JIGSAW PUZZLE SOLVING USING LOCAL FEATURE CO-OCCURRENCES IN DEEP NEURAL NETWORKS

Marie-Morgane Paumard[1], *David Picard*[1,2], *Hedi Tabia*[1]

[1]ETIS, UMR 8051, Université Paris Seine, Université Cergy-Pontoise, ENSEA, CNRS
[2]Sorbonne Université, CNRS, Laboratoire d'Informatique de Paris 6, LIP6, F-75005 Paris

ABSTRACT

Archaeologists are in dire need of automated object reconstruction methods. Fragments reassembly is close to puzzle problems, which may be solved by computer vision algorithms. As they are often beaten on most image related tasks by deep learning algorithms, we study a classification method that can solve jigsaw puzzles. In this paper, we focus on classifying the relative position: given a couple of fragments, we compute their local relation (e.g. on top). We propose several enhancements over the state of the art in this domain, which is outperformed by our method by 25%. We propose an original dataset composed of pictures from the Metropolitan Museum of Art. We propose a greedy reconstruction method based on the predicted relative positions.

Index Terms— Cultural heritage, fragment reassembly, jigsaw puzzle, image classification, deep learning.

1. INTRODUCTION

The reconstruction of pieces of art from shards is a time-consuming task for archaeologists. Close to puzzle solving problems, it may be automated. On the one hand, the computer vision algorithms struggle with those tasks. The dataset has to be scrupulously annotated by experts to reach a plausible solution, especially when the object fragments are lost, degraded or mixed among non-relevant fragments. Even so, the false-positive rate is still significant. On the other hand, deep learning algorithms are seen as an promising alternative, as they surpass other methods in most image-related tasks.

In this paper, we consider the image reassembly, which can be seen as solving a jigsaw puzzle: given two image fragments, we want to predict the relative position of the second fragment with respect to the first one. To solve this task, we investigate the setup proposed by Doersch et al. [1]. Given an image, we extract same-size squared 2D-tiles in a randomized grid pattern (see Figure 1). Visual features are then extracted from both fragments using a Convolutional Neural Network (CNN). These features are the combined and fed to a classifier in order to predict the correct relative position.

This work is supported by the Fondation des sciences du patrimoine, LabEx PATRIMA ANR-10-LABX-0094-01

However, the authors of [1] are not interested in solving the jigsaw puzzle problem *per se*, but merely in using it as a pretraining of generic visual features.

In order to solve the jigsaw puzzle, we extend the work of Doersch et al. [1]. Our contributions are the following: First, we propose a simpler, yet more effective, CNN architecture to extract visual features. Then, we propose a new combination scheme based on the Kronecker product that is able to better take into account correlations between localized parts of the fragments. We also propose a new dataset more closely related to cultural heritage puzzle solving tasks, which consists of 14,000 images from the Metropolitan Museum of Art (MET) archives. With our contributions, we obtain state of the art results on both the original ImageNet dataset proposed in [1] and our new MET dataset.

The remaining of this paper is organized as follows: in the next section, we present related work and contextual information on puzzle solving and fragment reassembly. Then, we detail our method in Section 3. Finally, we describe our new dataset and we examine our experimental results in Section 4.

Fig. 1. Example of fragments extracted using the randomized grid pattern on the MET dataset. Labels are the classes of the relative position with respect to the central fragment.

