

L'apprentissage profond pour le réassemblage d'images patrimoniales

Marie-Morgane PAUMARD, David PICARD, Hedi TABIA

ETIS, UMR 8051, Université Paris Seine, Université Cergy-Pontoise, ENSEA, CNRS
6 Avenue du Ponceau, 95000 Cergy, France

marie-morgane.paumard@ensea.fr, picard@ensea.fr, hedi.tabia@ensea.fr

Résumé – Dans cet article, nous présentons Deepzzle, une méthode de résolution de puzzles reposant sur l'apprentissage automatique, adaptée au réassemblage d'objets patrimoniaux. En effet, ceux-ci subissent des contraintes particulières : érosion, fragments manquants, morceaux mélangés entre objets, etc. Deepzzle apporte une solution à ces problèmes.

Abstract – In this paper, we present Deepzzle, a puzzle-solving method based on deep learning. Deepzzle is able to handle efficiently cultural heritage constraints: heritage collections suffer from erosion, missing fragments and mixed puzzles.

1 Introduction

Les collections muséales regorgent de fragments d'objets anciens, que les archéologues s'efforcent de réassembler. Les fragments préservés sont souvent très abîmés, et de formes très variées. Devant l'ampleur de la tâche, la résolution automatique de puzzle apparaît comme une solution nécessaire, qui trouve des applications aussi bien en archéologie, en biologie du génome ou encore en anthropologie judiciaire.

Les méthodes d'apprentissage profond ont rencontré un franc succès pour la résolution de problèmes de vision par ordinateur classiques. Elles sont peu à peu appliquées à des tâches moins usuelles, comme la résolution automatique de puzzles (Figure 1). La plupart des travaux se concentrent sur les problèmes à 9 pièces. On utilise alors un réseau de neurones pour prédire un réassemblage imprécis, complété par la suite.



FIGURE 1 – Exemple de puzzle à 9 pièces. À partir d'un jeu de fragment (à gauche), l'algorithme prédit un réassemblage global. Pour faciliter la compréhension du résultat, nous affichons en transparence l'image originale et plaçons précisément les fragments d'après les positions relatives prédites (à droite)

La résolution de puzzle s'effectue en deux étapes : la préparation et l'assemblage. La préparation consiste le plus souvent

à se doter de métriques d'évaluation permettant de comparer deux fragments et d'en déduire la position relative. L'assemblage utilise ces métriques pour résoudre le puzzle.

Dans cet article, nous présentons Deepzzle, une méthode de résolution de puzzles issus d'images patrimoniales. Deepzzle est adapté aux puzzles ayant subi une très forte érosion, à ceux pour lesquels on a égaré des pièces, ainsi qu'à ceux ayant été mélangés avec d'autres puzzles. Après avoir détaillé les méthodes usuelles de réassemblage, nous explicitons notre méthode. En section 3, nous présentons notre protocole expérimental et nous analysons les résultats obtenus.

2 La résolution de puzzles

Les méthodes mathématiques, basées sur les contours et les similitudes des fragments, permettent de résoudre la plupart des cas d'usage (érosion, fragments manquants, formes variées, etc.), sans toutefois parvenir à traiter les problèmes présentant plusieurs de ces caractéristiques. Dès lors, l'apprentissage profond semble être la solution idéale pour se départir de l'explosion combinatoire. Nous présentons les trois principales méthodes de résolution.

2.1 Méthodes mathématiques

Pour résoudre des puzzles, les méthodes plébiscitées s'appuient sur les contours des fragments ou sur leurs contenus. Les résolutions par le contour sont efficaces pour les puzzles 3D et ceux aux formes les plus diverses. La plupart du temps, l'algorithme identifie les cassures et apparie celles qui sont complémentaires. L'assemblage est calculé par un solveur SAT [1] ou grâce à des graphes [2].

