Viterbi Internship - Final Work Report

Arka Sadhu
Supervised by: Prof. Ram Nevatia

July 14, 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. This work mainly aims at detecting the base image given a probe image. Some additional experiments augmenting the base detection have also been carried out. Finally an attempt has been made towards extending the ideas to donor image as well.

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]

There are three technical areas of interest for integrity analytics. [2]

- 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 Datasets Used

Most of the datasets used for the MediFor project is from the publicly available Nimble Challenge Dataset [3]. In this work we use NC2016, NC2017 Dev1 Beta4 and NC2017 Dev3 Beta1. There are some differences between the NC2016 and NC2017 datasets, but among the Dev1 and Dev3 there is no other difference other than the size of the dataset.

In the NC2016 dataset, there are two types of images. Nimble-SCI and Nimble-World. The former consists of images of objects in controlled environment, whereas the later consists more natural images taken from Flickr. There are references for manipulations, removal, and splice. In the project only those images which are of the later category as well as are a target of manipulation (not removal or splice) are considered.

In the NC2017 dataset, there are only natural images with intermediate manipulations. There is an additional reference of provenance and its provenance nodes. The former gives a list of one to one matching of a probe image the actual base image, while the latter gives a list of all intermediate manipulations.

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 [4] models in this work. AlexNet[5] trained on Places365[6] 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. Table 1 shows the top5 accuracy.

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

Table 1: Places365 Validation

Correct Matches	Total Images	Accuracy
2975	3650	81.51
2969	3650	81.34
2952	3650	80.88
2993	3650	82
2977	3650	81.56
3036	3650	83.18
2941	3650	80.58
2976	3650	81.53
2941	3650	80.58
2938	3650	80.49

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 Figure 1 the metrics for the prob layers using the AlexNet trained on Places365 are shown in Table 2.

Table 2: Prob Layer Metrics

	Prob layer (Places 365)
SSD	0.07
SAD	0.29
NCC	0.98

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 empirically 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 capture 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 Figure 2 returns the NCC scores for the three layers tabulated in Table 3.

Table 3: NCC scores for different layers

	NCC
fc7	0.76
fc8	0.88
prob	0.35

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 empirical 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.

Different layers are compared in Table 4

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	fraction correct	correct	fraction correct	correct	fraction correct
0.95	203	0.63	271	0.85	175	0.55
0.9	243	0.76	288	0.90	243	0.76
0.8	264	0.83	312	0.98	287	0.90
0.5	295	0.92	316	0.99	313	0.98
0.4	302	0.94	320	1	316	0.99

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.

The accuracy increase using the fc8 layer and horizontal slicing is tabulated in Table 5

Table 5: Increase in accuracy using slicing

Tot Images = 320	fc8 sliced	
Threshold	correct	fraction correct
0.95	203	0.63
0.9	243	0.76
0.8	264	0.83
0.5	295	0.92
0.4	302	0.94

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 attempt 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 [7]. 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. Table 6 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. One cluster is shown in Figure 3.

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 [8]. 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 process 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 occurred (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	0.52	1357/2157	0.63
Total no. of base images	56223		56223	
No. of base images correctly identified	48732		49974	
	48732/56223	0.87	49974/56223	0.89

Table 7 details the results on using recurrent base detection using both the datasets. 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 recognized. 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 [9]. 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. One such image is given in Figure 4

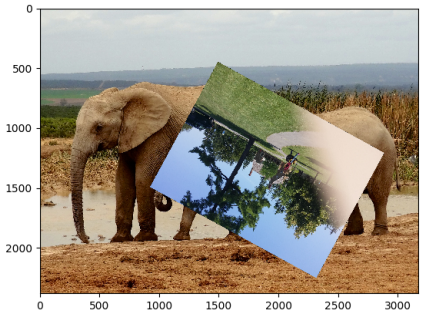


Figure 4: Self Gen Manipulated Image

Table 8: NCC on NC2017 Dev 1 Beta 4

Dev 1	Top1	Top5	Top10
Places 365	37/89	61/89	71/89
	0.42	0.69	0.80
Places 365 (with 0.95 cut off)	50/89	72/89	76/89
	0.56	0.81	0.85
Imagenet	29/89	44/89	51/89
	0.33	0.49	0.57
Imagenet (with 0.95 cut off)	34/89	47/89	54/89
	0.38	0.53	0.61