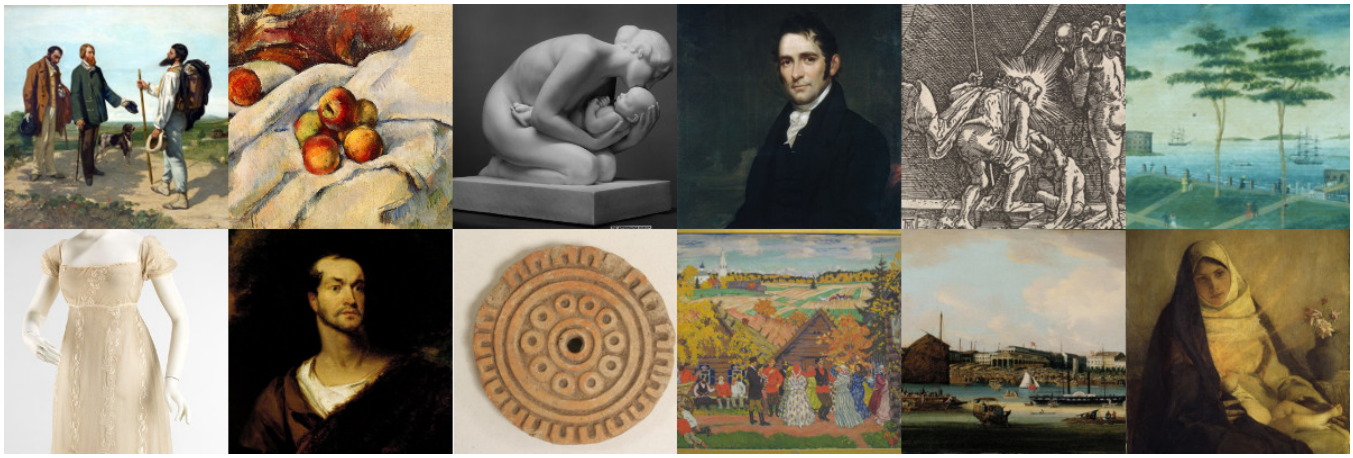

Fig. 2. Examples of images from the MET dataset.

2. RELATED WORK

Whether it comes to precisely align fragments or approximate a relative position, archaeological object reconstruction attracts numerous researchers, as shown by Rasheed and Nordin in [2, 3]. When most reassembly work is based on semi-automated methods, some stands out by the use of automated reconstruction, such as [4, 5, 6, 7]. These are mainly inspired by the research on puzzle solving, especially on jigsaw puzzles [8, 9, 10, 11]. Those methods study missing fragments or various-sized tiles, belonging to annotated datasets. They perform well on small datasets but poorly when the fragments are mixed from similar sources. Moreover, these methods are often very slow.

Since 2015, the deep learning community uses puzzle solving as a pretext task, proposing a reasonable alternative to data labeling. In [1], Doersch et al. introduce puzzle solving as a pretraining task for objects classification and detection. Their algorithm outperforms other unsupervised methods, which illustrates that CNN are able to learn object representations from the relative positions of the image part. Based on [1], Noroozi and Favaro [12] propose a network that observes all the nine tiles arranged in a grid pattern to obtain a precise object representation. They claim that the ambiguities may be wiped out more effectively when all fragments are examined. However, this leads to a much more complex classification due to the huge number of fragment orderings.

In our case, as we are not interested in building generic image features, but in solving approximatively the jigsaw puzzle itself. As such, the setup of [12] is impractical as it requires all fragments in order to make a prediction. This is unrealistic in cultural heritage where missing fragments are very common. However, the correlations between localized parts of the fragments (e.g., the correlation between the right part of the central fragment and the left part of the right fragment) are not taken into account in [1, 12], whereas we argue this information is essential to successful classification. This is what we investigate in our method.

3. PROPOSED METHOD

In this section, we detail our proposed method. We start by presenting the Feature Extraction Network that is shared between the two fragments. Then, we describe our propositions to combine the features of both fragments leading to the classification stage. Finally, we present a greedy algorithm to solve the jigsaw puzzle given any number of fragments.

3.1. Feature Extraction Network architecture

We use a CNN to compute the features associated with the fragments. Each fragment of size 96×96 pixel is processed such that pixels values vary between -1 and 1. Our architecture takes inspiration from VGG [13] and is composed of a sequence of 3×3 convolutions followed by batch-normalizations [14], ReLU activations [15] and max-poolings. The full architecture is shown on Table 1.

Our Feature Extraction Network ends with a fully connected layer that allows preserving the spatial layout of the input fragment. Although it is more costly than the worse popular global pooling layers [16], we conjecture that keeping the layout is essential to successfully predict the relative position of fragments.

3.2. Combination Layer

To predict the relative position of a fragment with respect to another one, we extract features for both fragments using the same feature extraction network. These two features are then combined using a combination layer and processed by a neural network as shown in Figure 3.

