Università
Ca'Foscari
Venezia

Corso di Dottorato di ricerca

in Computer Science

ciclo XXXIV

Tesi di Ricerca

Jigsaw Puzzle Solving as a Consistent Labeling Problem

SSD: INF/01

Coordinatore del Dottorato
ch. prof. Agostino Cortesi

Supervisore
ch. prof. Marcello Pelillo

Dottorando
Marina Khoroshiltseva
Matricola 956461

Jigsaw Puzzle Solving as a Consistent Labeling Problem

Marina Khoroshiltseva

June 3, 2022

Abstract

The thesis aims to develop a methodology for solving complex puzzle problems. In particular, we present a new formulation for general puzzle-solving tasks and then propose a further extension for a more challenging case with missing borders.

In the first part of the research, we explore the idea of abstracting the jigsaw puzzle problem as a consistent labeling problem, a classical concept for which a solid theory and powerful algorithms are available. A formal theory of consistency developed by Hummel and Zucker in the 1980s turned out to have intimate connections with non-cooperative game theory. The theory generalizes classical (boolean) constraint satisfaction problems to scenarios involving "soft" compatibility measures and probabilistic (as opposed to "hard") label assignments. The problem amounts to maximizing a well-known quadratic function over a probability space which we solve using relaxation labeling algorithms endowed with matrix balancing mechanisms to enforce one-to-one correspondence constraints. The preliminary experiments with square jigsaw puzzles demonstrate the feasibility of the proposed approach.

The second part addresses the problem of puzzles with eroded borders, a special challenging case of puzzle-solving. Solving puzzles with eroded borders is a common situation when dealing with the re-assembly of archaeological artifacts or ruined frescoes. In this particular condition, the puzzle's pieces do not align perfectly due to the erosion gaps; a direct matching of the patches is consequently unfeasible due to the lack of color and line continuations. To tackle this issue, we propose JiGAN, a GAN-based method for solving puzzles with ruined borders. JiGAN is a two-steps procedure: first, we repair the eroded borders with a GAN-based image extension model and measure the alignment affinity between pieces; then, we solve the puzzle with the relaxation labeling puzzle solver. The experiments on commonly used benchmark datasets demonstrate that our approach can address the problem of eroded borders and produce plausible reconstruction results.

Abstract

La tesi si propone di sviluppare una metodologia per la risoluzione di complessi problemi puzzle. Nella prima parte della ricerca, esploriamo l'idea di astrarre il problema del puzzle come un problema di consistent labeling, un concetto classico introdotto negli anni '80 da Hummel e Zucker per il quale esiste una solida teoria e potenti algoritmi. La formale teoria di consistenza sviluppata da Hummel e Zucker è in stretta connessione con la teoria dei giochi non cooperativi. La teoria generalizza i problemi di soddisfazione dei vincoli classici (booleani) in scenari che coinvolgono misure di compatibilità "soft" e assegnazioni di etichette probabilistiche (al contrario di "hard"). Il problema consiste nel massimizzare la funzione quadratica su uno spazio di probabilità; a tal fine si utilizza l'algoritmo di relaxation labeling dotato di meccanismo di bilanciamento delle matrici per imporre vincoli di corrispondenza uno-a-uno. La seconda parte affronta il caso speciale e impegnativo di risoluzione di puzzle in presenza di bordi erosi, situazione comune in archeologia (ricostruzione dei manufatti antichi, affreschi, ...). In questi particolari casi, i pezzi del puzzle non si allineano perfettamente a causa dell'erosione; un abbinamento diretto non è quindi praticabile per la mancanza di continuazione del colore. Per affrontare questo problema, proponiamo JiGAN, un metodo basato su GAN per risolvere i puzzle con bordi rovinati. Gli esperimenti sui set di dati di riferimento comunemente utilizzati dimostrano che il nostro approccio è in grado di affrontare il problema dei bordi erosi e produrre plausibili risultati di ricostruzione.

Acknowledgement

I would like to thank my academic supervisor Professor Marcello Pelillo for guiding me, supporting me, and encouraging me throughout this research project, and my adviser Professor Ohad Ben-Shahar for his expert advice, suggestion, and fruitful discussions along the way. I would also like to thank my colleagues - Ben, Alessandro, and Sebastiano - for their wonderful collaboration, it was nice to work with you all.

I would not have been able to complete this journey without my family for whose help I am endlessly grateful: my parents for their encouragement and belief in me; Luca, for his never-ending patience and kindness; and our kids, Sofia and Sebastiano, for having filled our lives with joy and love. I could not have done any of this without you. Thank you all.

