PHYLOGENETIC MINIMUM SPANNING TREE RECONSTRUCTION USING AUTOENCODERS

Riccardo Castelletto[♯], *Simone Milani*[♯], *Paolo Bestagini*[♭]

[♯]Dept. of Information Engineering (DEI), University of Padova, Italy
[♭]Dipartimento di Elettronica, Informazione e Bioingegneria (DEIB), Politecnico di Milano, Italy
e-mail: riccardo.castelletto@studenti.unipd.it, simone.milani@dei.unipd.it, paolo.bestagini@polimi.it

ABSTRACT

The history of a shared and re-posted multimedia content can be reconstructed by analyzing the mutual relations between all of its near-duplicate copies and solving a minimum spanning tree (MST) problem, as shown by multimedia phylogeny research field. Unfortunately, MST estimation strategies are severely impaired by the noise affecting dissimilarity measures between pairs of near-duplicate contents. For this reason, researchers have recently been investigating robust dissimilarity metrics.

This paper proposes a matrix denoising solution that both mitigates dissimilarity noise and reconstruct the desired phylogenetic tree at the same time. The proposed strategy is a first attempt to estimate a MST via a denoising autoencoder that returns an approximation of the adjacency matrix corresponding to the underlying tree. Experimental results prove that the proposed solution outperforms the previous approaches and easily adapts to different analysis scenarios.

Index Terms— noisy minimum spanning tree, image phylogeny, autoencoder, UNET

1. INTRODUCTION

The diffusion of digital contents (e.g., images, videos, twits, etc.) across the Internet is usually coupled with their alterations, as they mutate and change after each downloading/uploading operation. Given a set of N similar contents (also called *near-duplicates*, i.e., connected by a chain of modifications), it is possible to model the history of alterations with a tree (called phylogenetic tree or PT) [1, 2, 3, 4, 5]. The reconstruction of the phylogenetic tree relies on two core operations: the estimation of the similarity/dissimilarity between pairs of near-duplicate elements, and the estimation of the tree structure. Dissimilarity estimation is usually a computationally-demanding and complex operation that parameterizes how much similar two contents are. This allows building a complete directed graph where each node corresponds to one of the contents and the weight of each edge (i, j) is associated to the dissimilarity between the i-th and the j-th elements. As a result, the resulting PT can be associated to the minimum spanning tree (MST) underlying the generated graph.

Unfortunately, the accuracy of PT estimation is strongly degraded by the noise affecting the computed dissimilarity values. In fact, most of the multimedia phylogenetic algorithms make some assumptions on the sets of possible transformations that led to the creation of the analyzed set. If these are not well-modelled by the dissimilarity metric (which happens very frequently), the relations between different contents are not accurately characterized. Moreover, it is possible that some nodes/contents that are part of the real evolution tree are missing from the analyzed datasets, and therefore, it is impossible to recover the real structure. All of these non-idealities add a significant noise component to graph edges that often leads to a wrong PT reconstruction since most of the MST estimation algorithms prove to be optimal on error-free edge weights.

In this paper, we propose a MST estimation strategy that denoises the original dissimilarity matrix computed on a set of near duplicate images and outputs a final adjacency matrix that corresponds to the estimated image phylogeny tree. The proposed solution employs a UNET autoencoder defined by a concatenation of convolutional and deconvolutional layers. The network is trained using a set of appropriate loss functions that force the network in generating a proper adjacency matrix. Experimental results show that the final accuracy of the reconstructed tree is much higher than that provided by a traditional minimum spanning tree algorithm. To the best of our knowledge, this is the first attempt in using an image denoising strategy to solve a noisy MST problem. The approach can be extended to other fields where MST reconstruction is impaired due to the presence of distorted weight values [6] (e.g., communication networks, stock markets, etc.).

The rest of the paper is organized as follows. Section 2 describes the phylogenetic reconstruction problem and overviews some state-of-the-art solutions. Section 3 models the MST estimation as a denoising problem and describes the adopted network. Section 4 reports the accuracy of the reconstruction approach on different datasets and conditions, while final conclusions are drawn in Section 5.

