CS230

Image Level Forgery Identification and Pixel Level Forgery Localization via a Convolutional Neural Network

Haitao D. Deng
Mater Sci & Eng Dept
Stanford University
dhtdean@stanford.edu

Yitao Qiu
Civil & Environ Eng Dept
Stanford University
yitaoqiu@stanford.edu

Abstract

Image forgery detection is important for data credibility. However, progress in generic detection technique development is slow due to the varieties in forgery types and complexity in forgery locations. Here, we report on a light-weight (5k parameters) VGG-like convolutional neural network architecture that allows for image forgery detection with an accuracy of 91% and AUC of 85% on test data. The modified network is adapted and coupled with a local anomaly detection network for forgery area localization and achieved an accuracy of 92% and AUC of 80% on test data. Our result demonstrates a relatively light-weight detection networks that is easy to train and adopt for image level forgery identification and pixel level localized area detection.

1 Introduction

The Information Age has brought about a wealth of digital information – in particular the images – coming from videos, photos and online publication. However, the ease of manipulation of digital data through editing/cropping tools such as photoshop and photoeditor etc has often negatively impacted the information credibility. The authenticity of these sources of information is thus of crucial importance.

While several successful cases for forgery detection from physical feature based models [3, 6] and machine learning techniques have been demonstrated [5, 7], progress for a generic detection technique development has been stagnant due to reasons below. One, there are two many fundamentally different forgery types (e.g. copy-move, splicing, content removal and enhancement, etc). Two, it's difficult to pin point the location of forged regions. [3, 5, 6, 7, 9]. Convolutional neural networks provide the opportunity to overcome barriers above-mentioned given its success in image classifications by generating non-linear artificial features. Presumably the spatially dependent features generated entail forgery information that helps localize forged area down to individual pixels. Here, we first report on a VGG [10] derived network architecture that takes an image input and performs image level binary classification with 90% accuracy to for forgery identification. Then, we adopt the similar model architecture by taking out the pooling layers and then coupling it with a local anomaly detection network to obtain pixel-wise forged area prediction and achieved an AUC of 80% on the test data set. Our model accuracy and performance is comparable to the state of the art models for generic forgery detection in literature [11], but with fewer number total parameters(more than an order of magnitude lower).

CS230: Deep Learning, Winter 2018, Stanford University, CA. (LateX template borrowed from NIPS 2017.)

Catagory	Features/Models		#Parameters	Forgery Types
ML	[4]	CFA	<10	S, CM, R
ML	[12]	ELA	<10	S, CM, R
ML	[8]	NOI	<10	S, CM, R
DL	[1]	Bayar	∼20M	S
DL	[2]	SRM	∼50k	S,CM,R
DL	[11]	Artificial	7M	S,CM, R, E

Table 1: Literature Review on Image Forgery Detection. ML: conventional machine learning, DL: deep learning. CFA: color filter array, ELA: error level analysis, NOI, noise. SRM: steganalysis rich model. S: Splicing, CM: copy-move, R: removal, E: enhancement.

Dataset	ratio of forged images	ratio of forged pixels
Train 1	31.09%	9.10%
Dev 1	29.20%%	8.42%
Test 1	30.20%	8.02%
Train 2	98.39%	16.13%
Dev 2	97.13%	16.85%

Table 2: forgery information of dataset 1 and dataset 2

2 Related work

Multiple machine learning approaches exist for image forgery detection. Most of them rely on extracting some forgery type specific engineering features that differentiate the tampered regions from untampered regions based on pixel level feature statistics. Recently, several deep learning algorithms have been developed for localization based on convolutional neural networks that allows for pixel-wise forgery detection. However, most networks (e.g. [11, 9]) contains multiple convolutional neural networks that are both wide and deep and therefore difficult to train. Table 1 gives a summary of the literature relevant.

Our goal of the project is to develop a network that is practically easy to train and test. To achieve this, we adopted two strategies: (1)by using smaller image sizes (2) by reducing the number of total parameters. Both are discussed in later sections.