Contents

1 Introduction **9**
- 1.1 Reassembly in Cultural Heritage . . . 9
- 1.2 Cultural Heritage and Machine learning . . . 12
- 1.3 Puzzle game . . . 12
- 1.4 Main contributions . . . 14
  - 1.4.1 Square jigsaw puzzle . . . 14
  - 1.4.2 Puzzle with eroded border . . . 15
- 1.5 Thesis structure . . . 16
- 1.6 Publications . . . 17

2 Machine learning methods and Cultural Heritage **19**
- 2.1 Datasets . . . 22
- 2.2 Unsupervised Learning in Cultural Heritage . . . 22
- 2.3 Supervised Learning (Deep Neural Networks) . . . 24
- 2.4 Semi-supervised Learning . . . 25
- 2.5 Conclusion remarks to Chapter 2 . . . 27

3 Puzzle-solving methods **29**
- 3.1 Solving puzzle with computational methods . . . 29
- 3.2 Solving puzzle with neural networks . . . 30
- 3.3 Solving puzzle with missing border . . . 31
- 3.4 Puzzle-solving task . . . 32
  - 3.4.1 Compatibility measure . . . 32
  - 3.4.2 Problem formulation . . . 35
- 3.5 Our methods . . . 36

4 Consistent Labeling Problem **37**
- 4.1 Nonlinear Relaxation Labeling . . . 38
- 4.2 The dynamics of nonlinear relaxation labelling process . . . 41

5 Solving puzzle as Consistent Labeling problem **43**
5.1 Preliminaries - terminology 44
5.1.1 Permutation matrix 44
5.1.2 Doubly stochastic matrix 44
5.1.3 Sinkhorn-Knopp algorithm 45
5.1.4 Alternating projection method 45
5.1.5 Hungarian Algorithm 46
5.2 Model 48
5.2.1 Compatibility measure 49
5.2.2 Puzzle Solver 51
5.2.3 Final reconstruction 57
5.3 Preliminary study 58
5.3.1 Oracle compatibility 58
5.3.2 Perturbation test 58
5.4 Experiments with Natural Images 59

6 Solving Jigsaw Puzzles with Eroded Boundaries **67**
6.1 Introduction 67
6.2 Generative Adversarial Net (GAN) 69
6.3 JiGAN Model for puzzle-solving 73
6.3.1 Border Extension 74
6.3.2 Compatibility of patches 74
6.3.3 Puzzle Solver 75
6.4 Experiments & Results 77
6.4.1 Datasets 77
6.4.2 Accuracy metrics 77
6.4.3 Experiments 77
6.5 Experiments with Fresco dataset 86

7 Conclusions **90**

List of Figures

1.1 Two crates of fragments 9
1.2 Reassembling a fractured head model 10
1.3 The Toreador Fresco 11
1.4 Example of jigsaw puzzle-solving game 13

2.1 I-Statistics on Cultural Heritage articles 20
2.2 II-Statistics on Cultural Heritage articles 21

3.1 Pairwise compatibility measure performance comparison . . . 34

5.1 Alternating projection algorithm 46
5.2 Pipeline of the algorithm. 48
5.3 Sparsification effect of the compatibility matrix 50
5.4 Relaxation Labeling Puzzle solver 52
5.5 Sinkhorn-Comparison of the behaviour of the algorithm . . . 54
5.6 Alternating projection Comparison 56
5.7 Hungarian algorithm-Approximation of final solution 57
5.8 Average local consistency with oracle compatibilities 59
5.9 Drop in accuracy due to perturbation of oracle compatibility . 60
5.10 I-Qualitative reconstruction performance 62
5.11 II-Qualitative reconstruction performance 64
5.12 I-Assignment matrix Comparison 65
5.13 II-Assignment matrix Comparison 66

6.1 Schematic architecture of GAN 70
6.2 Boundless Model Architecture 71
6.3 Pipeline of the JiGAN algorithm 73
6.4 Border extension procedure 75
6.5 JiGAN vs RL model 79
6.6 I-Qualitative results for small puzzles from Pacs dataset . . . 80
6.7 II-Qualitative results for small puzzles from Pacs dataset . . . 81

6.8 III-Qualitative results for small puzzles from Pacs dataset . . 82
6.9 I-Qualitative results for big puzzles 84
6.10 II-Qualitative results for big puzzles 85
6.11 II-Qualitative results for Fresco puzzles - 1 88
6.12 II-Qualitative results for Fresco puzzles - 2 89

List of Tables

5.1 Direct accuracy-MIT, McGill and Pomeranz805 datasets . . . 61
5.2 Neighbour accuracy-MIT, McGill and Pomeranz805 datasets . 63
5.3 Perfect accuracy-MIT, McGill and Pomeranz805 datasets . . 63

6.1 Direct accuracy-PACS datasets 78
6.2 Perfect accuracy-PACS datasets 78
6.3 JiGAN Direct accuracy-MIT, McGill and Pomeranz805 datasets 83
6.4 JiGAN Neighbour accuracy-MIT, McGill and Pomeranz805 datasets 83
6.5 JiGAN Direct accuracy fresco dataset 87
6.6 JiGAN Neighbour accuracy fresco dataset 87

Chapter 1

Introduction

1.1 Reassembly in Cultural Heritage

The main goal of the thesis is to develop the computational methodology for solving complex puzzle problems that can be applied to the virtual reconstruction of broken artifacts. In particular, our ambition is to build an assembling tool able to face situations where adjacency of all fragments is uncertain and the appearance (ground true) of an original artifact is unknown. The idea to work on puzzle-solving came due to its intimate relation to the problem of fresco reconstruction in archeology. In fact, the reconstruction of archeological ancient artifacts can be seen as a particularly challenging case of puzzle-solving (that makes the puzzle-solving task a part of the branch of computer vision for cultural heritage).