2. BACKGROUND ON PHYLOGENETIC TREE ESTIMATION

As it was anticipated in the previous section, the first step of a phylogenetic analysis is the computation of dissimilarity values between pairs of near-duplicate contents. This operation is computationally-burdensome (since)it must be repeated for every pair of contents in the dataset) and impaired by strong noise levels since the analyst has a limited knowledge about the possible modifications that were operated [7]. The authors of [8] assume that the analyst knows exactly the set of possible transformations but ignores the exact value of the transformation parameters. He/she exhaustively tests a range of possible parameter values and chooses the one that minimizes the final mean square error. In [9, 10], authors assume that editing steps from a source image I_k to a target image I_t can be approximated by an affine transformation, which can be estimated by computing local descriptors on I_k, I_t (e.g., SIFT, SURF, etc.), finding the matching points, computing the homography matrix $H_{k,t}$ and warping I_k onto I_t after an equalization of color histogram values. This approach proves to be more flexible and less computational demanding, and therefore, it has also been adopted for spectrograms of audio tracks in [11, 12]. As for video contents, three-dimensional descriptors permit to match and align chunks of video sequences whenever simple cuts and interpolations were applied [13, 3]. In [14] word embedding has been used to generate dissimilarity measures between different texts, while cache miss statistics permits comparing different softwares that belong to a common development thread in [5].

Unfortunately, the reconstruction of a reliable phylogenetic tree proves to be quite hard since the resulting dissimilarity values are affected by a significant amount of noise [15], and standard MST estimation algorithms perform quite poorly [6]. For this reasons, researchers have tried to improve the accuracy of reconstruction by introducing noise robust strategies. Some of the proposed approaches aim at employing dissimilarity measurements that prove to be robust to noise [16]. The work in [17] evaluates different types of dissimilarity metrics in image phylogeny by measuring the final MST reconstruction performance. Other solutions resort to a probabilistic formulation of the metric in order to mitigate the final amount of noise on dissimilarity values [18]. Some of the proposed strategies introduces additional checks and comparison metrics to improve the reconstruction. The approach in [15] performs some dependency checks to avoid wrong parent-child assumptions. The solutions in [19, 20] introduce a no-reference aging measure that permits inferring the level of each node in the tree. The approach in [21] combines multiple dissimilarity metrics in order to have a more reliable reconstruction. It is also possible to estimate a noise-robust measurement using a deep learning approach [22].

The approach presented in this paper focuses on the MST reconstruction process: the core idea is to include noisy measures in the estimation algorithm and reduce the probability of faulty parenthood assumptions by processing the whole dissimilarity matrix rather than going through single local choices (as it is done by most MST estimation algorithm, like Kruskal). The following section will explain how.

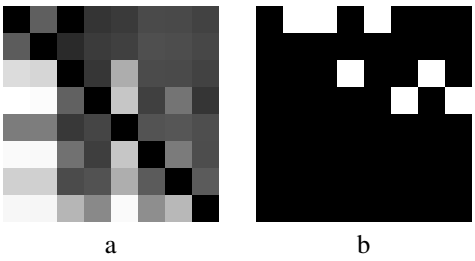

Fig. 1. Example of dissimilarity matrix D (a) and its corresponding adjacency matrix A (b) with $N = 8$.

3. DENOISING AUTOENCODER FOR MINIMUM SPANNING TREE RECONSTRUCTION

3.1. Problem setting

Given a set of N near-duplicate images I_k, it is possible to estimate an $N \times N$ matrix $D = [d_{k,t}]$, where $d_{k,t}$ is the dissimilarity between images I_k and I_t computed with the same strategy of [9, 10]. We assume that the matrix D is normalized with respect to the maximum dissimilarity value so that $d_{k,t} \in [0, 1]$. This matrix can be coupled to a ground truth adjacency matrix $A = [a_{k,t}]$ where $a_{k,t}$ is equal to 1 if image I_k is the parent of the image I_t and 0 otherwise; as a matter of fact, the matrix A represents the image phylogenetic tree to be reconstructed (Fig. 1 reports an example for D and A). Since in ideal conditions $d_{k,t} \to 0$ whenever I_t has been generated from I_k and $d_{k,t} \to 1$ otherwise, it is possible to write that

$$d_{k,t} = (1 - a_{k,t}) + e_{k,t} \tag{1}$$

where $E = [e_{k,t}]$ is a noise matrix that models all the non-idealities of the dissimilarity computation. From this premises, estimating A from D corresponds to a matrix/image denoising problem, and it is possible to inherit many solutions from the computer vision world. In this work, we adopt a UNET-based denoising autoencoder that permits obtaining from D an accurate adjacency matrix $\tilde{A} \simeq A$ with a limited computational effort. Since the final estimation $\tilde{a}_{k,t}$ will hardly be binary, oriented Kruskal algorithm [1] is run on $\tilde{A}$ to get the final tree. In the following, more details will be provided regarding the denoising network.