Table 9: NCC on NC2017 Dev 3 Beta 1

Dev 3	Top1	Top5	Top10
Places 365	10/92	28/92	46/92
	0.11	0.30	0.50
Places 365 (with 0.95 cut off)	50/92	63/92	66/92
	0.54	0.68	0.72
Imagenet	3/92	17/92	29/92
	0.03	0.18	0.32
Imagenet (with 0.95 cut off)	31/92	42/92	46/92
	0.34	0.46	0.50

In both Table 8 and Table 9, 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.

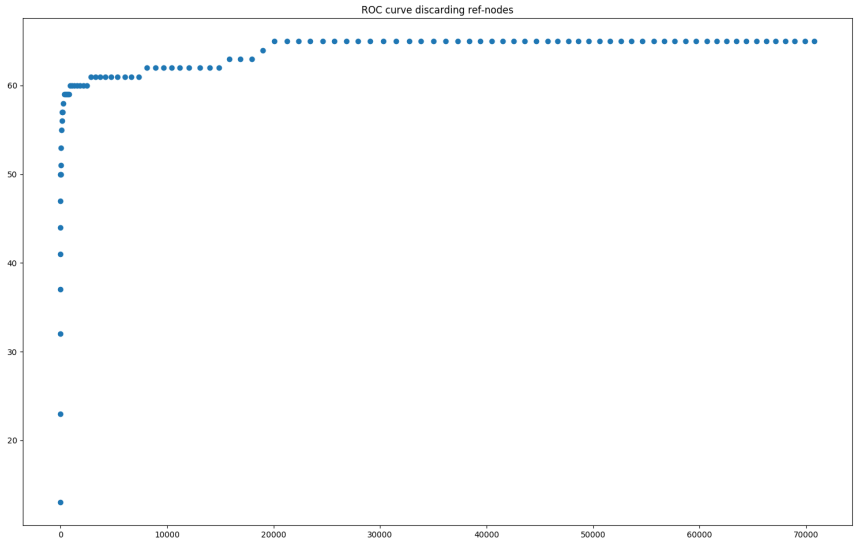
It is quite interesting to note that Places365 does significantly better than the ImageNet. This is presumably because the Places365 inherently tries to capture the scene information rather than the object information. While it is difficult to identify an object given a small part of it, it is much easier 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.

The low accuracy seen in Tables 8 and 9 is a point of worry. The most likely problem associated is that the manipulations are not natural, which means that manipulations are not smooth, and hence the features extracted by the net (Places365) are not able to capture the background features correctly.