3 Dataset and Features

Our data comes from the Image Manipulation Dataset [1] and the COCO Dataset [2]. In theory, our model is image size insensitive, but for practical reasons to train and excute the model, we divided the images into pixel size of 224 pixels × 224 pixels. The Image Manipulation Dataset contains 48 base images of size around 3000 × 2000, with the correpsonding ground truth mask (aka y labels). Using the division approaching, we generated 7200 images. Similarly, we produced another 2800 images with the size as 224 pixels × 224 pixels from the COCO Dataset. However, the COCO dataset only contains pristine images, and thus we manually added local enhancement to non-private regions via multiplication of a random coefficient at random regions and generated the mask accordingly. In summary, we have 10000 images (224 pixels × 224 pixels, 'Train1', 'Dev1' and 'Test1'), and split them into train set, development set, and test set with a ratio of 8:1:1. As in Table 2, 'Train1' and 'Dev1' contains only 30% forged images and 9% forged pixels. Since the localization model is a pixel-level model, 9% forged pixels would be insufficient for the training process. To resolve this issue, we manipulated some of the pristine images in the dataset by adjusting the RGB values of the pixels in a random area of the image ('Train2' and 'Dev2'). The ratio of forged pixels is now twice as that in 'Train1', 'Dev1'.

[1] https://www5.cs.fau.de/research/data/image-manipulation/

[2] http://cocodataset.org/#home

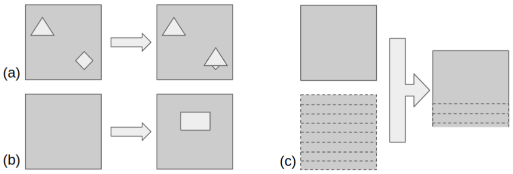

Figure 1: Type of forged images in the dataset: a) copy and move; b) Enhancement of RGB values at a random area; c) splicing

4 Methods

Here we first developed an image forgery detection network that allows for fake image identification. Because VGG[10] has been demonstrated to be effective for image classification, and roughly follows a far to near visual inspection, we thus based our model architecture on it. However, due to the limited data (8000 224x224 training images), the VGG network with nearly 13M parameters is likely an overkill and will cause significant overfit. We reduced network layers while maintaining the model test accuracy. The final network is shown in the schematics below. At the same time, because our goal is to develop a generic forgery detection network, the network should not be sensitive to image size, we further divided our image into 56x56 small frames and obtained 128k training image samples. Hyperparameters such as learning rate, optmizers and epoch number were explored, and the result provided later were all based on the highest training accuracy within the first 20 epochs. Model layers were also explored. Details can be seen in later section.

Second, we developed a network for localizing the forgery area. The pooling layers in the image level network condense the spatial information down to fewer pixels to make a final decision for classification. To achieve pixel wise prediction, the pooling layers need to be discarded, we thus have generated a pixel-wise feature extractor. Note that network from previous work such as Wu's[11] has parameters of more than 7 M while ours has 0.27 M, more than an order of magnitude smaller in size. For performance evaluation, we set our baseline as Wu's network [11] without training. We next looked at the model performance after training the Wu's local anomaly detection network (manTraNet LADN) while freezing the feature extraction network. For practical purposes, because VGG network sometimes suffers from vanishing gradient and training process can be very long, we've also explored the option of adding shortcut paths every two convolutional network in the intermediate blocks of the feature extractor network. Of note, end-to-end training of manTraNet was impractical due to oversized parameters to train. In this context, our light-weight model is more practical for model exploration purposes.

5 Experiments/Results/Discussion

5.1 Forgery Image Detection

We first built a network for forgery detection. We started with VGG [10] architecture roughly following a far-to-near approach. Two challenges exists: (1) Training data is not enough and overfit is unavoidable. (2) The model is too big to be trained on a 12 Gb GPU. We overcame this by cutting down layer numbers and divided our images into 56x56 smaller patches for training as classification should be insensitive to the pixel numbers. As a result, the number of parameters was significantly reduced and training data size is now 128k.

Learning rate, training batch size were tuned in order to speed up the training process. For classification, we specified a training batch size of 100 for 128 k total images. Each iteration took 1 min training was stopped after 20 epochs due to the convergence in accuracy (see Figure 2). In order to differentiate model performance, AUC plots were generated. Specifically, we took each test sample as a single data point and compared the sample-wise true positive rate vs false negative rate at different thresholds.