3.2. Autoencoder structure

The architecture behind this model has been built based on the work proposed in [23]. Fig. 2 reports a block diagram of the network operating on dissimilarity matrices with size 16×16. It is possible to notice that the proposed struc-

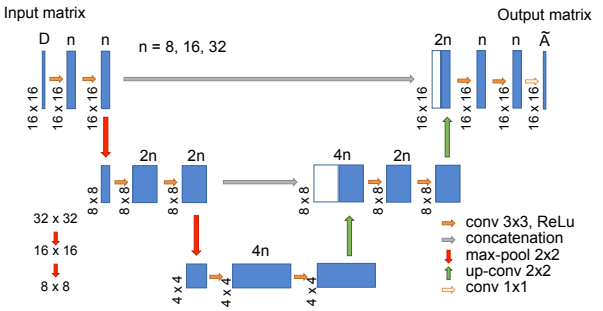

Fig. 2. Block diagram of the adopted denoising autoencoder.

ture is a convolutional autoencoder with skip connections between encodig/decoding stages at the same decomposition level. In fact, convolutional coding layers are followed by max-pooling units that perform a dimensionality reduction. Decoding stages are specular to the encoding ones in terms of operations and order: the compressed layer are rescaled via a set of up-convolution layers and concatenated with the input data by means of skip connections. A final convolutional layer is applied in order to combine both input and upsampled features. This procedure is hierarchically re-applied several times depending on the size of the input.

The general structure of the network built for this work is a simplification of the original model, and it is reported in Fig. 2. In the following, we report some details concerning its building blocks.

- Convolutional layers employ n (3×3) filters, with a ReLu activation function and zero-padding.
- Max pooling layers operate on a 2×2 neighborhood (halving the dimensions each time).
- As a matter of fact, up-convolution layers use a stride value equal to 2 and hold interpolation for padding.
- Some dropout layers are added in order to avoid overfitting.
- The last layer is a (1×1) convolutional layer that outputs the final prediction.

The size of the input matrices D determines the number of decomposition that can be applied: we have only one decomposition level for size 8×8, two for 16×16 and 32×32.

3.3. Training strategy

The designed architecture processes the n-th input D^n and generates an output matrix $\tilde{A}^n$ which needs to be as close as possible to A^n. In this way, the network is trained to solve a minimum spanning tree problem. This task can be effectively accomplished by choosing an appropriate set of loss functions to be minimized over the pairs (D^n, A^n) composing the dataset.

Operation	Parameter range
Rotation	$[-5°, +5°]$
Rescaling	$[90\%, 110\%]$
Cropping	$[0\%, 10\%] \times [0\%, 10\%]$
JPEG QF	$[75, 100]$

Table 1. Transformation and parameters used to generate each dataset.

The first loss function we considered is a simple Frobenius norm between $\tilde{A}^n$ and A^n, i.e.,

$$L_a\left(\tilde{A}^n, A^n\right) = \sum_n \|\tilde{A}^n - A^n\|_F. \tag{2}$$

This accuracy measurement was refined considering an additional function L_b that measures how much $\tilde{A}^n$ presents $N-1$ columns with norms equal to 1 (like the adjacency matrix of a directed graph without loops), i.e.,

$$L_b\left(\tilde{A}^n\right) = \left|\sum_k \|\tilde{A}^n_k\|_2 - N + 1\right| \tag{3}$$

where $\tilde{A}^n_k$ denotes the k-th column of $\tilde{A}^n$. This metric measures how likely the nodes in the reconstructed graph have one parent only.

