Viterbi Internship - Final Work Report

Arka Sadhu
Supervised by: Prof. Ram Nevatia

July 11, 2017

Contents

1 Abstract

Media forensics in general involves detection of the tampered media, identification of the tampered portion as well as trying to recover the original media.

2 Introduction

The work is done as a part of the MediFor Project. The MediFor project aims at pushing the state of the art research in the field of media forensics which in broad sense deals with the tampering of the media (image, video or audio) and its detection. This work only deals with image forensics. For each manipulated image the MediFor project demands the actual image on which manipulation is done (this is called the baseline image), the kind of manipulation, and in case of splice manipulation where one image is spliced onto another image it also demands the donor image. This work focuses only on the first part, where the aim is to find the baseline image. It is assumed that the world set contains the true baseline image. All experiments are done on Nimble Dataset which is publicly available for use.

3 Theory

3.1 Basic Definitions

- Probe Image : This is the given image. It may or may not be manipulated.
- Probe folder : Folder containing the probe images.
- Base Image : This the actual image corresponding to a probe image with no manipulations exists.
- Donor Image : In the case where the manipulation is such that a part of image A is pasted onto image B, then image A is called the Donor Image and B is the base image. The resulting image would be the manipulated image which would exist in the probe folder.
- World folder : Folder contaning all the images. This includes base, donor as well as the probe images.
- World set : The collection of images in the world folder. It is used interchangeably with world images.
- Provenance : Provenance in simple sense means the origin, so it defines the original image of a particular probe image.
- Provenance Graph : A relational graph which depicts all the transformations a particular baseline image would've undergone to reach the probe image. It is assumed that all the intermediate images are also a part of the world dataset.
- Base detection : Detection of the base image from a given probe image and the entire world set.
- Donor detection : Detection of the donor image from a given probe image and the entire world set.

3.2 MediFor Project

The MediFor project broadly has two main categories Video and Image. For any kind of media, MediFor Project wants automated assessment of the integrity of the media. If successful, the MediFor platform will automatically detect manipulations, provide detailed information about how these manipulations were performed, and reason about the overall integrity of visual media to facilitate decisions regarding the use of any questionable image or video.[1]

Table 1: Places365 Validation

Correct Matches	Total Images	Accuracy
2975	3650	81.5068493151
2969	3650	81.3424657534
2952	3650	80.8767123288
2993	3650	82
2977	3650	81.5616438356
3036	3650	83.1780821918
2941	3650	80.5753424658
2976	3650	81.5342465753
2941	3650	80.5753424658
2938	3650	80.4931506849

There are three technical areas of interest for integrity analytics. [2]
—-May need to add a few more lines here———

- Digital Integrity : This is related to the noise modelling and statistics and its consistency.
- Physical Integrity : This is related to shadow consistency.
- Semantic Integrity : This is related to semantic consistency

In this work we are concerned only with semantic integrity.

3.3 Contributions of this Work

3.4 Base Detection and Provenance

Base detection problem is essentially finding the underlying base image given a probe image. Here we make the assumption that the base image exists in the world set. The next problem is to get all the manipulated images derived from the base image. And beyond this is to create a provenance graph of the collected manipulated images. The last problem is not addressed in this work.

3.4.1 Neural Networks used

We use two pre-trained caffe [3] models in this work. AlexNet[4] trained on Places365[5] and AlexNet trained on ImageNet. The reason for using AlexNet instead of VGG16 or any other models is that we wanted to work with a simplest model and test our performances without compromising memory and time. Places365 is a scene-centric dataset while ImageNet is object centric dataset. And as such we expect there should be a difference in their base detection capability.

In this work we use the AlexNet trained on Places-365 everywhere unless explicitly mentioned that the AlexNet trained on ImageNet is used.

3.4.2 Which layer and metric to choose?

To find the baseline image, we employ the following method. We use the Nx1 dimensional vector produced by the network. As we go deeper into the layers, we expect more semantic features to be captured. The features are represented in the form of a vector and is known as a feature vector. In the AlexNet architecture we specifically compare three layers fc7, fc8, and prob layer which is the output after the operation of softmax function.

So we intend to find a way such that given the feature vectors from the probe image, we want to be find the base image. We use a simple approach for this. We find the feature vectors of the base image as well, and then compare the feature vectors using different metrics. For a metric to

be good we would ideally want for a probe base pair it should give a high value and for unrelated images it should return a very low value. Also we would prefer a substantial difference between related and unrelated images. For this work we tried the following metrics :

- SSD : Sum of Squared Distances
- SAD : Sum of Absolute Distances
- NCC : Pearson's correlation coefficient

It turned out that NCC gave the most desirable results.

