One comparison method employs Siamese networks whereas the other uses CNNs that operate on concatenated representations of two regions. Experiments on our synthetic and standard datasets show that the latter, when used with RoIs from the two-stream network, markedly improves copy-move detection performance. This paper is limited due to computational constraints but our methods can be easily replicated.

1. Introduction

Image manipulation is an art that has steadily improved over the years. As the ability to construct misleading synthetic images improves, the ability for the general public to recognize well-manipulated images diminishes. From legal cases, where images are used as evidence, to news stories, where fake images are on the rise, detecting forms of manipulation would have a positive impact on society. For a truly effective detection system, a simple binary label denoting forgery is not enough, we would need to localize the region that has been tampered.

For this project, we refer to the three categories of image manipulation: (i) splicing, where a region is copied from one image and pasted onto another, (ii) copy-move, where the region is pasted in the same image and (iii) removal, where regions are erased and filled in with features consistent with the image.

In this paper, we aim to improve the ability of state-of-the-art methods to localize copy-move forgeries. We intend to show that comparing image regions as is the approach of traditional, non deep-learning approaches can be effectively applied to CNN-based approaches that produce good task-specific features. Finally, we evaluate the factors that make some comparison methods more effective than others.

2. Related Work

Existing work in image manipulation detection often relies on statistical analysis of image pixel value histograms such as in [10]. For copy-move detection, most approaches first define feature vectors from image blocks or keypoints, using techniques like Discrete Cosine Transforms, and then match these to identify original-tampered pairs [7]. However, these feature representations are often ill-suited for the task, because blocks fail to effectively localize the tampered region or the region itself lacks notable keypoints.

The state-of-the-art on most datasets, however, uses a two-stream Faster R-CNN approach introduced in [5] by Zhou et al. Briefly, the RGB-N first uses SRM filters to produce maps of noise patterns in the image, which are passed through a CNN. Simultaneously, the original RGB image is passed through a Faster-RCNN, with the noise representations concatenated to the feature maps before RoI pooling. The concatenated RoIs are then used to decide if the corresponding region is tampered while only the RGB RoI features are used for bounding box regression.

While this network detects all forms of forgery, we note that the noise and RGB streams are not as effective for copy-move detection. In particular, copied regions often share similar noise profiles as the region in which they're pasted and tampering artifacts are easier to hide in they same image. The network could, however, benefit from a technique used by previous methods: comparing regions to each other.

3. Proposed Methods

Given that the original copy of any copy-move tampered region is present within the same image, we seek to compare different regions and identify an original-tampered pair. Instead of representing regions via crops of the image, we choose to represent it with the RoIs RGB-N produces because these are well suited for detecting tampered objects. Each of the models described in this section performs a pairwise comparison of these features, assigning a score to each pair. Scores satisfying a threshold, chosen as a hyper-parameter, are assumed to be copy-move forgeries. From a chosen pair, we choose the region with the higher RPN rank as the tampered region and use the corresponding bounding box predicted by RGB-N to identify the tampered region in the image. By elimination, the other region in the pair corresponds to the original copy.

As far as we are aware, no attempts have been made to improve or augment the RGB-N architecture. Furthermore, while deep learning approaches to image forgery detection exist, those that compare image patches focus on copy-move detection in the limited domain of biological or scientific images [4] or focus only on identifying regions that have been post-processed with operations such as Gaussian blurring [1]. Finally, current approaches have not applied the two-channel network [11] to copy-move detection.

3.1. Baseline

Clearly, the approach used in [5] leads to RoIs that capture important semantic information about forged regions. Consequently, it is reasonable to assume that feature maps corresponding to tampered and original regions are already sufficiently similar and require no additional transformations before comparison. To test this hypothesis, we use **Cosine Similarity** on pairs of flattened feature maps and identify the pair with the smallest absolute score as the original-tampered pair.

3.2. Siamese Network

Siamese networks consist of two, weight-sharing convolutional branches. Each branch takes as input a distinct RoI and is followed by a fully connected layer that outputs a descriptor of the corresponding input. While various ways to compare these descriptors exist, for this paper we follow the approach in [6] and use their l_2 distance as a representation of their similarity.