In the end, as $\tilde{A}^n$ get closer to A^n, it is possible to approximate the resulting adjacency matrix by thresholding $\tilde{A}^n$ with respect to a value $\delta \in [0, 1]$. Therefore, it is possible to include a third loss function

$$L_c\left(\tilde{A}^n, A^n\right) = \sum_{k,t} \mathcal{I}\left(\tilde{a}^n_{k,t} > \delta\right) \oplus a_{k,t} \tag{4}$$

where $\mathcal{I}(\cdot)$ is the indicating function and $\oplus$ denote a xor operation. This third measurement refines that in eq. (2) introducing a sort of quantization on the approximated adjacency matrix $\tilde{A}^n$.

The final loss function minimized in the training phase can be then composed as

$$L_f = \sum_n \left(\tilde{A}^n, A^n\right) + \lambda_1 L_b\left(\tilde{A}^n\right) + \lambda_2 L_c\left(\tilde{A}^n, A^n\right). \tag{5}$$

where λ_1, λ_2 are two weighting constants that were estimated maximizing the final accuracy of the approach on a validation set of data.

4. EXPERIMENTAL RESULTS

In order to train and evaluate the efficiency of the proposed approach, we prepared different synthetic databases of samples (D^n, A^n) made of sets of near-duplicate images I_k, $k = 0, \ldots, N-1$. Each set was created starting from an original uncompressed image from UCID dataset [24], which was edited and compressed: this was the root of the phylogenetic

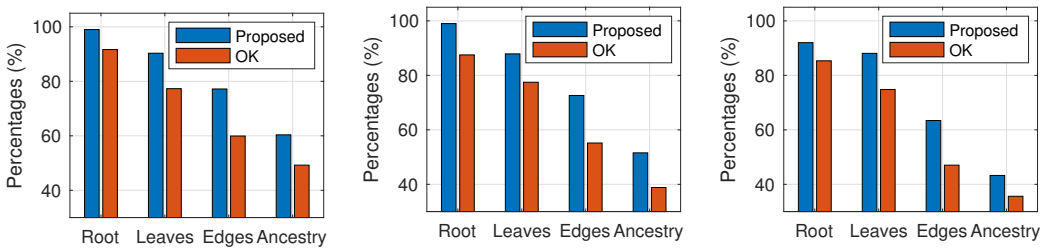

Fig. 3. RELA metric results on datasets $\mathcal{D}_T^8$ (a), $\mathcal{D}_T^{16}$ (b), and $\mathcal{D}_T^{32}$ (c) for the proposed strategy and Oriented Kruskal algorithm.

tree to be reconstructed. Other images were created recursively by selecting one random image among those altready-included in the set and applying a random set of transformations (chosen among those reported in Table 1). The transformation parameter were randomly chosen within the reported range, and each image was generated applying up to 4 operations (the number of editing step is random as well).

For every original image we generated 4 different near-duplicate sets; after this, image indexes were scrambled in order to avoid systematic structures in the dissimilarity matrices (i.e., we do not want the root to always be I_0). This lead to the creation of a dataset of 20000 pairs $\mathcal{D}^N = \{(D^n, A^n)\}$. The operation was repeated with $N = 8, 16, 32$ leading to matrices with size 8×8, 16×16, and 32×32. Dataset $\mathcal{D}^N$ was then divided into three subsets that included matrices obtained from disjoint subsets of the original UCID images. The first subset $\mathcal{D}_L^N$ consists in 7140 pairs and was used in the training operations; the second $\mathcal{D}_V^N$ was made of 7140 couples and was used in validation; the remaining pairs were included in the test set $\mathcal{D}_T^N$. Training was performed using the Adam optimizer with patience equal to 10 epochs and Δloss equal to 0.0001. The parameters $\lambda_1 = \lambda_2 = 0.01$ were set maximizing RELA metrics [9] on the validation set.

The accuracy of the proposed solution was evaluated using the RELA metrics reported in [9], which parameterize the percentages of correctly identified roots, edges, leaves and ancestry relations in the reconstructed phylogenetic tree. Fig. 3 reports the reconstruction results on datasets D_T^8, D_T^{16}, and D_T^{32}. The proposed solution is compared with Oriented Kruskal (OK) strategy adopted in [9]. It is possible to notice that the accuracy in identifying the root node increases of about 5 %, but the real improvements are to be found on the other metrics (increment up to 15 %). This was possible since most of the dissimilarity noise has been removed by comparing multiple edge weights together and imposing reconstruction coherence thanks to the composition of metrics L_a, L_b, and L_c. Table 2(a) reports the accuracy obtained on $\mathcal{D}_T^8$ after training the network with different loss functions.