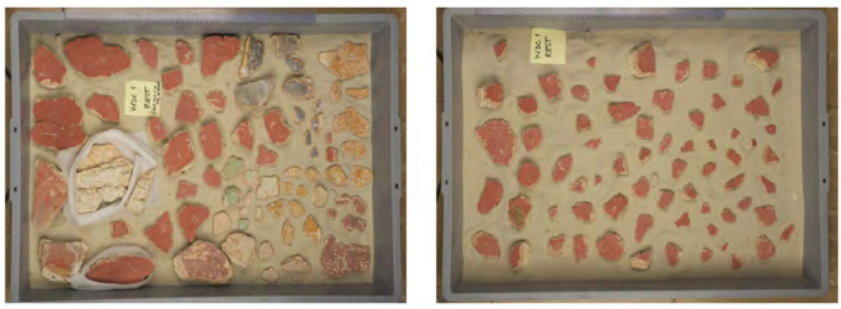

Figure 1.1: Two crates of fragments from the Tongeren Vrijthof excavation, containing 144 fragments. (Copyright Gallo Romeins Museum Tongeren).

Reconstruction of the ancient artifact is a problem of great interest in archeology and is important because helps archeologists to better understand past civilizations and cultures. The artifacts are often found in a fractured state and must be reassembled before being displayed and studied. Manual reconstruction is a labor-intensive and time-consuming job. Furthermore, in some cases, it is infeasible due to the elevated number of fragments or the fragility of objects (when the excessive handling of pieces is undesirable). For these reasons, a great amount of archeological material remains in storage un-examined and unstudied.

To overcome this problem, computer-aided technologies make it possible to digitize detailed shape, color, and surface information for each fragment, while computational algorithms can help match fragments and virtually reconstruct the objects, especially in situations where the space of potential "matches" is large. The problem essentially can be seen as enormous jigsaw puzzle-solving, where the pieces are often with ruined borders and faded colors, some pieces may be missing, and the shape of an original artifact is unknown.

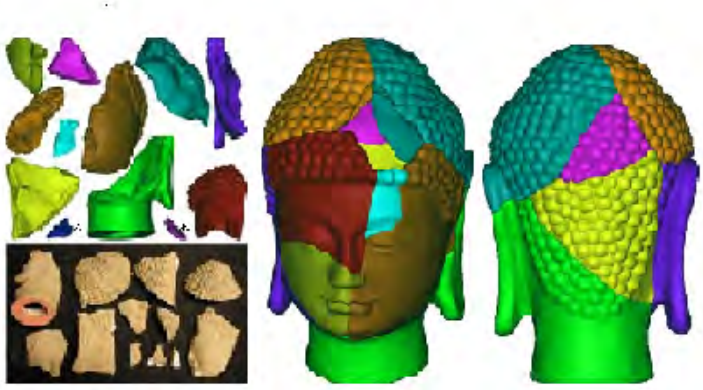

Figure 1.2: Reassembling a fractured head model [37]

.

The problem of reassembling archeological "puzzles" may include ruined frescoes, fractured pottery, ancient manuscripts, and other types of objects. The reassembly problem is extensively studied in the computer science community. The method of computational reassembly of an artifact generally consists of two main steps: local matching and global assembly [37]. The

local matching is based on finding matches between pairs of fragments or searching the most probable neighbors of a given fragment. Global assembly is looking for a complete solution to the reconstruction problem, taking into consideration pairwise match hypotheses.

Figure 1.3: The Toreador Fresco illustrates the Minoan fascination with bulls. Late Bronze Age 1550 1450 BC. Court of the Stone Spout, Knossos, Crete. (Photo by Werner Forman/Universal Images Group/Getty Images) The Minoan culture controlled Crete, an island in the Mediterranean Sea south of Greece from about 2000 to 1600 BCE. Theirs was a sophisticated and advanced society.

Various techniques are used for finding good matches between fragments, including comparisons of color, shape, texture. *Pictorial* methods are usually based on the colors close to the edges of the fragments or analyses the content of the entire fragment. *Geometric* methods focus on the contour of the fragments, or the surface area of the fragment edges; also, some properties

of the material [56] can be used for matching fragments. The main difficulty in global reassembly is a poor scaling of the problem with the number of fragments, due to the combinatorial explosion of potential solutions. For this reason, global assembly algorithms tend to use greedy approaches that iteratively grow and merge clusters while maximizing a scoring function.

The reassembly problem is often modeled as a graph, where nodes represent fragments and edges represent probable matches between pairs of fragments. In these methods, they try to consider as many matches as possible (to have more chances to find correct matches); the reassembly process consists in building a connection graph from candidate matches and finding the most (globally) consistent set of edges [1].

1.2 Cultural Heritage and Machine learning

Certainly, reassembly is not the only area in cultural heritage where Machine Learning (ML) might be engaged in resolving various challenging problems. Nevertheless, it seems that ML has still few applications in the fields of Cultural Heritage and the use of the modern computational technique is limited. To investigate this problem, we have conducted an overview of the application of the machine learning technique in cultural heritage. Most applications in the literature tend to rely on statistical toolboxes applied on small datasets that moreover, are usually not publicly available. The modern methods take time to become popular in the fields of cultural heritage and the limited access to a sufficient amount of data is an additional constraint on its growth.

Although machine learning is not widely used in cultural heritage, there are examples of applications in the literature where interesting results were obtained. In the first part of this research (Chapter 2), we present an overview of the machine learning methods applied to cultural heritage and confirm that plenty of cultural heritage issues can be solved with the use of machine learning techniques.