Figure 1: Probe Base Pair taken from Nimble Dataset 2017 Dev 1 Beta 4

For example in the image pair 1 the metrics for the prob layers were as follows :

Table 2: Prob Layer Metrics

	Prob layer
SSD	0.072518922
SAD	0.29026049
NCC	0.98303729249860383

Clearly SAD is not desirable since it gives a medium score to a matching pair. Both SSD and NCC give good results in this case, but emperically it was found that NCC is not only easier for comparison (need not invert the high and low score), but is also more robust, that is gives high score even in cases where the images have been manipulated to larger degree and SSD isn't able to caputre the similarity. As a result, NCC has been the prime candidate for the rest of the work.

Another important part was to choose a layer. It was found that for many cases that using the last layer (prob) gave a very low score for a matching pair.

Figure 2: Probe Base Pair taken from Nimble Dataset 2017 Dev 1 Beta 4

For example the image pair 2 returns the following NCC scores for the three layers.

Table 3: NCC scores for different layers

	NCC
fc7	0.75791334934860821
fc8	0.87673938938958917
prob	0.35012885670990512

Clearly fc8 gives the most desirable result, but it is interesting to theorize the reason why prob gives such a low score. We hypothesize that the SoftMax layer in some sense disturbs the features because it gives the probability of closeness to a particular scene. So if a scene is not present in the Places365 database, this would give a weird output. Also going by the emperical knowledge that the deeper layers tend to extract out more semantic features, fc8 should give the best result and this intuition follows our finding. fc8 gives consistently higher score than fc7 and fc6 for probe base pair and lower score for unrelated images.

Here is a small table comparing the different methods

Table 4: Layers and Correlation threshold on the Nimble 2016 dataset which were known to be manipulated

Tot Images = 320	Prob		fc8		fc7	
Threshold	correct	%correct	correct	%correct	correct	%correct
0.95	203	0.634375	271	0.846875	175	0.546875
0.9	243	0.759375	288	0.9	243	0.759375
0.8	264	0.825	312	0.975	287	0.896875
0.5	295	0.921875	316	0.9875	313	0.978125
0.4	302	0.94375	320	1	316	0.9875

3.4.3 Speeding up the Feature Extraction Process

The feature extraction process (for all images in the world set) can be time consuming. For this reason we used multiprocessing to spawn new processes. One process was reserved for the Caffe net, and all the other processes were used for pre-processing the images. This lead to a significant speed. For the to process 3650 passes of through the net it took 652 seconds in a standalone code, while using multiprocessing reduced it to 87 seconds.

3.4.4 Image Slicing

A general observation in the datasets was that the manipulation existed in only a part of the probe image. For this reason, we use the method of image slicing, that is cutting the image into two halves horizontally or vertically, even getting four quadrants as well. Then we match each slice with the corresponding slice in the other image. This gives a very easy boost to the accuracy but at the same time demands more computational resources or time.

Here is the table for layer fc8 with horizontal slicing:

Table 5: Increase in accuracy using slicing

Tot Images = 320	fc8 sliced	
Threshold	correct	%correct
0.95	203	0.634375
0.9	243	0.759375
0.8	264	0.825
0.5	295	0.921875
0.4	302	0.94375

Some problems with this method include that it is not able to identify if the image has been rotated. This also fails miserably if the donor image occupies a significant (more than 70%) of the whole image.

3.4.5 Clustering using K-means

A detour atttempt was made to check how far the correlation matching could take us. We simply tried using the feature vectors derived from all the probe image which were known to be manipulated and all the world images from the Nimble 2017 Dev 1 Beta 4 dataset and tried to use a k-means clustering implemented in scikit-learn [6]. In the dataset there were 65 probe images which were known to be manipulated, and hence k in k-means was chosen to be 65. The number of iterations was kept to 100 but changing to 1000 or even higher didn't change the actual result. This table summarizes the findings :

Table 6: Clustering observation

Cluster observed	Number of clusters
Very good	37
one bad	10
two bad	3
two cluster overlap	6
bad cases	9

Very good implies no false positives or negatives, whereas one bad and two bad imply that there is one or two false positive. Two cluster overlap implies two clusters overlapped and couldn't form distinct clusters and a possible reason this happened would be because of less number cluster centres available. Bad cases include all other cases which includes random images together, three clusters, more than two bad images etc. Here is one very good cluster

Figure 3: Cluster 1 form NC2017 Dev 1 Beta 4 dataset

3.4.6 Getting all the Base Images

The Nimble Dataset 2017 Dev 3 Beta 1 had a lot more probe and world images, specifically 2157 manipulated probe images and 4098 world images. The world set not only consisted of the base image, but also of the probe image as well as the intermediate manipulated images. This essentially means that there is one particular base image and then subsequent manipulations on the top of that base image gives us the probe image. We aim to find all the manipulated images along with the base image.

For this we use a graph based approach. All graphs are made using networkx [7]. We first create a graph G with all the probe images as the node. Then we add all the images in the world (discarding the probe images) to the Graph and create an edge with all the nodes, with the weight of the edge as the correlation between the two images. We then start with one probe image, and then look at the edge with highest weight. We now contract the two nodes into one, and in this processs recompute the correlation taking the maximum of the two correlations. Then we repeat the process. The termination step is not exactly defined and for now we terminate based on the existing knowledge of the number of matches that should have occured (using the ground truth data). We then simply repeat this process for the rest of the probe images (initializing the Graph as well).