We modify the network suggested in [6] slightly to account for the smaller height, width and higher number of channels of the 7x7x1024 RoIs. In particular, we still have three modules, each with a convolutional, ReLU and max-pooling operation but with convolutional kernel sizes of 3, 2 and 1 respectively. We arrived at these sizes experimentally using a subset of the synthetic dataset described in section 4.1. Our fully connected layer has 512 units. Finally, due to computational constraints, we assume that a subset of the RoI's channels is sufficient to distinguish the features and as such use only the first 5 of the 1024 channels as inputs to a siamese branch.

Defining the two input RoIs as x_1 and x_2, the l_2 distance between their descriptors as D_W, y as 1 if the two correspond to an original-tampered pair and 0 otherwise and a margin $m > 0$, we use contrastive loss [6] to train the above network:

$$L_S(W, y, x_1, x_2) = \frac{y}{2}(D_w)^2 + \frac{1-y}{2}(max(0, m - D_w))^2$$

Using this loss function forces the network to embed the RoIs of the two copies close together and separate unrelated RoIs by larger distances. In doing so, the network implicitly learns the similarities between the two RoIs, allowing it to identify these pairs.

3.3. Joint Comparison Network

Following an approach used in [11], we create a CNN followed by a decision network that takes as input two RoIs and concatenates them before operating on them. For an effective comparison, the CNN's structure exactly matches the single siamese branch described in section 3.2. The fully connected decision network consists of a hidden layer of 512 units and 1 output, with sigmoid activation.

Defining y as in section 3.2 and p as the network's output, we use binary cross-entropy loss to train the above network:

$$L_{JC} = -(ylog(p) + (1 - y)log(1 - p))$$

This approach views the problem from a different perspective. Instead of learning an implicit similarity function, operating on the two RoIs together affords the network the opportunity to learn the explicit differences between the pair. In doing so, it should learn to identify the original-tampered pair by discounting aspects of their RoIs.

4. Implementation

4.1. Datasets

We evaluate our models on the COVER [2] and CASIA [3] datasets, along with with our own synthetic dataset, which was used to alleviate concerns of the limited examples and ground truth bounding boxes of the original copied region in the former two. To create this dataset, we use a process similar to [5], where we use segmentation annotations to randomly select objects from COCO and paste them onto the same image (*copy-move*) or another image (*splicing*). Due to computational constraints, our synthetic dataset was created using objects from five categories: *kite*, *dog*, *sports ball*, *handbag* and *bottle*. These were chosen to represent the forgery of items in various contexts and of different sizes. Unlike [5], we did not have access to a dataset of *removal* forgeries.

	Copy-Move			Splicing	
	Synthetic	COVER	CASIA	Synthetic	CASIA
Train	7600	75	2946	7600	1664
Test	1900	25	328	1900	185

Table 1. Training and testing split on the three datasets.

4.2. Training

We train RGB-N on copy-move and spliced images from the synthetic dataset for 20 epochs. Next, we store the RoIs produced for copy-move examples. For each such image, we identify RoIs corresponding to regions with maximum IoU with the ground truth original and tampered regions. This is the original-tampered pair and all other pairs are untampered examples. Alternatively sampling positive and negative pairs from this set, we train our models for 10 epochs. We fine-tune our models on the other datasets using the same steps but with 5 and 3 epochs respectively.

5. Experiments

5.1. Overall Results

We use pixel-wise F1 to evaluate our methods, inline with [5] and using the protocol in [8] to calculate this score.

	Synthetic	COVER	CASIA
RGB-N [5]	0.120	0.082	0.101
Baseline	0.096	0.100	0.112
Siamese	0.143	0.062	0.099
Joint Comparison (JC)	0.338	0.212	0.287
RGB-N + JC	**0.362**	**0.232**	**0.310**

Table 2. F1 scores on three datasets. For Synthetic and CASIA, we limited the above evaluation to copy-move forgeries only.

Table 2 compares the F1 scores of RGB-N and our models. We replicate the evaluation protocol for RGB-N instead of reporting literature results because of the different training protocol used. Interestingly, each of the models alone performs better than RGB-N. However, we attribute this result to the low examples and time allotted to training this network, compared to [5]. However, because all our methods use the RoIs produced by RGB-N, we can still evaluate their relative performance.