In [1], the authors propose to concatenate both features in the combination layer. This leads the subsequent fully con-

Layer	Shape	# parameters
Input	$96 \times 96 \times 3$	0
Conv+BN+ReLU	$96 \times 96 \times 32$	1k
Maxpooling	$48 \times 48 \times 32$	-
Conv+BN+ReLU	$48 \times 48 \times 64$	19k
Maxpooling	$24 \times 24 \times 64$	-
Conv+BN+ReLU	$24 \times 24 \times 128$	74k
Maxpooling	$12 \times 12 \times 128$	-
Conv+BN+ReLU	$12 \times 12 \times 256$	296k
Maxpooling	$6 \times 6 \times 256$	-
Conv+BN+ReLU	$6 \times 6 \times 512$	1.2M
Maxpooling	$3 \times 3 \times 512$	-
Fully Connected+BN	512	2.4M

Table 1. Architecture of the Feature Extraction Network. Conv: 3×3 convolution, BN: Batch-Normalization, ReLU: ReLU activation.

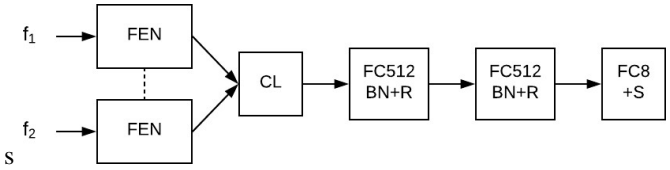

Fig. 3. Full network architecture. FEN: Feature Extractor Network. CL Combination Layer. FC: Fully Connected. BN: Batch-Normalization. R: ReLU activation. S: Softmax activation.

nected layer to perform a linear combination of the fragment features:

$$\forall i, y_i = \sum_m \alpha_{i,m}\phi_m(f_1) + \sum_m \beta_{i,m}\phi_m(f_2)$$

where y_i is the output of the neuron i in the fully connected layer, $\phi(f_1)$ is the feature of the first fragment (respectively, $\phi(f_2)$ for the second fragment) and $\alpha_{i,m}, \beta_{i,m}$ are the weights of neuron i.

Such a linear combination does not highlight co-occurrences of features, although it can be argued that subsequent layers with non-linearities may eventually be able to achieve similar results.

To circumvent these problems, we propose to use the Kronecker product in the combination layer. The output of the fully connected layer is then:

$$\forall i, y_i = \sum_{m,n} \alpha_{i,m,n}\phi_m(f_1)\phi_n(f_2),$$

with $\alpha_{i,m,n}$ being the weights of neuron i. The Kronecker product enables to explicitly model the co-occurrences between features of both fragments. This comes however at the cost of an increased number of parameters.

The output of the full network consists of a fully connected layer with k neurons followed by a Softmax activation, corresponding to the probabilities of the k possible relative locations. The full network (the Feature Extraction part and the Classification part) is trained at once using stochastic gradient descent on batches of fragments pairs.

3.3. Puzzle solving

In order to solve the puzzle problem, we consider the case where given a central fragment, we want to assign each of the remaining 8 fragments to its correct location. For each fragment, we compute the probabilities of assignment using the full network. This results in an 8×8 matrix where each row is a fragment and each column is a possible location. Solving the puzzle problem corresponds then to an assignment problem where we have to pick 8 values from the matrix (only one per row/column) such that their sum is maximized. We propose a greedy algorithm where we iteratively pick the maximum value and remove its corresponding row and column. Remark that this problem corresponds to a graph-cut problem for which much more involved algorithms exist. However, we found out that our greedy algorithm provides correct results in our case.

4. EXPERIMENTS

In this section, we first describe our new dataset and experimental setup. Then, we comment on the results comparing our approach to [1]. Finally, we give some qualitative results on puzzle solving using the greedy algorithm.

4.1. MET dataset and experimental setup

In order to be closer to our aimed application regarding cultural heritage puzzle solving, we propose a new dataset consisting of images from the Metropolitan Museum of Art. We collect 14,000 open-source images of paintings and pieces of art. Contrarily to ImageNet, the quality of the sensors taking the pictures allows avoiding the unbalanced colors distribution. Images from the MET dataset are shown in Figure 2.