We use both Alexnet trained on Places365 as well as ImageNet. This method is henceforth referred to as recurrent base detection.

Table 7: Recurrent Base Detection

	Alexnet on Places		Alexnet on ImageNet	
No. of probe images	2157		2157	
No. of probe images with all baseline correct match	1120		1357	
	1120/2157	0.5192396847	1357/2157	0.6291145109
Total no. of base images	56223		56223	
No. of base images correctly identified	48732		49974	
	48732/56223	0.8667627128	49974/56223	0.8888533163

The first set of rows define the number of probe image with correct matches. We define a probe to be correctly matched if and only if all the manipulations were successfully captured. As can be seen this number is on the lower side. The second set of rows define the number of base images correctly recognised. That is for a particular probe image it is likely that there 7 correct images identified and 3 incorrect, even though this would make the probe image be an incorrect match, it would still be counted as 7 correct matches for the base images. We note that this is moderately on the higher side 85-89%.

It is quite interesting to note that the AlexNet trained on ImageNet outperforms the AlexNet trained on Places365. This is probably because the manipulations in the Nimble Dataset 2017 Dev 3 Beta 1 involved small manipulations.

3.4.7 Self Generated Images with more Manipulations

This was done mostly as an experiment to understand if it was possible to extend the use simple NCC to bigger manipulations, which involved more than 25% of the base image being covered by another image. For this we use scikit-image [8]. We pick any two images at random one being the base other being the donor, choose an angle of rotation at random, get a portion of the donor image (or the whole donor image) with both width and height half of that of the base image, then rotate it and place it on the base image to get a new image. We create 100 such images for using images from each of the dataset NC2017 Dev 1 and Dev 3. Now we use NCC to find the

correct matches. There were few errors in processing a few of them hence the reduced number of total images.

Table 8: NCC on NC2017 Dev 1 Beta 4

Dev 1	Top1	Top5	Top10
Places 365	37/89	61/89	71/89
	0.4157303371	0.6853932584	0.797752809
Places 365 (with 0.95 cut off)	50/89	72/89	76/89
	0.5617977528	0.808988764	0.8539325843
bvlc AlexNet	29/89	44/89	51/89
	0.3258426966	0.4943820225	0.5730337079
bvlc AlexNet(with 0.95 cut off)	34/89	47/89	54/89
	0.3820224719	0.5280898876	0.606741573

Table 9: NCC on NC2017 Dev 3 Beta 1

Dev 3	Top1	Top5	Top10
Places 365	10/92	28/92	46/92
	0.1086956522	0.3043478261	0.5
Places 365 (with 0.95 cut off)	50/92	63/92	66/92
	0.5434782609	0.6847826087	0.7173913043
bvlc AlexNet	3/92	17/92	29/92
	0.0326086957	0.1847826087	0.3152173913
bvlc AlexNet (with 0.95 cut off)	31/92	42/92	46/92
	0.3369565217	0.4565217391	0.5

In both the tables, the top-k denotes if the actual base image is found within the top k results. Since the datasets had many similar images (images with small manipulations), we have assumed that if the image predicted and the actual image have a correlation greater 0.95 then it can be said to be a correct match.

It is quite interesting to note that Places365 does significantly better than the ImageNet. This is presumably because the Places365 inherently tries to capture the scene information rather than the object information. While it is difficult to identify an object given a small part of it, it is much easier to identify the scene even if a significant portion of it is occluded. In this sense Places365 definitely proves to be a much better candidate than the ImageNet for cases with significant amount of manipulations.

3.4.8 ROC and CMC curves

3.5 Protest Dataset

3.5.1 Text Detection

References

[1] DARPA, "Medifor project description." `http://www.darpa.mil/program/media-forensics`.

[2] "Medifor project description 2." `https://researchfunding.duke.edu/media-forensics-medifor`.

[3] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional architecture for fast feature embedding," *arXiv preprint arXiv:1408.5093*, 2014.

[4] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in Neural Information Processing Systems 25* (F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, eds.), pp. 1097–1105, Curran Associates, Inc., 2012.

[5] B. Zhou, A. Lapedriza, A. Khosla, A. Oliva, and A. Torralba, "Places: A 10 million image database for scene recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[6] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[7] A. A. Hagberg, D. A. Schult, and P. J. Swart, "Exploring network structure, dynamics, and function using NetworkX," in *Proceedings of the 7th Python in Science Conference (SciPy2008)*, (Pasadena, CA USA), pp. 11–15, Aug. 2008.

[8] S. van der Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart, T. Yu, and the scikit-image contributors, "scikit-image: image processing in Python," *PeerJ*, vol. 2, p. e453, 6 2014.