Les méthodes exploitant les contenus des fragments sont adaptées aux cas d'érosions, aux pièces manquantes et aux très grands

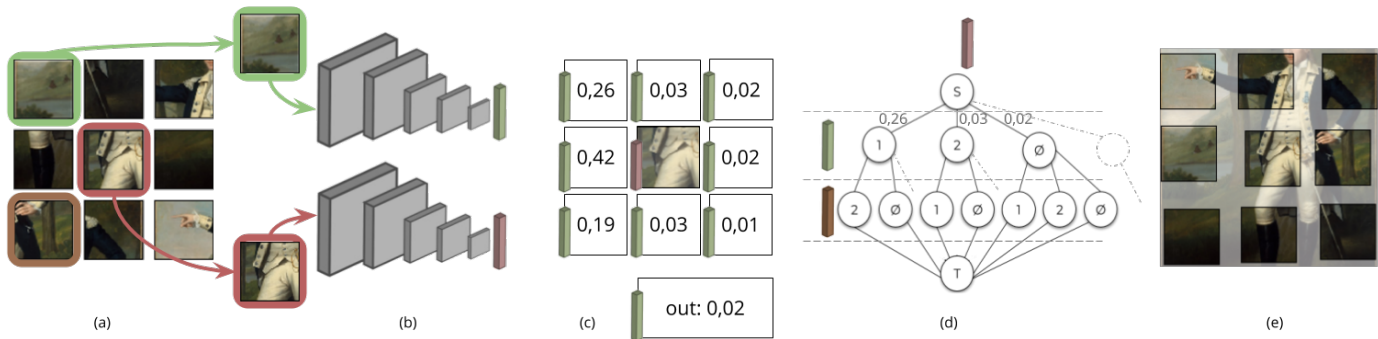


FIGURE 2 – Deepzzle : les caractéristiques extraites des fragments (b) permettent d’obtenir la classification de la position relative (c). Le plus court chemin dans un graphe est utilisé pour sélectionner le meilleur réassemblage (d)

puzzles. Souvent, la préparation permet d’associer les fragments deux à deux et l’assemblage est construit itérativement à partir des couples les plus prometteurs, en comparant les fragments à placer avec ceux déjà en place [3, 4].

Certaines méthodes mélangent ces deux stratégies [5], permettant de mélanger formes variées et érosion. La robustesse gagnée se fait au détriment du temps de calcul, et ces méthodes ne sont appliquées qu’à des puzzles de peu de fragments.

2.2 Apprentissage profond

Carl DOERSCH et al. [6] ont été les premiers à explorer l’usage de l’apprentissage profond sur des puzzles à 9 pièces. Ils ont montré qu’un réseau pré-entraîné à prédire la position relative des fragments obtenait d’excellents résultats sur les tâches usuelles de classification et de détection. Ils ont proposé une architecture à base de réseaux siamois, c’est-à-dire de réseaux identiques dont les poids sont liés. Leur méthode ne couvre que la phase préparatoire et ne produit pas d’assemblage.

Mehdi NOROOZI et al. [7] résolvent des puzzles à 9 pièces. Dans certains cas, une à deux pièces des puzzles a été remplacée par un fragment d’une autre image. Leur réseau compare tous les fragments simultanément, fusionnant phase préparatoire et assemblage en une étape. L’ensemble de permutations possibles produit un problème à 10^5 classes ; pour obtenir des résultats probants en quelques jours, ils interdisent beaucoup de combinaisons, faisant décroître le nombre de classes.

Une autre méthode de résolution par l’apprentissage profond est celle mise en œuvre dans [8]. Après avoir disposé aléatoirement les 9 fragments, le réseau de neurones propose des interventions, jusqu’à ce qu’il estime que le puzzle est résolu.

Les travaux de Carl DOERSCH et al. se prêtant à la résolution de puzzles à partir d’un nombre varié de fragments, nous proposons une méthode dérivée de leurs travaux.

2.3 Deepzzle

La méthode que nous utilisons compare le fragment central à un fragment latéral, et prédit sa position parmi 9 classes : les 8 positions adjacentes et la classe "fragment n’appartenant pas au

puzzle". Nous organisons ensuite ces probabilités sous la forme d’un arbre de décisions dont les feuilles correspondent aux réassemblages. Le meilleur réassemblage est celui qui maximise la probabilité jointe. Lorsque le fragment central n’est pas connu, l’algorithme calcule les meilleurs réassemblages, en considérant tour à tour que chaque fragment est central, puis sélectionne le meilleur réassemblage.

Nous utilisons un réseau siamois inspiré de VGG-16 pour extraire les caractéristiques de chaque couple de fragments latéral et central (Figure 2b). Pour obtenir de meilleurs résultats, nous fusionnons ces caractéristiques via le produit de Kronecker. Le vecteur produit traverse alors plusieurs couches « entièrement connectées » jusqu’à classification (Figure 2c). L’architecture est justifiée dans [9].