Final tests verified the performance of the proposed approach whenever the analyzed set is missing some of the near-duplicates. This is a very likely situation in a real set-up since it is possible that the forensic analyst was not able to retrieve all the elements that form the phylogenetic tree. In this

Metric	**Final loss**	**♯epochs**	**Root**	**Leaves**	**Edges**	**Ancestry**
L_a	0.025	25	98.00	87.28	74.81	56.14
$L_a + \lambda_1 L_b$	0.035	22	98.67	88.75	76.24	58.36
L_f	0.033	25	99.00	90.32	77.19	60.36

a

Algorithm	**Dataset**	**Root**	**Leaves**	**Edges**	**Ancestry**
Proposed	$\mathcal{D}_S^8$	85.00	80.68	48.10	36.76
Proposed	$\mathcal{D}_S^{16}$	82.23	79.84	39.28	28.25

b

Table 2. Performances with different loss functions (a) and different test sets (b). RELA metrics are reported in percentages.

case, the reconstructed tree is an approximation of the real one. In our tests, we decimated datasets $\mathcal{D}_T^{16}$ and $\mathcal{D}_T^{32}$ with 50 % random loss percentage reducing the original 16×16 and 32×32 matrices to 8×8 and 16×16, respectively. In this way, datasets $\mathcal{D}_S^8$ and $\mathcal{D}_S^{16}$ were obtained and classified using the networks that were trained on $\mathcal{D}_L^8$ and $\mathcal{D}_L^{16}$ (no re-training). The obtained results are reported in Table 2(b). It is possible to notice that, despite some loss due to the increased noise level, the network is still able to reconstruct correctly the tree. The trained autoencoders can be used to estimate the underlying phylogenetic tree with a lower amount of nodes: it is necessary to introduce some dummy nodes and their relative entries in the dissimilarity matrix so that dissimilarity weights' order is not altered.

5. CONCLUSIONS

The paper presented a denoising convolutional autoencoder to estimate the minimum spanning tree underlying a complete noisy graph. The proposed architecture was used to reconstruct the phylogenetic tree that describes the transformation history for a set of near-duplicate images shared on the Internet. Nevertheless, the approach is quite general and can be applied to other noisy MST estimation problems. Performances on the phylogenetic tree were evaluated measuring the percentage of correctly-detected roots, leaves, edges and ancestry relations from a dataset of synthetically-generated near-duplicate images; results showed that the proposed solution outperforms previous MST approaches for different graph sizes and whenever some contents of the tree are missing.

6. REFERENCES

[1] Z. Dias, A. Rocha, and S. Goldenstein, "First steps toward image phylogeny," in *Proc. of 2010 IEEE WIFS*, Dec. 2010, pp. 1 –6.

[2] A. De Rosa, F. Uccheddu, A. Costanzo, A. Piva, and M. Barni, "Exploring image dependencies: a new challenge in image forensics," in *Proceeding of SPIE*, 2010, pp. X1 – X12.

[3] F. O. Costa, S. Lameri, P. Bestagini, Z. Dias, A. Rocha, M. Tagliasacchi, and S. Tubaro, "Phylogeny reconstruction for misaligned and compressed video sequences," in *Proc. of IEEE ICIP 2015*, Sep. 2015, pp. 301–305.

[4] N. Le Philippe, W. Puech, and C. Fiorio, "Phylogeny of jpeg images by ancestor estimation using missing markers on image pairs," in *International Conference on Image Processing Theory, Tools and Applications (IPTA)*, Dec. 2016, pp. 1–6.

[5] S. Verde, S. Milani, and G. Calvagno, "Phylogenetic analysis of software using cache miss statistics," in *Proc. of IEEE ICASSP 2019*, May 2019, pp. 2552–2556.

[6] A. Gronskiy and J. M. Buhmann, "How informative are minimum spanning tree algorithms?," in *2014 IEEE International Symposium on Information Theory*, June 2014, pp. 2277–2281.