We use 10k images for training and evaluate the performances on the remaining 4k. During training, each image from the training set is resized and square-cropped so that its size is 398×398 pixels. We divide it into 9 parts separated by a 48-pixels gap, mimicking an erosion of the fragments. This value was the one used by Doersch et al. in [1]. Then, we extract the center fragment and one of its 8 neighbors. Each fragment is of size 96×96 pixels, and we randomly move the location of the fragment by $\pm$7pixels in each direction. The learning rate is 0.1 For the validation, we only consider a single pair of random fragments per image.

In the evaluation, we consider the following three setups: 1) we train the neural network on ImageNet and evaluate it on ImageNet ; 2) we train the network on ImageNet, fine-tune it on MET and evaluate it on MET (transfer setup) ; 3) we

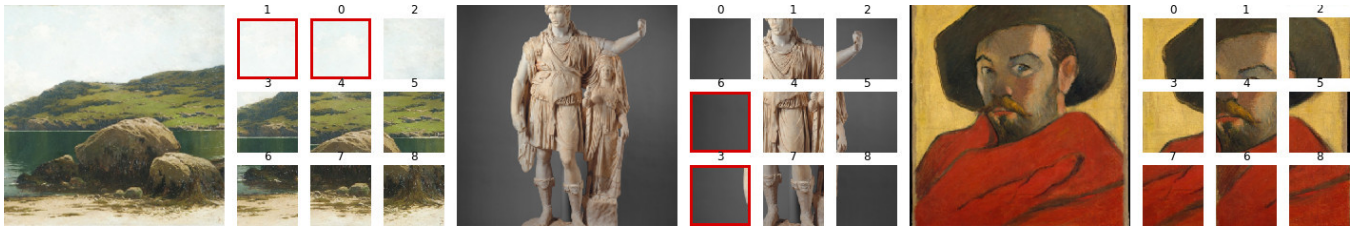

Fig. 4. Examples of reconstructions using the greedy algorithm on images taken from the MET dataset. The red outlined fragments are misplaced.

train the network from scratch on MET (MET setup). The results are compared using the accuracy of correct location prediction.

4.2. Results

On Figure 5, we show the evolution of the validation accuracy on ImageNet for a network similar to that of [1], compared to our proposed architecture using either the concatenation or the Kronecker combination layer. Our implementation of the network proposed in [1] has fewer parameters (we reduced the fully connected layers from 4k to 512 neurons) but nonetheless outperforms what is reported in [1], achieving 57% accuracy on ImageNet compared to the 40% reported in the paper. Our proposed architecture significantly outperforms that of [1] by a 25% margin. In accordance with our intuition, the Kronecker combination consistently outperforms the concatenation combination, reaching a validation accuracy of 65% after 100 epochs. All the results were obtained through a single run.

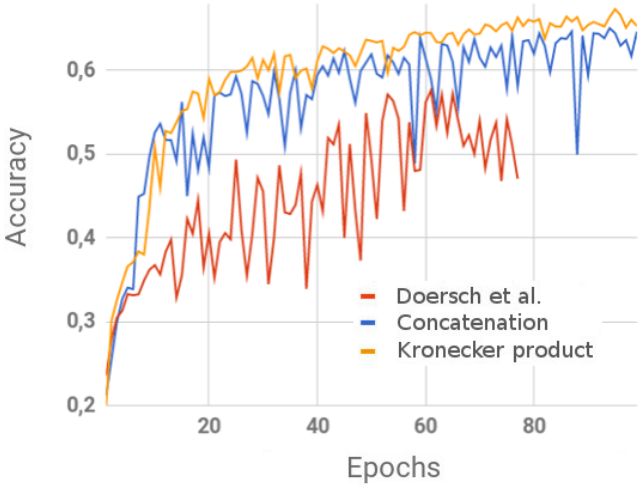

Fig. 5. Evolution of the validation accuracy on ImageNet.