Suite à la classification, nous organisons les prédictions dans un graphe décrivant tous les réassemblages possibles. Nous cherchons à maximiser la probabilité jointe, ce qui revient à trouver le plus court chemin dans le graphe des réassemblages (Figure 2d). Chaque ligne du graphe correspond à un fragment, et chaque nœud est une position encore libre au moment de placer le fragment, sachant toutes les positions occupées lors des lignes antérieures. Cette méthode est présentée dans [10].

Nous nous intéressons désormais à la résolution de puzzles plus complexes, comme ceux avec des fragments manquants et/ou des fragments supplémentaires — c’est à dire provenant d’autres puzzles. La taille du graphe augmentant lorsque l’on ajoute des fragments, nous utilisons des coupes dans le graphe afin de ne pas étudier les branches pour lesquelles les positions prédites ont une probabilité inférieure à un certain seuil. Cela nous a permis de diviser notre temps de calcul d’un facteur 10^3 .

3 Expérimentations et résultats

Dans cette section, nous présentons les spécificités de notre protocole expérimental et les métriques utilisées. Nous explicitons ensuite les résultats obtenus.

Nous utilisons 12 000 images issues des collections du *Metropolitan Museum of Art*. Nous en sélectionnons aléatoirement 10 000 pour l’entraînement et 2 000 pour la validation.

3.1 Protocole expérimental

À chaque étape, nous générons des fragments de manière arbitraire, en les espaçant d'environ 50 % leur taille, modélisant l'érosion (que l'on peut observer sur l'image de droite de la Figure 1). Cet espacement prive le réseau des informations des bords, l'incitant à apprendre à partir des contenus des fragments. Dans le cas de fragments additionnels, ceux-ci sont sélectionnés arbitrairement au sein d'une seule autre image.

Pour étudier la qualité des réassemblages dans le cas des fragments manquants et additionnels, nous utilisons le réseau de [10], dont la précision est à 64,6 %. Nous disposons de deux métriques pour évaluer ses performances de Deepzlle à l'issue de l'assemblage. La première indique le pourcentage de fragments bien placés après réassemblage. La seconde est le nombre d'images « presque parfaitement » reconstruites. Nous privilégions cette métrique à celle du nombre d'images parfaitement reconstruites pour nous départir du biais principal de notre jeu de données : les arrières-plans homogènes (Figures 3b et 3c). Un puzzle est « presque parfaitement » résolu si seuls ses fragments d'arrière-plan sont mal placés.

3.2 Résultats

Les Tables 1 et 2 évaluent les performances de Deepzlle dans les cas de fragments manquants et/ou additionnels. S'il n'y a aucun fragment manquant ni supplémentaire, les chances d'obtenir la bonne résolution en plaçant les fragments au hasard sont de $2,4 \cdot 10^{-5}$; nous obtenons 24,7 % avec la deuxième métrique, pour 64,6 % de fragments bien placés. Ainsi, lorsque la résolution est erronée, il y a en moyenne 4 fragments mal placés par puzzle. De manière générale, nous obtenons 2 % de différence entre la résolution « presque parfaite » et la résolution parfaite. Cela indique que les erreurs liées à l'arrière-plan sont rares, ou que notre seuil pour reconnaître la similarité entre deux fragments ne couvre pas tous les cas.

Les Tables 1 et 2 nous permettent de tirer les conclusions suivantes : premièrement, plus on ajoute de fragments, plus le puzzle est difficile à résoudre. Deuxièmement, les meilleurs scores sont obtenus lorsque la plupart des fragments sont manquants. Avec sept fragments manquants, il s'agit alors de prédire la position d'un seul fragment, ce qui correspond à la sortie de notre réseau de neurones. Nous obtenons également de bons résultats quand nous avons toutes les pièces. En effet, quand il nous en manque quelques-unes, l'évaluation du meilleur réassemblage est compromise : un fragment qui obtient des résultats homogènes pour toutes les classes sera placé arbitrairement. Si nous avons disposé de plus de fragments, le placement des autres fragments aurait permis de lever l'ambiguïté.