1.3 Puzzle game

The jigsaw puzzle is a well-known game where small (and often irregular) pieces must be fitted together to reconstruct the complete image or shape. Despite its entertaining and educational origins, solving a puzzle has numerous applications in different fields, such as image editing, shredded documents [19], genome biology [93], reconstruction of broken artifacts [23].

Figure 1.4: Example of jigsaw puzzle-solving game

Another meaning of the word puzzle is *"a complicated or mysterious problem that can only be solved or explained by connecting several pieces of information"*. In fact, the problem of the puzzle is demonstrated to be NP-complete. Nevertheless, the automatic puzzle-solving problem puzzles for years the minds of researchers in fields of computer science, mathematics, engineering, and numerous approaches have been proposed involving different computational schemes such as functional optimization, greedy algorithm, machine learning.

There is no standard puzzle-solving task as numerous variations of puzzle problems are discussed in the literature. These variations mainly differ in the fragment's quality and orientation, the presence of missing or irrelevant pieces, and knowledge about the final image shape. The simplest version of the puzzle-solving task, known as *square jigsaw puzzle*, is represented by a set of perfectly matching and oriented square pieces that must be re-assembled in a picture of a known shape. Such a puzzle-solving task can be considered as a starting point for more complex puzzle tasks, i.e. puzzle with missing fragments, ruined border, unknown orientation, and other characteristics typical of archaeological puzzles.

1.4 Main contributions

In this research, we first tackle the puzzle-solving problem in its simplest version and propose a new formulation of the problem. In the second part of this work, we address the more complex task of solving puzzles with eroded borders, which is a common case in archaeology.

1.4.1 Square jigsaw puzzle

The *square jigsaw puzzle* is the simplest version of puzzle-solving task, where the square pieces should be reordered on a 2D grid to form a coherent image. Formally, one should look for a permutation matrix that encodes such reordering and represents the correct solution of the puzzle. Although demonstrated to be NP-complete [20], the automatic puzzle-solving problem puzzles the minds of researchers in computer science, mathematics, and engineering for years. Numerous approaches tackled the problem, involving functional optimization [14, 2, 40], greedy algorithm [61, 30, 70, 74, 35], and machine learning [58, 10, 45].

In this research, we take a different route and explore the idea of abstracting the jigsaw puzzle problem as a *consistent labeling* problem, a class of problems widely studied in the computer vision and pattern recognition communities since the 1970s [65, 36]. Attempts at formalizing the notion of consistent labeling culminated in a seminal paper by Hummel and Zucker [38] who, motivated by the theory of variational inequalities, developed a formal theory of consistency that later turned out to have intimate connections with non-cooperative game theory [52]. The theory generalizes classical (boolean) constraint satisfaction problems to scenarios involving "soft" compatibility measures and probabilistic (as opposed to "hard") label assignments. Within this framework, under a certain symmetry assumption, consistent labelings also turn out to be equivalent to local solutions of a linearly constrained quadratic optimization problem.

In our formulation, the jigsaw puzzle problem is viewed as the problem of finding a consistent labeling satisfying certain compatibility relations, with an additional requirement for one-to-one correspondences between the puzzle's tiles and their positions. We solve the problem using classical *relaxation labeling* algorithm which enjoys nice theoretical properties [60] and offers the advantage of avoiding *ad hoc* projections and problematic step size choices. To enforce the one-to-one constraints we endow the algorithm with two "matrix balancing" mechanisms, one based on the well-known Sinkhorn-Knopp procedure and the other inspired by von Neumann's method of alternating

projections. Related to our work is the jigsaw puzzle problem formulation suggested by Andaló *et al.* [2], where a similar quadratic objective function is optimized with a different gradient-projection technique.

We conducted some preliminary experiments aimed at testing the plausibility of the proposed approach. We first show that in the presence of an ideal, or "oracle" compatibility measure, the "plain" relaxation labeling algorithm (that is, *without* enforcing one-to-one correspondence between pieces and locations) is always able to return perfect reconstruction results. We then show how performance deteriorates as we move away from this ideal setting, thereby demonstrating the necessity of the balancing operation. We conclude with experiments using real-world compatibility on publicly available datasets, which show the feasibility of the proposed combined technique.

1.4.2 Puzzle with eroded border

A more complex task concerns finding a solution when pieces are missing or eroded. Many real-world problems, such as recovering of ancient documents and broken artifacts [23], can be seen as jigsaw puzzles with missing information (boundaries or entire pieces). This task has been only partially explored in the last year due to its complexity [10, 58], as it requires developing the compatibility method that takes into account the gaps between the pieces and the imprecision of the matching bounds.

In the second part of this research we extend our method [40] for the case where the borders of the patches are ruined. To simulate the erosion in the puzzle, we create gaps between pieces removing pixels lying on the borders. The gaps interrupt the color and the line continuation between patches, making compatibility functions unusable or highly inaccurate.