The poor performance of the baseline suggests that the RoIs for the original-tampered pair are sufficiently differentiated, validating the need to test other methods. The contrast between the performances of the siamese and JC networks suggests that embedding RoIs individually into some feature space is insufficient and the flexibility of pairwise RoI comparison is crucial. The models' performance on CASIA highlights this point. JC's relatively more stable scores suggest that this method applies generally whereas the siamese embeddings are unable to abstract away the input image type, atleast without more training.

For RGB-N + JC, if both models assign different bounding boxes, we suppress that with the lower score. Otherwise, we choose the bounding box of the model that makes a prediction. This combination provides the best results. However, we expect a larger jump in performance compared to only JC. That this is not seen could be because of an overlap in JC and RGB-N predictions or that incorrect RGB-N bounding boxes are assigned higher scores. The latter could be remedied by using a different way to combine the two.

Lastly, all models see a drop in performance on COVER. Without further experiments, we hypothesize that this could be a result of the limited data available for fine tuning or the singular nature of copy-moves in all examples. Specifically, all authentic examples feature two instances of an object and the tampered set is created by copying one over the other.

5.2. Category Specific Copy-Moves

	Kites	Dogs	Sports Ball	Handbag	Bottle
Siamese	0.153	0.217	0.097	0.100	0.134
JC	0.567	0.519	0.287	0.297	0.314

Table 3. Category-wise F1 scores on copy-move forgeries in the synthetic dataset.

Table 3 focuses on the models' performance on specific types of copy-moved regions and allows us to qualify the generalizability of the JC method. Specifically, there are factors that make some comparisons easier than others. Qualitatively, we observe that kites and dogs occur as much larger tampered regions in the dataset, suggesting that the network is better at detecting such regions, given its much higher scores on these categories. Furthermore, we observe that handbags and bottles appear in more cluttered contexts than sports balls but this fact doesn't seem to affect the network's performance, suggesting that the context in which the copies appear is less relevant to the detection. These conclusions don't hold for the siamese network given its generally poor performance on all categories. Further experiments will be required to validate these observations.

5.3. Visualizing Comparisons by JC

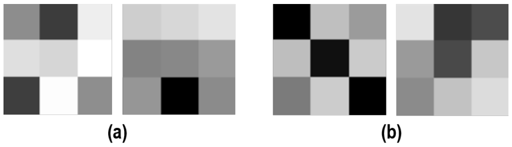

Figure 1. Sample partial filters from the first convolutional layer of JC. (a) and (b) are two different kernels. The left samples of (a), (b) show the first 3 layers of the kernel, whereas the right show the 6th to 8th layers.

Figure 1 displays partial filters used by the JC network. Similar to [11], we observe that some kernels (1a) are subtracting regions of one RoI from the other since the left and right visualizations are roughly negatives of each other. Importantly, this is not true for all the kernels, with 1b suggesting a more complicated method of differentiating the RoIs.

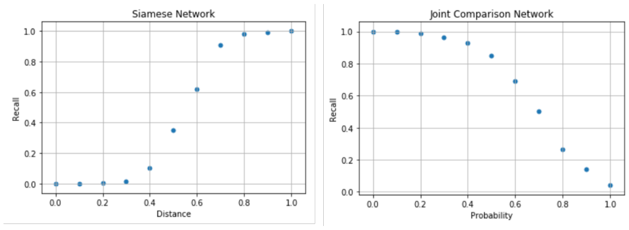

Figure 2. Recall *vs.* Threshold for the two models on copy-move forgeries in the synthetic dataset.

5.4. Analyzing Our Models' Predictions

Taking inspiration from [9], we calculate the recall at various thresholds in Figure 2. For each image, if the ground truth original-tampered pair's predicted distance is less than (*siamese*) or probability is greater than (*JC*) the threshold, we adjudge the prediction as correct. That we see a rapid increase in recall from 0 to 1 between the 0.4 and 0.8 distance mark shows us that most of the ground truth predictions by the siamese network are contained in this interval. This validates the idea that the network's poor performance is because of its inability to learn the similarities between the original-tampered RoIs.

Siamese	JC	Ground Truth
0.227 (0.279)	0.036 (0.117)	0.021 (0.082)

Table 4. Mean (and Std.) IoU of predicted and ground truth pairs.