La Figure 3 présente quelques-unes de nos résolutions avec fragments manquants et fragments issus d'une autre image. La Figure 3a est parfaitement réassemblée. La Figure 3b contient 4 erreurs. La Figure 3c illustre un puzzle « presque parfaitement » réassemblé. Le réassemblage de la Figure 3d met en exergue une erreur d'arrière-plan : les fragments de ciel ont été échangés, mais ils ne sont pas assez similaires pour que l'inver-

TABLE 1 – Pourcentage de fragments bien placés

		Nombre de fragments supplémentaires			
		0	1	2	3
Nombre de fragments manquants	0	64,6	62,8	60,9	60,3
	1	61,6	59,4	57,9	57,0
	2	61,1	57,8	55,7	55,4
	3	63,0	59,6	57,3	54,6
	4	66,9	62,0	58,3	56,0
	5	72,4	66,9	62,2	59,3
	6	80,0	73,5	67,5	62,0
	7	89,4	81,1	73,6	67,6

TABLE 2 – Pourcentage de puzzles « presque parfaitement » résolu (où 100% des fragments sont « presque » bien placés)

		Nombre de fragments supplémentaires			
		0	1	2	3
Nombre de fragments manquants	0	24,7	19,9	18,3	16,9
	1	20,8	12,9	11,3	11,0
	2	21,1	10,6	8,8	8,3
	3	22,6	12,0	9,8	6,5
	4	24,9	12,2	8,4	6,8
	5	31,1	16,6	10,9	8,3
	6	43,4	22,7	13,9	10,6
	7	64,0	33,7	21,0	13,0

sion soit acceptée par la deuxième métrique. En revanche, pour l'œil humain, le réassemblage semble correct.

Nous constatons que la similitude entre fragments et fragments ajoutés joue un rôle. Dans les Figures 3a et 3d, les fragments supplémentaires sont jaunes, couleur qui est absente du fragment central. En revanche, dans la Figure 3b, les fragments supplémentaires sont beiges, comme l'arrière-plan de l'image à reconstituer. Quand les fragments sont suffisamment similaires, Deepzlle est enclin à remplir les vides laissés par des fragments manquants avec des fragments extérieurs (3b). À l'inverse, quand il n'y a aucun fragment manquant, le risque d'erreurs liées aux fragments additionnels décroît (3d), ce qui correspond avec nos conclusions liées aux Tables.

4 Conclusion

Dans cet article, nous avons présenté une méthode de résolution de puzzle à 9 pièces subissant une forte érosion. Nous avons augmenté sa précision grâce à l'introduction d'une métrique supplémentaire et sa rapidité grâce aux coupes dans les graphes. Nous avons analysé ses performances pour des tâches complexes, telles que la résolution en cas de fragments manquants ou lorsque deux puzzles ont été mélangés ; pour autant, Deepzlle a réussi à placer correctement la plupart des fragments, ne faisant qu'une inversion de position en moyenne.

