USC ISI DiSPARITY Self-Evaluation Report

1 Introduction

This report summarizes the self-evaluation efforts of the University of Southern California's Information Sciences Institute (USC/ISI) DiSPARITY team. The report is organized as follows. Section 2 presents the technical approach and self-evaluation results for the provenance detection task, including provenance filtering and provenance graph construction, as well as evaluation results on the NIMBLE 2017 dataset. Section 3 presents our novel, end-to-end, deep learning approach to splicing detection and localization, and the evaluation results on a new datasets that we internally created to train and test the developed approach. Finally, Section 4 summarizes the work done.

2 Provenance Detection

2.1 Problem Definition and Challenges

Provenance detection consists of two main tasks – provenance filtering and provenance graph construction. Provenance filtering can be thought of as a generalization of content-based image retrieval (CBIR), in which a query (or probe) image is used to retrieve the top N visually similar images from a reference (or world) dataset. In CBIR, the query images is assumed not to be digitally manipulate. In provenance filtering, however, the query/probe image is likely to include one or more digital manipulations (e.g. additions or removal of objects or people). Therefore, in provenance filtering, the task is to retrieve the base image, which contributed the scene background, and all donor images, which contributed smaller parts of the query image (in case of object addition, for example). The set of base and donor images constitute the provenance of a given probe image. In the second provenance detection task, i.e. provenance graph building, a directed genealogy graph must be constructed from the provenance of the probe images to represent how the probe was created step by step using its provenance images.

Retrieving the base of a query image is a relatively standard CBIR task. However, one of the main challenges in provenance detection is that the probe image must be segmented into different regions that can serve as *query images* for retrieving donors. Furthermore, reference images must be searched for matching regions of the queries, rather than overall image-to-image matching, which significantly increases the computational complexity of provenance detection. For example, we implemented a vanilla retrieval system using Constitutional Neural Network (CNN) features extracted using Overfeat Library [20] and the Fisher vector-based feature embedding [18][19]. Although the performance of image retrieval using this method yields 77.9% mean average precision on the Holidays dataset, the donor detection performance using this method on the NIST NC2016 [1] Web data only yields 13.6% recall rate from top 10 retrieval results.

2.2 Technical Approach

Figure 1 shows the overall architecture of the developed provenance detection system. Our system consists of a feature representation stage, which samples sub-regions from reference and probe images and extracts learned representations for each sub-region using VGG19 CNN. For a given probe image, base and donor images are retrieved independently and combined to provide the provenance set of the probe image. In order to construct a genealogy graph of the probe, we use a minimum spanning tree (MST) of the top 50 matching images from the provenance set. In the following, we discuss

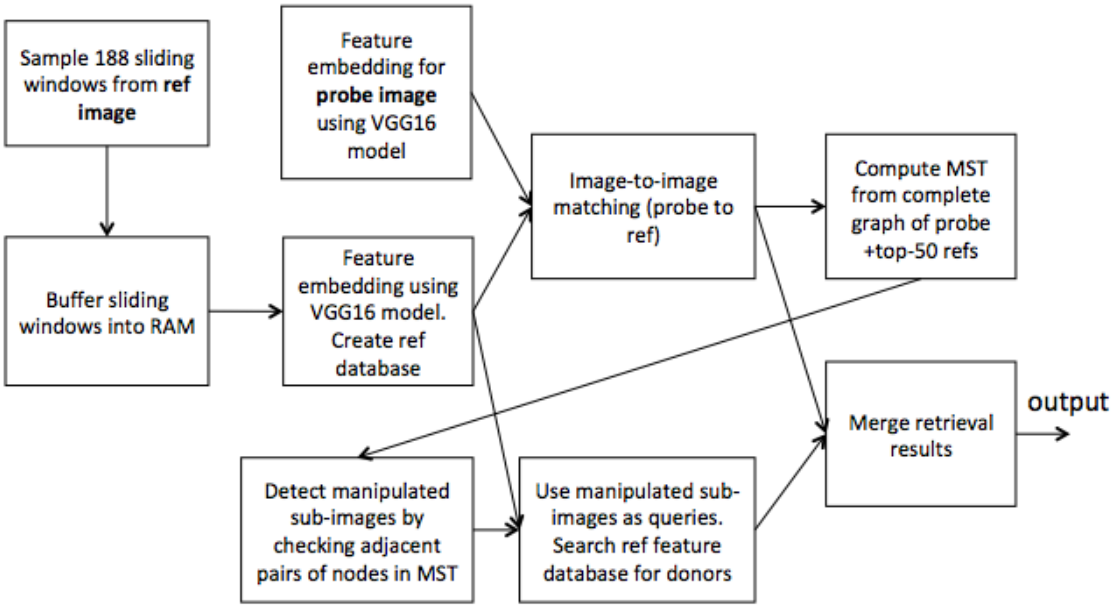

Figure 1: Provenance system diagram

each of these steps in more details.