Because our training data (denoted as pristine training data) only contains less than 10% pixels that were fake, we augmented the data by incorporating more manipulation types manually to the training data (augmented data, see Figure **??**). This is done by selecting non-private regions and manually

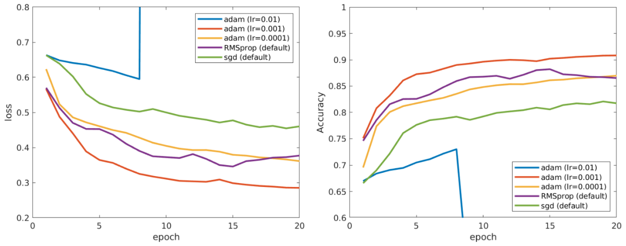

Figure 2: Hyperparameter tuning

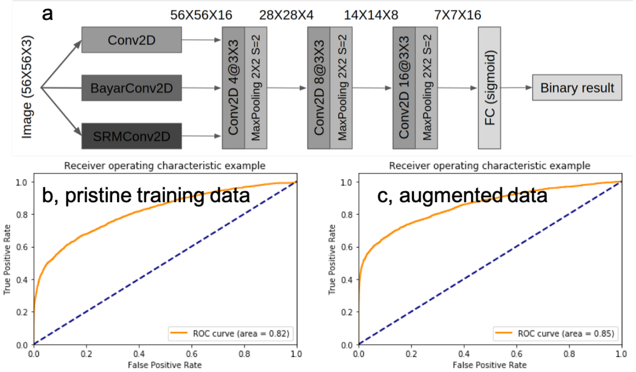

Figure 3: a, Classification network. b and c, AUC plots showing data augmentation result. All network are compared after training of 20 epochs.

enhancing the regions. To be specific, the regions were randomly chosen and multiplied by a different coefficient for enhancement. We have found that after data augmentation, the performance of the network has been increased (see Figure **??**)

5.2 Image forgery localization

As discussed in Methods, we generated a feature extrator network and fed into a local anomaly detection network proposed by Wu et al[11], see Figure 4.

Pixel-wise AUC plots were generated for performance evaluation. The network we developed achieved an AUC of 0.6, which is better than blind guessing. Following data augmentation method mentioned in classification, we improved the model AUC by 4%. Next, we made the network wider by generating more channels, (b and c in Figure **??**). Finally, we achieved an AUC value of almost 0.8.

Comparison of the network we developed (model C) with manTraNet is shown in Table **??**. We can see that our full model achieves better performance than the untrained raw model from Wu's paper and similar performance to the partially trained. To avoid the potential problem of vanishing gradients and to expedite the training process, we've also modified the network in C by adding shortcut paths every two convolution layers in the intermediate blocks (Model D) and have found similar performance within half the training time, Table **??**. We illustrate the network perforamnce by testing dataset from Wu's paper. (Figure 6) Of note, our network is much lighter-weight and

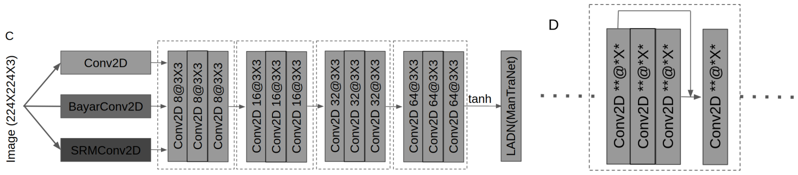

Figure 4: Full network for localization, model C and model D. LADN network was proposed in [11]

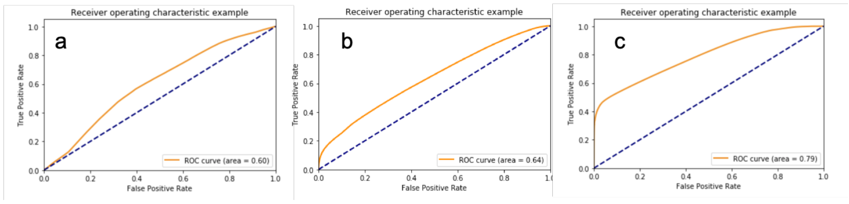

Figure 5: AUC plots for forgery localization network fine tuning. a, network trained on pristine training data. b, network trained on augmented training data. c, network trained on a wider feature extractor network. All network are compared after training of 20 epochs.