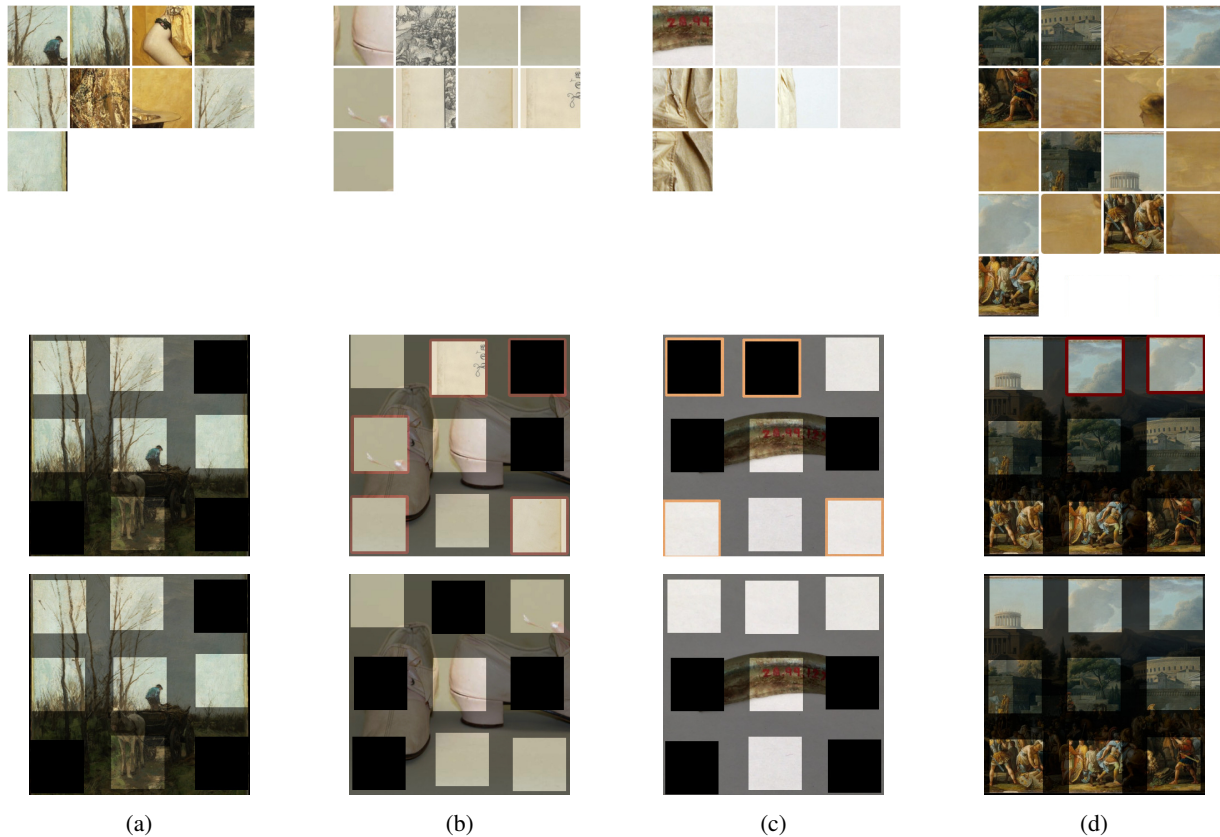


FIGURE 3 – Exemples de résolutions de puzzles. La première ligne contient les fragments proposés au réseau de neurones, le fragment central étant celui en haut à gauche. La deuxième ligne montre les résolutions de Deepzzzle. Les fragments bordés de rouge sont ceux ayant été mal placés ; ceux en orange sont « presque parfaitement » placés. La dernière ligne présente les solutions.

Remerciements

Ces travaux ont été réalisés dans le cadre du projet ARCHE-PUZ'3D financé par la Fondation des Sciences du Patrimoine (*EUR-17-EURE-0021*).

Références

- [1] G. Papaioannou, T. Schreck, A. Andreadis, P. Mavridis, R. Gregor, I. Sipiran et K. Vardis. *From Reassembly to Object Completion - A Complete Systems Pipeline*. ACM Journal of Computing and Cultural Heritage, vol. 10, n° 2, art. 8, 2017.
- [2] K. Zhang, Y. Wuyi M. Manhein, W. Waggenspack et X. Li. *3D Fragment Reassembly using Integrated Template Guidance and Fracture-Region Matching*. International Conference on Computer Vision, 2015, pages 2138–2146.
- [3] G. Paikin et A. Tal. *Solving Multiple Square Jigsaw Puzzles with Missing Pieces*. Conference on Computer Vision and Pattern Recognition, 2015.
- [4] K. Son, J. Hays et D. Cooper. *Solving Square Jigsaw Puzzles with Loop Constraints*. European Conference on Computer Vision, 2014.
- [5] K. Zhang et X. Li. *A graph-based optimization algorithm for fragmented image reassembly*. Graphical Models, vol. 76, n° 5, pages 484-495, 2014.
- [6] C. Doersch, A. Gupta et A.A. Efros. *Unsupervised Visual Representation Learning by Context Prediction*. International Conference on Computer Vision, 2015.
- [7] M. Noroozi, A. Vinjimoor, P. Favaro et H. Pirsiavash. *Boosting Self-Supervised Learning via Knowledge Transfer*. Conference on Computer Vision and Pattern Recognition, 2018.
- [8] R. Santa Cruz, B. Fernando, A. Cherian et S. Gould. *DeepPermNet : Visual Permutation Learning*. European Conference on Computer Vision, 2016.
- [9] M.-M. Paumard, D. Picard et H. Tabia. *Jigsaw Puzzle Solving Using Local Feature Co-Occurrences in Deep Neural Networks*. IEEE International Conference on Image Processing, 2018.
- [10] M.-M. Paumard, D. Picard et H. Tabia. *Image Reassembly Combining Deep Learning and Shortest Path Problem*. European Conference on Computer Vision, 2018.