To alleviate this problem, we adopt an image extension technique; we extend the patches borders to cover the eroded parts in the picture with synthetically generated pixels. Image inpainting and extension are broadly studied in computer vision, and various techniques were proposed [77, 92, 17, 54, 5]. We consider that for our task, the image extension model is more suitable than the inpainting model, as we want to extend the images outside the original border rather than filling missing parts inside of each patch. The GAN-based model for image extension proposed in [77] shows impressive results, hence we adopt their model for our procedure: first, we recover the eroded borders of each patch by extending it in all directions and we compute the pairwise compatibility on repaired patches; then we apply the relaxation labeling solver [40] to reconstruct the image.

To summarize, the contributions of this paper are three-fold:

1. We formalize the puzzle-solving problem as a problem of finding a consistent labeling problem, with an additional requirement for one-to-one correspondences between the puzzle's tiles and their positions.

2. We develop the puzzle-solving algorithm based on relaxation labeling process, which we endow with matrix balancing mechanisms to enforce one-to-one correspondence constraints. We show the feasibility of our model on a variety of different datasets.

3. We extended the proposed model to handle a more complex task, such as jigsaw puzzles with eroded borders, proposing a model that exploits generative adversarial networks and relaxation labeling processes together. To the best of our knowledge, no such method was proposed for solving puzzles; hence, here we suggest a novel approach to address the complexity of puzzle games.

1.5 Thesis structure

The thesis is organized as follows: in Chapter 2 we offer the reader a brief survey of machine learning applications in Cultural Heritage; in Chapter 3 we discuss the state-of-the-art of puzzle-solving methods, puzzle-solving problem formulation and compatibility computation methods; in Chapter 4 we recap the basic concepts of consistent labeling problem and relaxation labeling process; in Chapter 5 after presenting our approach for puzzle solving and relaxation labeling puzzle solver, we discuss some preliminary studies and results; Chapter 6 presents our approach for solving puzzles with eroded border, where we make use of the GAN-model for image extension to visually reconstruct the ruined parts of the patches.

1.6 Publications

- Khoroshiltseva, M., Vardi, B., Torcinovich, A., Traviglia, A., Ben-Shahar, O., Pelillo, M.: Jigsaw puzzle solving as a consistent labeling problem. In: Computer Analysis of Images and Patterns. pp. 392–402. Springer International Publishing (2021)

- Khoroshiltseva M., Traviglia A., Pelillo M., Vascon S.: Relaxation Labeling Meets GANs: Solving Jigsaw Puzzles with Eroded Boundaries. In: 21st International Conference on Image Analysis and Processing, (accepted). Springer International Publishing (2022)

- Fiorucci M.; Khoroshiltseva M.; Pontil M.; Traviglia A.; Del Bue A.; James S. Machine Learning for Cultural Heritage: A Survey. In: Pattern Recognition Letters, vol. 133, pp. 102-108

Chapter 2

Machine learning methods and Cultural Heritage

Before diving into the discussion of puzzle-solving methods and solutions, here we propose an overview of machine learning methods nowadays applied in the field of Cultural Heritage. This survey was conducted at the beginning of this research journey and was useful to determine the direction of the research project.

The use of Machine Learning (ML) techniques within Cultural Heritage (CH) are still limited since most of CH literature shows a tendency to rely on statistical toolboxes, which are commonly applied as a 'black-box' on small datasets that are not generally publicly available. Despite this, we look to reflect on the reciprocal effects of ML on CH and of CH on ML. As usual in ML surveys, we break the field of ML into three distinctions: Supervised, Semi-supervised, and Unsupervised.

As state-of-the-art techniques take time to become popular in other fields such as CH, it is intuitive that more classical classification and regression techniques, such as Linear and Logistic regression, have a distinct and useful application within CH. While these can be applied in conservation efforts, such as historical building integrity prediction [62], there are numerous other examples of supervised approaches. Interestingly the application of Support Vector Machines (SVM) [32] refined the hyper-parameter estimation to support multiple-instance learning for recognizing iconographic elements in artworks.

With increasing efforts for digitization of CH assets, the progression to Deep Learning models is natural, where modern data-trained models are fine-tuned to CH data. This process is generally placed under the umbrella

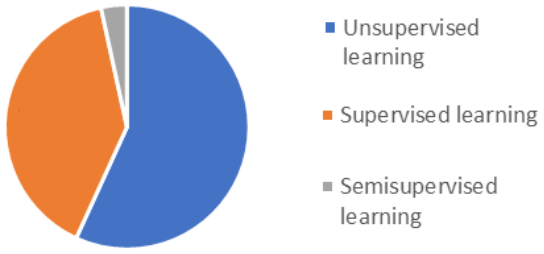

Figure 2.1: Distribution of Cultural Heritage papers among the three Machine Learning categories.

of transfer learning; such approaches are simple to apply when small amounts of labeled data are available, a common issue in CH and where it is frequently applied for digital artwork classification [66]. Building on transfer learning is a useful ability to learn a mapping from real-world imagery, in which we have many annotated examples, to artwork datasets where there are few available. Such approaches have received increasing attention as they can naturally be formulated within a deep learning context. Using high accuracy supervised Convolutional Neural Networks is desirable if the embedded knowledge can be transferred, especially where the transfer function can be learned in an unsupervised manner. While these techniques apply to many problems, they are predominantly seen on digital artwork as it is mainly a style difference to be overcome.