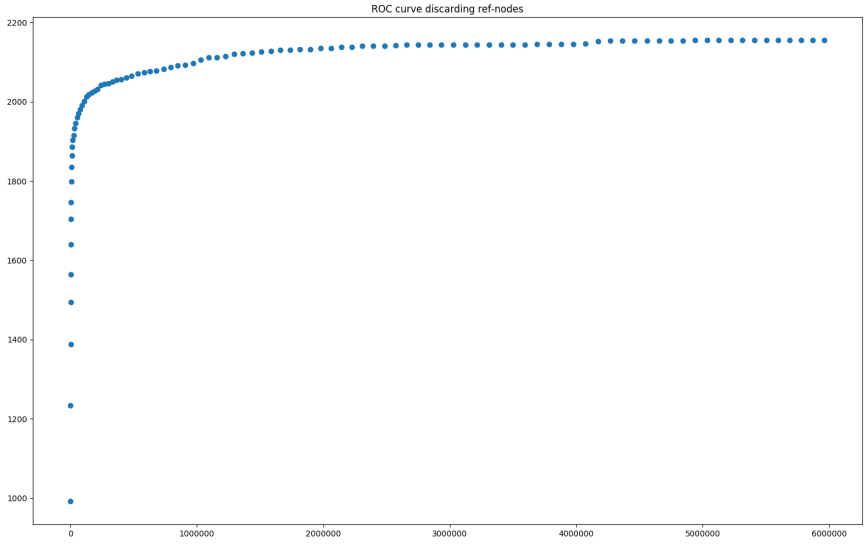
3.4.8 ROC curves

ROC (Receiver Operation Characteristics) is a metric to evaluate the performance of a system [10]. ROC curve is a plot of True positive (TP) vs False positive (FP) made at different thresholds. Because of the nature of the dataset that the world folder contains all probe images, intermediate manipulations and actual base images, special care was taken to discard irrelevant matching cases. First a graph is generated between each of the probe and world set (without the probe images), with their edges having weight equal to the correlation. Now a threshold for correlation is set and any edge with weight below the threshold is discarded. Now any node of the Graph which is connected to the probe image and is present in the list of intermediate manipulations is not counted. Of the remaining nodes, if the actual base image is present, it leads to increment in the number of true positives, and if it is neither the intermediate manipulations nor the base image,

it leads to increment in the number of false positives. For each threshold, this is done for each probe image (which is primarily the reason for high number of False Positives with less number of true positives). ROC curves for the dataset NC2017 Dev 1 and NC2017 Dev 3 are shown in 5



(a) ROC for NC2017 dev1



(b) ROC for NC2017 dev3

Figure 5: ROC Plots

3.5 Donor Detection with Protest Dataset

This is a dataset (containing about 1971 images) created using the keyword 'Protest' in YFCC (Yahoo Flickr Creative Commons 100 Million (YFCC100m) dataset) [11]. A few additional scenes were taken and the protest images were carefully cropped and put into the the scenes (with changes to illumination and lighting) to create 10 manipulated images.

Here is an example of the original and manipulated image 6

Figure 6: Original and Manipulated Images from Protest Dataset

The aim on this dataset is to try to match the donor (and not the base), i.e. donor detection. For this reason we used[12] to get bounding boxes based on the objectness score. We note that the donor image and at least one of the top 20 bounding boxes have (Intersection of Union)$IOU > 0.5$ with probability 0.96. We also note that if we also insert the condition that the bounding box also needs to circumscribe the donor image (perhaps for better detection) the probability reduces to 0.78.

Again as an experiment, we used the ground truth bounding boxes and cropped out the portion which contained the donor image and tried to find the best match in the world set which consisted of all the protest images. It was quite interesting to note that the comparison using the feature vectors from Places365 were terrible and gave 0/10 correct matches. Rather the feature vectors from Imagenet gave 7/10 correct matches.

This in some sense shows the distinction between Places365 and Imagenet. Places365 is trying to capture information about the scene which is in the background, while Imagenet is trying to capture foreground information about the objects, which is the reason that it is able to classify them correctly. An additional step was introduced, in which we do histogram equalization prior to sending them into the net, but that gave no improvement at all.

4 Miscellaneous

4.1 Text

4.1.1 Text Detection

Since there was no obvious way out, we speculated that perhaps matching text would be easier rather the whole image. Also since we are dealing with protest image, we expect there to be text in the image. For text detection we directly use the code provided by the paper [13]. We use text detection and not recognition because we are only interested in image matching and not in exactly what is written. The text detection works fairly well 7.

Figure 7: Text Detection in Modified and Original Images

But there are obvious problems with this method. There is no guarentee that the same part of the text in both images will be captured. Also this method will fail if there are no text scenes in the donor image. Also if the base image already consists many text scenes (like a city place or a mall) then there will be too many texts to be detected. Also even in the case of only donor image having text, the ordering of the bounding boxes may not be consistent, which will lead us to do a brute force search eventually. Even when two images of the same text are cropped out, there are considerable differences. One such example is in 8. The correlation between 8a and 8b is 0.84 while the correlation between 8a and 8c is 0.86 when using the AlexNet trained on Imagenet. We have been unable to identify the cause for such large correlation between two completely different text. This is most likely due to the same object being shown, that is even the ImageNet is not able to identify fine grained features like the text in the image.