We attempt to gain some insight into why it has difficulty doing so by examining the IoU of the pairs it identifies as being forged. From the high mean IoU in Table 4, we can conclude that the network is unable to effectively separate close, but un-tampered regions and instead only manages to perform a form of duplicate region detection. Coupled with the previous observation, this suggests that the network is unable to account for the different contexts in which the copies may appear.

5.5. Improving the Siamese Network

Results from the previous section motivate us to alter the process of sampling negative examples during training, to give the network a broader view of un-tampered pairs. We train the network on 15,200 RoI pairs produced from copy-move images in the synthetic dataset with a positive - negative sample ratio of 1 to 4. The 4 negative samples are chosen by pairing the original and tampered region each with a region that marginally overlaps with it and a region that doesn't overlap with it at all. Table 5 compares this approach to choosing 4 random negative pairs.

We see a drop in performance when choosing 4 random negative samples because of the imbalance in the number of examples from each class. Interestingly, however, not only does the performance improve when these examples are carefully chosen, it marginally outperforms our original network. This suggests that the problem of learning a similarity function is not as intractable as we earlier believed and that with enough data and the correct training protocol, the network could be effective.

1 : 2	1 : 4 (random)	1 : 4 (constructed)
0.143	0.110	0.149

Table 5. F1 scores on copy-move forgeries in the synthetic dataset using the siamese network trained with various sampling methods.

5.6. Effect on Other Types of Forgeries

RGB-N was designed to detect all 3 types of forgery. Table 6 shows that our augmentations don't lead to a significant drop in the network's splicing detection performance.

	Synthetic	CASIA
RGB-N	**0.134**	**0.129**
RGB-N + Siamese	0.134	0.125
RGB-N + JC	0.132	0.126

Table 6. Splicing F1 scores on two datasets.

6. Conclusion

We evaluate two augmentations to the RGB-N [5] network that compare RoIs to improve copy-move detection. Building off [11], we find that concatenating two RoIs to perform pairwise comparisons leads to a marked improvement in copy-move detection performance and does not significantly affect other forgery detection on our synthetic and standard datasets. We also show that using a Siamese Network yields a weak form of duplicate region detection, though experiments show that with enough data and a carefully designed training protocol, this network could be effective.

7. Future Directions

Three immediate extensions could explore more sophisticated ways to choose the correct RoI from the original-tampered pair, different ways to combine the JC and RGB-N networks and the application of triplet loss to our siamese network. Additionally, we believe it would be beneficial to have a process of incorporating global context when comparing RoIs, as opposed to simple pairwise comparisons.

8. Acknowledgements

We would like to thank David Chu (dc788) for lending us his Google Cloud Platform credits.

References

[1] Y. S. T. Aniruddha Mazumdar, Jaya Singh and P. K. Bora. Universal image manipulation detection using deep siamese convolutional neural network, 2018.

[2] R. S. X. S. B. Wen, Y. Zhu and S. Winkler. Coveragea novel database for copy-move forgery detection, 2016. In *IEEE International Conference on Image Processing (ICIP)*.

[3] W. W. J. Dong and T. Tan. Casia image tampering detection evaluation database, 2013. In *ChinaSIP*.

[4] D. R. D. W. M. Cicconet, H. Elliott and M. Walsh. Image forensics: Detecting duplication of scientific images with manipulation-invariant image similarity, 2018.

[5] V. I. M. P. Zhou, X. Han and L. S. Davis. Learning rich features for image manipulation detection, 2018. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*.

[6] S. C. Raia Hadsell and Y. LeCun. Dimensionality reduction by learning an invariant mapping, 2006. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*.

[7] J. M. Rani Susan Oommen and S. S. A survey of copy-move forgery detection techniques for digital images, 2015. In *International Journal of Innovations in Engineering and Technology (IJIET)*.

[8] Y. R. Ronald Salloum and C.-C. J. Kuo. Image splicing localization using a multi-task fully convolutional network (mfcn), 2017.

[9] R. G. Shaoqing Ren, Kaiming He and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks, 2015. In *Neural Information Processing Systems (NIPS)*.

[10] M. C. Stamm and K. J. R. Liu. Forensic detection of image tampering using intrinsic statistical fingerprints in histograms, 2009. In *Proc. APSIPA Annual Summit and Conference*.

[11] S. Zagoruyko and N. Komodakis. Learning to compare image patches via convolutional neural networks, 2015. In *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*.