Feature Representation

We use a *multi-scale* approach in which we exploit VGG19 CNN to extract learned representation of probe and reference images. Each image is processed at 6 different scales. At each scale, the image is divided evenly into multiple sub-images. The total number of sub-images for all 6 scales is 188. VGG is used to extract a 1000D feature vector for each scale and sub-images, creating a 188×1000 representation for each image.

Base Detection

It is important to clarify that a base image not only means the unaltered image that is used as a foundation for creating the tampered image, but also all intermediates images derived from the unaltered image. In the mean time, a donor image is not derived from the base. The donor image provides a source for an object (e.g. people, vehicles, etc.) to be spliced into the probe image. In this stage, we only detect all base images (unaltered and intermediate) that match the probe images

In order to retrieve the base image(s) of a given probe, we search the world set for reference images most similar to the probe image, by calculating the cosine similarity between each sub-image from the probe image and the corresponding sub-image in the reference image. The similarity between the probe image and the reference image is then calculated as the maximum of all these sub-image similarity scores. Only the top 3 scales are used in this step.

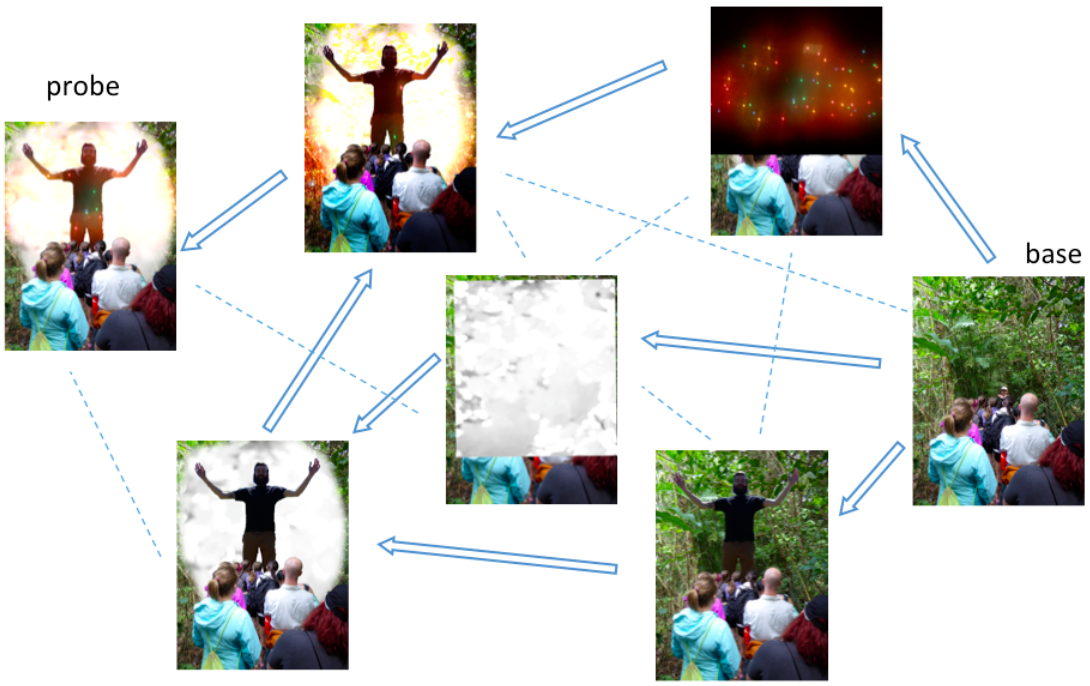

Figure 2: A sample genealogy graph of provenances

Donor detection

The major challenge in donor detection is finding objects spliced into base images during the course of manipulation and finally into the probe. Once these spliced objects are found, we can use them as queries, and search sub-regions in reference images for these queries.

Figure 2 illustrates our idea of finding spliced objects. First, we build complete graph using base images detected in the first round which is based on the base detection method mentioned above. We keep top N (we use $N = 50$ in the experimental evaluation) base images for each probe. The complete graph uses these 50 base images as nodes and pairwise similarity scores between them as edge values.