(a) (b) (c)

Figure 8: Text cropped out

4.1.2 Text Segmentation

Since simple text detection didn't lead to any good results, we tried methods for text segmentation. We initially didn't want to focus on text recognition, because the text may be written in different languages, and should not really matter for the purpose of image matching.

We first explored simple segmentation techniques like the popular watershed algorithm (implemented in OpenCV)[14]. Unfortunately this didin't return good results. We then tried the Fully convolutional Network for semantic segmentation[15], but unfortunately all the models were trained on the PASCAL VOC dataset [16] and it doesn't have a class for text, so segmenting text was not possible.

We then also tried using the algorithm proposed in [17] but this too did not give satisfactory results.

We then turned our attention to text recognition methods rather than segmentation. We referred to the MS Coco Text dataset [18]. This paper cites Text spotter[19] which returned nice results on the protest dataset, but due to lack of time, no further progress was made in this direction.

5 Conclusion

This work shows a comparison between the Places365 and the Imagenet dataset, their fundamental differences in being scene-centric and object-centric datasets respectively and the corresponding effect on Image Matching. Both the datasets work quite well in cases where the image manipulations are less and we show empirically that using Pearson's correlation on the output of fc8 layer of the AlexNet is a good metric for image matching. In our experience AlexNet trained on Imagenet outperforms Alexnet trained on Places365 in such cases by a small margin.

In cases of non-sensical manipulations like in Figure 4 both neural nets show a significant drop in performance, but in different proportions. It is seen that Places365 results are much better than the ImageNet results. This is attributed to the ability of Places365 to detect features relating to the scene rather than the object. But at the same time it also shows that there remains a lot to be done for the scene detection, because a small portion of the scene should be enough to identify the actual scene and the neural net should not be fooled by the manipulation. In short the neural net should be able to distinguish a foreground object from the background object.

Another stark example is the case of protest dataset 6 where the Places365 doesn't give any useful information, so in a way this does prove that Places365 indeed detects features which are related to the background.

One of the easiest improvements to the base detection approach is to use a different net than AlexNet (like VGG16 or ResNet) for comparison. Perhaps a better approach could be to use semantic segmentation to identify the different objects and their relation to each other in an image. An approach could be to use all the relations in natural (non-manipulated images) and train a neural network on the relations obtained from the images to get a binary classifier whether the relation is semantically correct or not. In the process of obtaining relations the cases of text detection and recognition might need to be given certain emphasis and this might turn out to be fruitful direction for the future.

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] NIST, "Nimble challenge 2017 evaluation." `http://www.darpa.mil/program/media-forensics`.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] A. Ross, "Relating roc and cmc curves." `https://www.nist.gov/sites/default/files/documents/2016/12/06/12_ross_cmc-roc_ibpc2016.pdf`, 2016.

[11] B. Thomee, D. A. Shamma, G. Friedland, B. Elizalde, K. Ni, D. Poland, D. Borth, and L. Li, "The new data and new challenges in multimedia research," *CoRR*, vol. abs/1503.01817, 2015.

[12] K. E. A. van de Sande, J. R. R. Uijlings, T. Gevers, and A. W. M. Smeulders, "Segmentation as selective search for object recognition," in *2011 International Conference on Computer Vision*, pp. 1879–1886, Nov 2011.

[13] Z. Tian, W. Huang, T. He, P. He, and Y. Qiao, "Detecting text in natural image with connectionist text proposal network," *CoRR*, vol. abs/1609.03605, 2016.

[14] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[15] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[16] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International Journal of Computer Vision*, vol. 88, pp. 303–338, June 2010.

[17] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *Int. J. Comput. Vision*, vol. 59, pp. 167–181, Sept. 2004.

[18] A. Veit, T. Matera, L. Neumann, J. Matas, and S. Belongie, "Coco-text: Dataset and benchmark for text detection and recognition in natural images," in *arXiv preprint arXiv:1601.07140*, 2016.

[19] A. Gupta, A. Vedaldi, and A. Zisserman, "Synthetic data for text localisation in natural images," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016.