The unsupervised techniques are generally applied to clustering of data, with K-Means being regularly employed within CH [85, 72], and going beyond artwork with the clustering of chemical signatures for iron-making complexes [24]. Clustering can be seen to be highly important in CH facilitating the association of complex representations of assets. Dimensionality reduction is extensively used within CH, with Principal Component Analysis (PCA) being a common technique. Although many of the applications of CH are applied in the Computer Vision setting, due to the ease of data acquisition, there are also some examples from other fields including chemical analysis which exploits ML techniques for CH problems.

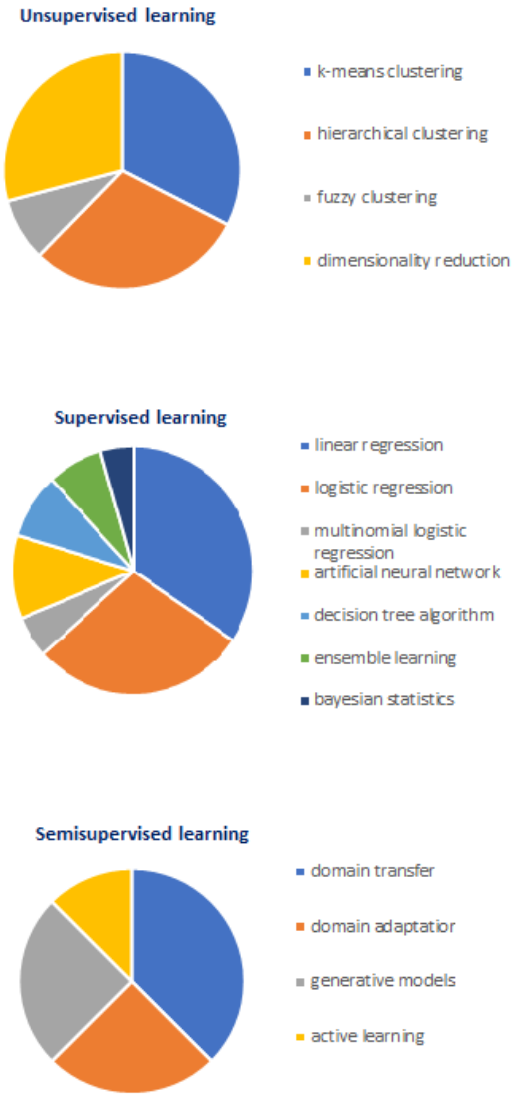

Figure 2.2: (1) Distributions of the Cultural Heritage papers among Unsupervised methods; (2) Distributions of the Cultural Heritage articles among Supervised methods; (3) Distributions of the Cultural Heritage articles among Semi-supervised methods

2.1 Datasets

The presence of different social and technical barriers to the cross-fertilisation between ML and CH become apparent after reviewing the most recent literature regarding the interplay between the two disciplines. The major part of these barriers have been generated by some issues that are strongly related to the quality and to the access of datasets collected by CH researchers. These datasets are often small and are not publicly available. However, recently, CH institutions have worked hard to make available large digital collections of artworks.

One of the largest museic-centric dataset is *OmniArt* [1] [75] which is composed of digitised artworks aggregated from a multiple collection around the world. The authors provided baseline scores on multiple tasks such as author, period, gender and style prediction. Another large collection of digitised artworks is *Wikiart paintings*[2] [39] which is composed of paintings from 1119 artists ranging from fifteen century to contemporary painters. Available metadata allow to classify a painting based on its style, gender and author.

A dataset of contemporary artworks is *BAM* [3] [87] which was built by collecting artworks from a portfolio website for professional and commercial artists (*Behance* [4]). Other collections of digital artworks are IconArt [32] contained painting images ranging from the 11th to the 20th century, PrintArt [11] composed of artwork prints collected from the *Artstor* digital image library [5] and Rijksmuseum the Rijksmuseum dataset [51] contained photographic reproductions of the artworks exhibited in photographic reproductions of the artworks exhibited in this museum.

2.2 Unsupervised Learning in Cultural Heritage

Unsupervised learning aims to find the structure and the regularity of an unlabelled dataset to extract useful representations. Among the unsupervised learning methods, clustering algorithms are the most widely used in CH. Clustering methods assign data points into groups, called clusters so that the pairwise similarities between points assigned to the same cluster tend to be higher than those in different clusters. It is worth noting that

[1] http://www.vistory-omniart.com/
[2] http://www.wikiart.org/
[3] https://bam-dataset.org/
[4] https://www.behance.net/
[5] https://www.artstor.org/

dimensionality reduction is heavily used within CH. However, it is mainly used only as a pre-processing step or as a visualization tool for representing high-dimensional data in 2-D plots.

Clustering algorithms can be divided into two main groups, namely, partition-based, where the points can be grouped in disjoint or overlapping clusters and hierarchical clustering where a nested series of partitions are produced given a criterion for merging or splitting clusters based on a similarity measure.

Clustering algorithms were reported for several applications in CH, among which constructing a codebook of visual words to chronologically classify ancient paintings [13]; recognizing objects in artistic modalities by unsupervised style adaptation Thomas; grouping paintings by artistic style using unsupervised feature learning [78]; determining maximum firing temperatures of ancient ceramics [34]; grouping 3D morphometric data of pounding stones to infer the intensity of humane use[6]; studying osseous projectiles using geometric morphometrics [26], and for chemical characterizing of Portuguese 18th-century glasswares [46].