[7] D. Moreira, A. Bharati, J. Brogan, A. Pinto, M. Parowski, K. W. Bowyer, P. J. Flynn, A. Rocha, and W. J. Scheirer, "Image provenance analysis at scale," *IEEE Transactions on Image Processing*, vol. 27, no. 12, pp. 6109–6123, Dec 2018.

[8] M. Nucci, M. Tagliasacchi, and S. Tubaro, "A phylogenetic analysis of near-duplicate audio tracks," in *Proc. of IEEE MMSP 2013*, Sep. 2013, pp. 099–104.

[9] Z. Dias, A. Rocha, and S. Goldenstein, "Video phylogeny: Recovering near-duplicate video relationships," in *Proc. of IEEE WIFS 2011*, dec. 2011.

[10] S. Milani, P. Bestagini, and S. Tubaro, "Phylogenetic analysis of near-duplicate and semantically-similar images using viewpoint localization," in *Proc. of IEEE WIFS 2016*, Dec 2016, pp. 1–6.

[11] S. Verde, S. Milani, P. Bestagini, and S. Tubaro, "Audio phylogenetic analysis using geometric transforms," in *Proc. of IEEE WIFS 2017*, Dec 2017, pp. 1–6.

[12] S. Verde, N. Pretto, S. Milani, and S. Canazza, "Stay true to the sound of history: Philology, phylogenetics and information engineering in musicology," *Applied Sciences*, vol. 8, no. 2, 2018.

[13] S. Lameri, P. Bestagini, A. Mellon, S. Milani, A. Rocha, M. Tagliasacchi, and S. Tubaro, "Who is my parent? reconstructing video sequences from partially matching shots," in *Proc. of IEEE ICIP*, 2014.

[14] B. Shen, C. W. Forstall, A. D. R. Rocha, and W. J. Scheirer, "Practical text phylogeny for real-world settings," *IEEE Access*, pp. 41002–41012, 2018.

[15] A. Melloni, P. Bestagini, S. Milani, M. Tagliasacchi, A. Rocha, and S. Tubaro, "Image phylogeny through dissimilarity metrics fusion," in *Proc. of EUVIP*, 2014.

[16] Paul Riot, Andrés Almansa, Yann Gousseau, and Florence Tupin, "A correlation-based dissimilarity measure for noisy patches," in *Proc. of SSVM 2017, Kolding, Denmark, June 4-8, 2017*, 2017, pp. 184–195.

[17] F. Costa, A. Oliveira, P. Ferrara, Z. Dias, S. Goldenstein, and A. Rocha, "New dissimilarity measures for image phylogeny reconstruction," *Pattern Analysis and Applications*, vol. 20, 03 2017.

[18] Charles Deledalle, Loic Denis, and Florence Tupin, "How to compare noisy patches? patch similarity beyond gaussian noise," *International Journal of Computer Vision*, vol. 99, pp. 1–17, 08 2012.

[19] S. Milani, M. Fontana, P. Bestagini, and S. Tubaro, "Phylogenetic analysis of near-duplicate images using processing age metrics," in *Proc. of 2016 IEEE ICASSP*, 2016, pp. 2054–2058.

[20] S. Milani, P. Bestagini, and S. Tubaro, "Video phylogeny tree reconstruction using aging measures," in *Proc. of EUSIPCO 2017*, 2017, pp. 2181–2185.

[21] Sebastiano Verde, Simone Milani, and Giancarlo Calvagno, "Phylogenetic analysis of multimedia codec software," *2018 26th European Signal Processing Conference (EUSIPCO)*, pp. 1427–1431, 2018.

[22] T. Yuan, W. Deng, J. Tang, Y. Tang, and B. Chen, "Signal-to-noise ratio: A robust distance metric for deep metric learning," in *Proc. of IEEE CVPR 2019*, June 2019.

[23] Olaf Ronneberger, Philipp Fischer, and Thomas Brox, "U-net: Convolutional networks for biomedical image segmentation," *ArXiv*, vol. abs/1505.04597, 2015.

[24] Gerald Schaefer and Michal Stich, "UCID: An uncompressed color image database," 01 2004, vol. 5307, pp. 472–480.