A minimum spanning tree is computed from this complete graph. Each pair of adjacent nodes from the minimum spanning tree are examined by subtracting one from another. The result image may have non-zero regions indicating manipulated regions in one of the two reference images. We take each manipulated region as a query, and search the world set for donors containing this query image. Since manipulations do not have to be splicing but can also be copy/move or some regional image enhancement, we only use a region of a base image as a donor query when the region is the only one found from that base. False alarms can be effectively filtered out using this strategy.

When we search reference images for donors, we compute the similarity between each donor query and each sub-region of a reference image. We still use VGG features and cosine similarity measurement. But all 6 scales, all 188 sub-images from each reference image, are considered in this step.

Provenance Set and Genealogy Graph

For a given probe image, we generate top-200 provenance set by top matching base and donor images based on the similarity scores. The procedure is optimized using NC2017 Dev3Beta1 dataset.

To generate the genealogy graph of a probe image, we start by creating the provenance set and the probe image, as discussed before. A minimum spanning tree is constructed using all pairs of images in the provenance set, where each image represents a node and the similarity scores represent the edge weights between the nodes. The graph is then reorganized such that the probe image is the root. Further, we prune the graph using the edge weights to produce only the top K nodes, where K is selected empirically using NC2017 Dev3Beta1 dataset.

2.3 Evaluation Data

The method was evaluated using a number of NIST NIMBLE dataset releases [1] [2]. The NC2016 Web dataset has 724 probes and a world set of 1124 reference images. Out of 724 probes, we found 128 that have spliced objects corresponding to 264 donor images in the world set of 1124 reference images. The NC2017Dev1Beta4 dataset has 65 probe images and a world set of 1631 reference images. The NC2017Dev3Beta1 dataset has 2260 probe images and a world set of 3441 reference images. The NIMBLE Evaluation world set has 1008681 reference images.

2.4 Evaluation Results

Segmentation-based donor detection:

We ran the first set of experiments on NC2016 Web to evaluate several donor detection strategies. Bases and donors were explicitly annotated in NC2016. Probes in NC2016 Web have very little modification in their background scenes. Thus, base detection was very accurate. We ran an image-to-image matching using 512d Fisher vectors embedding raw Overfeat features and got 100% recall rate from top 1 base detection results. We also used a Deep Metric Learning (DML) [27] to represent the image using a 128d feature vector. DML has several pre-trained models. First, we tried the model trained on the CARS dataset. We obtained manipulated regions for each probe by subtracting the probe from its base, and used these manipulated regions as queries to search the NC2016 world set for donor images. In contrast to the sampling approach described above, we tried the Berkeley Semantic Segmentation algorithm [24] to divide each reference image into semantically homogeneous regions, and computed a feature vector from each region (later, we will present experimental results showing the sampling method performs better than this segmentation method though.) For donor detection, we took the maximum of cosine similarities between donor queries of the probe and regions of the reference image as the similarity between the probe and the reference image. As a contrastive condition, we also measured recall rates of donors using image-to-image matching score (no selection of sub-images from the probe and the reference image). The recall rates of donors from top 10 results of these experiments are shown in Table 1. By comparing the first two rows of results, the DML features from the CARS model outperforms the Fisher vector method. By comparing the 2^{nd} and the 3^{rd} rows of results, the introduction of segmentation schemes for both probes and reference images significantly improved donor detection performance.

Table 1: Impact of the segmentation scheme on donor detection (NC2016).

Method	Top 10 recall rate
Fisher w/o segmentation	13.6%
DML (CARS) w/o segmentation	18.9%
DML (CARS) w/ semantic segmentation	31.4%

Fine-tuning Donor Detection

We made two attempts to further improve our core algorithm for donor detection. First, keeping Berkerly's semantic segmentation unchanged, we tried the pre-trained DML models using the online_products dataset and CUB dataset, respectively, and two Imagenet contest participating object detection feature extractors: Inception V3 [29] and VGG-19 [26]. The recall rates of donor detection from top 10 results are shown in Table 2. The best performance is obtained from using the VGG-19 model.

Table 2: Comparison of feature embedding models (NC2016).

Method	Top 10 recall rate
DML (CARS) w/ semantic segmentation	31.4%
DML (online_products) w/ semantic segmentation	45.1%
DML (CUB) w/ semantic segmentation	61.7%
Inception V3 w/ semantic segmentation	81.1%
VGG-19 w/ semantic segmentation	81.8%

The second attempt was to replace semantic segmentation with dense sampling of probe and reference images. As described before, we used 188 sub-images and extracted CNN learned representations from each sub-image. Table 3 indicates significant improvement of donor detection.