A method for recognizing the modeling style of Dazu Bodhisattva head images was introduced by Wang et al.[85]. They proposed a two-step approach where first a pre-trained VGGNet [72] was used to extract prominent features of resized head images, and then k-means was applied to cluster the extracted features to verify if statues with similar style came from the same cave or region. The experimental results showed that statues in the same cave have a similar modeling style and are also similar for statues on the same subject even if they came from different regions.

An approach for comparing the chemical signature of iron artifacts to infer the origin of the metal supplied to the building yard of the Metz city was proposed by Disser et al.[24]. They introduced a multi-step approach, using PCA to determine the more characteristic element of a given chemical domain (since the chemical signatures of iron-making complexes were geometrically represented by multidimensional clusters) and the minimum-variance hierarchical clustering to group iron-making complexes that are coherent in terms of chemical composition. The experimental results showed that their method was able to detect slight modifications of the level of iron ores and slags, beating the state-of-the-art in comparing artifact chemical signatures.

2.3 Supervised Learning (Deep Neural Networks)

In this section we chose to review the supervised deep learning methods used in Cultural heritage. In the last years, Deep Neural Networks (DNNs) have successfully been used for several computer vision and natural language processing applications. That is due to the ability of DNNs to learn high-level features from data replacing the need for handcrafted features, which requires a great deal of human time and effort. Since for CH, there is often a lack of large labeled datasets, researchers tackle the feature learning task following a transfer learning approach, where the last layers of a pre-trained network are fine-tuned on the target CH dataset. However, only recently, DNNs have been attracting the interest of CH scholars, who have begun applying them to digital work analysis and archaeological remote sensing, as technologies to efficiently collect large datasets are now readily available.

Recent contributions to digital work analysis are the study of similarity metric learning methods for making aesthetic-related semantic-level judgments, such as predicting the painting's style, genre, and artist [67]; the detection of fake artworks by stroke analysis [28], and the artistic style transfer using adversarial networks to regularise the generation of stylized images [91].

A study of the applicability of Convolutional Neural Networks (CNNs) for attributing the authorship to different artworks, recognizing the material which has been used by the artist in their creations, and classifying artworks into different artistic categories was conducted by Sabatelli et al.[66]. They followed two transfer learning approaches: an off-the-shelf classification where only a final soft-max classifier was trained on the target training set, while the pre-trained CNN weights did not change; and a fine-tuning approach where the CNN was trained together with the final soft-max classifier on the target domain by optimizing the last layers of the pre-trained neural network.

A comparative experimental analysis was conducted using four CNNs pre-trained on ImageNet: VVGG19 [72], Inception-V3 [76], Xception [15] and ResNet50 [90]. The experimental evaluations were performed on two datasets of paintings: the Rijksmuseum Challenge 2014 dataset [51] and a much smaller dataset obtained by random sampling from the DAMS (Digital Asset Management System) repository, which aggregates several digital collections come from the city of Antwerpen. The experimental results showed that the fine-tuning approach outperformed the off-the-shelf one since fine-tuned CNNs provided novel selective attention mechanisms over the images. However, the off-the-shelf approach was effective in recognizing materials

and in classifying artworks, while it failed in attributing the authorship.

Recent applications of DNNs to archaeological remote sensing are the classification of sub-surface sites using R-CNNs on LiDAR data [81] and the detection of buried sites on Arc GIS data [68]. Both contributions followed a transfer learning approach by fine-tuning on LiDAR data a pre-trained CNN on ImageNet [21]. However, pre-training DNNs on RGB ImageNet images to identify objects in one channel depth LiDAR images may lead to performance degradation. Moreover, objects in ImageNet can appear at different scales but in not many different rotations, while for aerial data the scale variations are relatively small, but objects can have several different rotations.

To overcome these limitations, Gallwey et al.[31] proposed a method to detect industrial heritage sites by employing a pre-trained CNN, called DeepMoon [71], on a single channel Digital Elevation Model (DEM) images of the lunar surface [4]. The authors fine-tuned the DeepMoon network on the Dartmoor dataset for detecting historic mining pits. The experimental results showed that the proposed approach was able to differentiate between natural depressions and man-made ones with a false positive rate of less than 20%. Hence, this approach can be employed as a pre-prospecting tool for helping archaeologists to vastly reduce the area to be manually analyzed.

2.4 Semi-supervised Learning

Semi-Supervised Learning (SSL) aims to leverage both labeled and unlabeled data to improve learning performance. Most parts of SSL algorithms learn by jointly optimizing a supervised loss over labeled data and an unsupervised loss over both labeled and unlabeled data. Among these methods, domain adaptation is the most widely used in CH. Its goal is to transfer the knowledge learned from a source domain to a target domain, for which labels are usually not available, by finding a mapping between the data distribution of these two domains

In recent years, semi-supervised DNNs attracted increasing interest in the ML community. This arises from the idea of exploiting the powerful representation-learning ability of DNNs using only a small number of labeled examples, which are often expensive and difficult to collect. Following this idea, semi-supervised DNNs have proven to be very effective tools to tackle the domain adaptation problem [86]. This is of particular interest for the CH community, since domain adaptation has found applications in visual work analysis. Recent contributions in this direction are the automatic annotation

of visual contents in ancient manuscripts [3], and the prediction of painting style [87].