We show on Table 2 the validation accuracy of the MET dataset for both Combination Layers and comparing the MET setup to the transfer setup (training on ImageNet, validation on MET). As we can see, training on ImageNet followed by a fine tuning on MET provides better results than training on MET alone. In this setup, we also remark that the Kronecker combination performs significantly better than the concatenation layer, which confirms the soundness of the approach.

	ImageNet → MET		MET setup	
	concat	kron	concat	kron
Accuracy	59.7%	64.9%	48.9%	47.9%

Table 2. Comparison between the transfer setup (ImageNet → MET) and the MET setup for various combination layers.

Finally, we show on Figure 4 examples of puzzle solving using the greedy algorithm on several images taken from the MET dataset. As we can see, most of the predictions are correct. In the case where the algorithm wrongly predicts the position of the fragments, we can see that the corresponding fragments are visually close to what is expected to be in that location (*e.g.*, sky and could pattern in the first example).

Using this greedy algorithm, we are able to solve the jigsaw puzzle perfectly 28.8% of the time. The proportion of correctly placed fragments is 68.8%, which means that on average only 2 fragments are swapped per image which is consistent with the accuracy at predicting individual positions of our neural network.

5. CONCLUSION

We proposed a robust deep learning method to classify the position of two neighboring fragments, which outperforms the state of the art by 25%. We successfully apply it to solve 9-tiles puzzles, and we show promising results on a new proposed dataset composed of images taken from the Metropolitan Museum of Art.

In the future, we plan to add a ninth class describing the not-neighbor relationship. Such class will allow us to solve more challenging puzzles. We are extending our method to non-squared fragments with irregularities.

6. REFERENCES

[1] C. Doersch, A. Gupta, and A.A. Efros, "Unsupervised visual representation learning by context prediction," ICCV, 2015.

[2] N.A. Rasheed and Nordin M.J., "A survey of computer methods in reconstruction of 3d archaeological pottery objects," ISSN, 2015, vol. 3, pp. 712–724.

[3] N.A. Rasheed and Nordin M.J., "A survey of classification and reconstruction methods for the 2d archaeological objects," ISTMET, August 2015, pp. 142–147.

[4] J.C. McBride and B.B. Kimia, "Archaeological fragment reconstruction using curve-matching," CVPRW, 2003.

[5] F. Jampy, A. Hostein, E. Fauvet, O. Laligant, and F. Truchetet, "3d puzzle reconstruction for archeological fragments," 3DIPM, 2015.

[6] L. Zhu, Z. Zhou, J. Zhang, and D. Hu, "A partial curve matching method for automatic reassembly of 2d fragments," *ICI*, vol. LNCIS 345, pp. 645–650, 2006.

[7] F. Stanco, D. Tanasi, and G. Gallo, "Virtual restoration of fragmented glass plate photographs of archaeological repertoires," vol. 2, pp. 141–144, 2011.

[8] Z. Hammoudeh and C. Pollett, "Clustering-based, fully automated mixed-bag jigsaw puzzle solving," in *Computer Analysis of Images and Patterns*, M. Felsberg, A. Heyden, and N. Krüger, Eds. 2017, pp. 205–217, Springer International Publishing.

[9] F.A. Andaló, G. Taubin, and S. Goldenstein, "Psqp: Puzzle solving by quadratic programming," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, pp. 385–396, Feb 2017.

[10] C. Lifang, D. Cao, and Y. Liu, "A new intelligent jigsaw puzzle algorithm base on mixed similarity and symbol matrix," *IJPRAI*, vol. 32, 2018.

[11] S. Gur and O. Ben-Shahar, "From square pieces to brick walls: The next challenge in solving jigsaw puzzles," ICCV, 2017.

[12] M. Noroozi and P. Favaro, "Unsupervised learning of visual representations by solving jigsaw puzzles," 2015.

[13] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," ILSVRC, 2014.

[14] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," ICML, 2015.

[15] V. Nair and G.C. Hinton, "Rectified linear units improve restricted boltzmann machines," ICML, 2010.

[16] M. Lin, Q. Chen, and S. Yan, "Network in network," ICLR, 2014.