Table 3: Comparison of semantic segmentation vs. densely sampled sub-images (NC2016).

Method	Top 10 recall rate
DML (CUB) w/ semantic segmentation	61.7%
DML (CUB) w/ densely sampled sub-images	81.8%

Provenance Filtering Performance on NC2017 Data

We evaluated provenance filtering performance on both NC2017Dev1Beta4 and NC2017Dev3Beta1. The experiment on NC2017Dev1Beta4 used the probe set and world set from NC2017Dev1Beta4 only. We generated top 200 provenance filtering results by taking top 50 base detection results and additional top 150 donor detection results. The performance is shown in Table 4.

Table 4: Provenance filtering results on NC2017Dev1Beta4.

Result	Recall rate
Top 50 base detection results	78.1%
Top 50 base detection + top 150 donor detection results	90.9%

For NC2017Dev3Beta1, we had a chance to run our search system on a much larger world set created by merging the NC2017Dev3Beta1 world set and the NIMBLE evaluation world set. To generate top 200 results for each probe, we first filled the bucket with top 110 detected base images, and then filled the 90 remaining with top donors detected. When fusing base images and donor images, we used the cosine similarity scores for base images but did not use them for donors. Instead, we used the following score for each donor:

$$score(\text{donor}) = 0.9/\text{rank}(\text{donor}) \quad (1)$$

where rank(donor) is the donor detection rank for the donor image. This score is always smaller or equal to 0.9. The cap 0.9 imposed to donor scores prevent donors from being 1.0 when rank(donor) is 1. The performance is shown in Table 5. The gain from adding donor detection results is very small. We cannot find which provenances are bases or donors from the ground truth of that dataset but we found the dataset has very few donors after a visual examination.

Table 5: Provenance filtering results on NC2017Dev3Beta1.

Result	Recall rate (base only)	Recall rate (110 bases and then 90 donors)
Top 50 results	93.7%	94.0%
Top 100 results	93.9%	94.3%
Top 200 results	93.9%	94.3%

Graph Building Results

We evaluated our provenance graph building method with respect to different maximum number of nodes on NC2017 Dev3-Beta1 dataset. Table 6 shows these results, where best scores are indicated in bold font.

Table 6: Provenance results on NC2017Dev3Beta1.

#Max Nodes	MeanSimNLO	MeanSimNO	MeanSimLO	MeanNodeRecall
5	0.181238125315	0.323336383573	0.0294198584849	0.193914517725
10	0.314880510106	0.534688387660	0.0794494984609	0.371907844130
15	0.409656695674	0.682396122668	0.1170226257260	0.540148103242
20	0.472234556772	**0.777937157058**	0.1403277670010	0.690469666044
25	**0.477790504935**	0.765763055720	**0.1473198018190**	0.752900066207
30	0.456001256462	0.705338594064	0.1462554818990	0.759856364840
35	0.436594355859	0.654438834853	0.1454015553690	0.765888823850
40	0.418745397019	0.610120086271	0.1448606644320	0.770309485196
45	0.402475077677	0.571594074708	0.1444433649680	0.773902573332
50	0.387435517583	0.537760247947	0.1439085714310	**0.776572327323**

2.5 Discussion

Experimental evaluations show that base detection has much higher accuracy than donor detection. This is primarily because of the fact that base regions are much larger than donor regions, and therefore representation matching for base regions is relatively easier. We will conduct a detailed

analysis of detection results to determine failure cases for donor and base detection. We currently use VGG19 for extraction learned representations. In order to improve base detection, we plan to use more recent feature extractors, such as Residual Networks and Inception networks.

In terms of donor detection, we have shown that using segmented objects as queries improves the detection results, since it generates a more object-specific representation, in contrast to direct image-to-image matching. Rather than using a fixed image sub-sampling (188 windows), we will continue investigating using image segmentation and object detection techniques. Further, we will also investigate re-ranking methods to improve the accuracy of donor detection.

To improve the accuracy of our provenance graph construction algorithm, we plan to develop multi-tasking deep networks that can classify the manipulation type and estimate the manipulation parameters, if any. Even though the NC2017Dev3Beta1 dataset only has a very small fraction of donor provenances (note that links coming from donor nodes are far less accurate than links connecting base images due to the lower provenance detection performance of donors than that of base images), the 14.7% link similarity still looks a bit lower than what we expect to be useful practically. To overcome the data limitation challenge, we will create a new dataset that can be used to train the deep network. The new dataset will be similar to the one we created for splicing detection, with emphasis on probe-donor relationships and manipulation parameters.