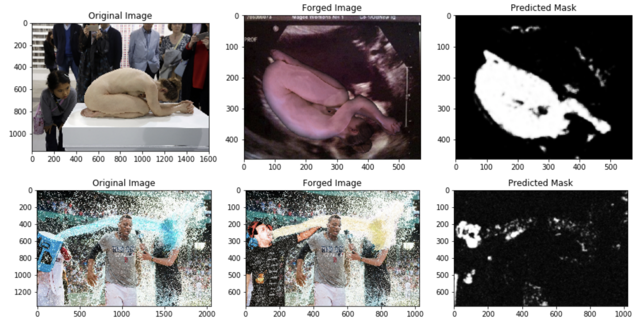

Figure 6: Forgery localization demonstration

provides a more practical route for end-to-end training. For example, training the full model of manTraNet requires a GPU memory of more than 12 Gb, which is the higher end of GPU memory most computers are equipped with.

6 Conclusion/Future Work

Here, we report on a light-weight (5k and 700k) parameters) VGG-derived convolutional neural network architecture that allows for image level forgery detection with an accuracy of 91% and AUC of 85% on test data and for 93% and AUC of 79% pixel level forgery detection. Given the performance, there is still room for accuracy improvement. For example, kernels that compute CFA, ELA and NOI features could be adopted in the first layer of the network which might help improve the performance. Nonetheless, our result demonstrates a relatively light-weight detection networks that is easy to train and adopt for image level forgery identification(classification) and pixel level localized area detection.

7 Contributions

The authors contributed equally to the work. H.D.D. wrote the report and implemented the classification and localization portion of the work. Y. Q. processed the training data and data augmentation. Both authors discussed the results and designed the experiment.

Model	Trained Parameters	Accuracy	AUC
Baseline 1: ManTraNet, A	0 (7 M in total)	0.93	0.67
Baseline 2: LADN trained ManTraNet, B	0.2 M (7M in total)	0.91	0.798
Test: Our Model, C	0.27 M (0.27 M in total)	0.92	0.794
Test: Our model with residual blocks, D	0.27 M (0.27 M in total)	0.93	0.78

Table 3: Localization network summary and comparison. Our model with 0.9 M parameter numbers performs similarly to MantraNet, which is out performs the model from MantraNet itself, probably due to data mismatchi.

Code

`https://github.com/dhtdean/IFDN`

References

[1] 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*. ACM. 2016, pp. 5–10.

[2] Belhassen Bayar and Matthew C Stamm. “Constrained convolutional neural networks: A new approach towards general purpose image manipulation detection”. In: *IEEE Transactions on Information Forensics and Security* 13.11 (2018), pp. 2691–2706.

[3] Gajanan K Birajdar and Vijay H Mankar. “Digital image forgery detection using passive techniques: A survey”. In: *Digital investigation* 10.3 (2013), pp. 226–245.

[4] Pasquale Ferrara et al. “Image forgery localization via fine-grained analysis of CFA artifacts”. In: *IEEE Transactions on Information Forensics and Security* 7.5 (2012), pp. 1566–1577.

[5] Gregory Hill and Emily Rager. “Image Forgery Detection”. In: ().

[6] Y.-F. Hsu and S.-F. Chang. “Detecting Image Splicing Using Geometry Invariants and Camera Characteristics Consistency”. In: *International Conference on Multimedia and Expo*. Toronto, Canada, 2006.

[7] Minyoung Huh et al. “Fighting fake news: Image splice detection via learned self-consistency”. In: *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018, pp. 101–117.

[8] Babak Mahdian and Stanislav Saic. “Using noise inconsistencies for blind image forensics”. In: *Image and Vision Computing* 27.10 (2009), pp. 1497–1503.

[9] Yuan Rao and Jiangqun Ni. “A deep learning approach to detection of splicing and copy-move forgeries in images”. In: *2016 IEEE International Workshop on Information Forensics and Security (WIFS)*. IEEE. 2016, pp. 1–6.

[10] Karen Simonyan and Andrew Zisserman. “Very deep convolutional networks for large-scale image recognition”. In: *arXiv preprint arXiv:1409.1556* (2014).

[11] Yue Wu, Wael AbdAlmageed, and Premkumar Natarajan. “ManTra-Net: Manipulation Tracing Network for Detection and Localization of Image Forgeries With Anomalous Features”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2019, pp. 9543–9552.

[12] Peng Zhou et al. “Learning rich features for image manipulation detection”. In: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018, pp. 1053–1061.