A semi-supervised visual-semantic model for cross-modal retrieval of images and captions, in which the pairing between images and captions was not known at training time, was proposed by Carraggi et al. [12]. In their approach, two autoencoders were trained, respectively for visual and textual data of the source domain, producing an intermediate representation used to create a common embedding space, where both modalities can be projected and compared.

A semi-supervised visual-semantic alignment was then applied to learn the relationship between the visual and textual features in the target unsupervised dataset. To evaluate the method, they introduced a new CH dataset, called EsteArtworks, which contains 553 artworks and 1278 textual annotations related to the artwork's visual contents. The experimental results showed that the distribution alignment gives a significant contribution to the final performance if the visual and textual distributions of the target domain are not like those of the source domain.

A semi-supervised method to retrieve artworks presenting near-duplicate visual elements was introduced by Shen et al. [69]. A two-step approach for learning deep features by leveraging spatial consistency across matches was proposed. First, hard-positive matching examples were found using spatial consistency as a supervisory signal, and then the positive matched features were updated using a single gradient step of the triplet loss. Experimental results showed the effectiveness of the proposed method in retrieving near-duplicated elements across different artworks.

2.5 Conclusion remarks to Chapter 2

The widespread adoption of Machine Learning algorithms within Cultural Heritage is clear throughout the literature. However, in most cases, it has been applied within a 'black box' setting where there are only a few examples of changes to the underlying formulation of the algorithm. Only a few approaches take advantage of more advanced ML.

The ability to access data in sufficient quantities limits the applications of ML methods in CH. Therefore, it is logical that articles relating to adaption to ML algorithms are predominantly on digital artwork analysis, as acquisition and data are readily available. The more active areas in ML for CH relate to archaeological artifacts as well as their chemical analysis. The trend to increase the joint optimization of features and classification algorithm clearly has had a profound effect on the accuracy and usefulness of algorithms. It is therefore foreseeable that more CH applications can take advantage of the developments in deep learning.

Chapter 3

Puzzle-solving methods

In this chapter we discuss existing puzzle solving methods, compatibility functions, and formalization of the puzzle-solving problem.

3.1 Solving puzzle with computational methods

In this section, we discuss the puzzle-solving methods that do not rely on artificial intelligence methods. Usually, these methods operate 2D patches exploiting the content of each patch, such as colors and patterns. These characteristics make it possible to solve a puzzle thanks to the identification of visual continuities. Based on the content, the puzzle-solving algorithm matches the patches and thus may produce the precise reassembly (solution of the puzzle). In recent years, the puzzle problem was tackled with different computational approaches proposing a variety of solutions, involving greedy algorithms, graphical models, and functional optimization [61, 55, 70, 74, 14, 30, 9, 2, 40].

Pomeranz et al. [61] introduced the first fully automatic puzzle solver proposing a greedy placer and a novel prediction-based dissimilarity. Their approach relies on finding pairs of pieces with a very high probability of being together ("best buddies").

Cho et al. [14] presented a graphical model based on the patch transform and proposed an algorithm that minimizes a probability function via loopy belief propagation.

Paikin et al. [55] extended the work in [61] by solving puzzles with unknown orientations and with missing pieces, introducing new affinity measures based on dissimilarity of the patches and the "best buddies" metric as in [61]. Similarly to [61] they used a greedy placement algorithm that

iteratively selects the best candidate to be placed, and takes particular care for selecting the first piece.

Sholomon et al. [70] proposed a solver based on a genetic algorithm that can solve large puzzles. The algorithm merges two wrongly-solved parents into a child that minimizes dissimilarity, while the fitness function is based on the compatibility of the pairs.

Gallagher et al. [30] represented a puzzle as a graph. Their algorithm considers edges connecting all pieces in all possible geometric combinations and then trims edges by finding a Minimum Spanning Tree. They also introduced a new compatibility function based on the Mahalanobis distance, which considers the local gradients near a patch's borders.

Brandao et al. [9] extended the work introduced in [30] by modeling the jigsaw problem as an edge selection problem in a graph, where the nodes represented the various tile orientations.

Andalo et al. [2] presented a global formulation for jigsaw problems, optimizing the affinity between adjacent pieces by numerically solving a constrained quadratic program.

Son et al. [74] considerably improved solving puzzles with an unknown orientation by using loop constraints. In their work, they reduced the dependency on dissimilarity instead of exploiting the consensus. The algorithm solves puzzles in a bottom-up fashion: starting from pairs of patches, it iteratively assembles the pair of pairs and so on, till merge all the structures.

All these works take the approach of local solutions making the problem more tractable, but fail to include the global information in a satisfactory manner. Although the graphical model of Cho al. [14] and Andalo et al. [2] present a more global solutions, the pairwise nature of this problem still persists. In addition all these approaches become increasingly computationally expensive as all permutations have to be explored. To overcome this limitation, more recent works were introduced using neural networks, which solve puzzles by unsupervised learning images structures.

3.2 Solving puzzle with neural networks

Most recent works introduce neural networks, which solve puzzles by unsupervised learning images structures [53, 18, 50, 93]. These approaches radically differ from the previously discussed ones, as they do not use the similarity between adjacent tiles; instead, they use the overall structure of shapes in the environment. The idea is to exploit different libeling that are available within visual data and to use them as intrinsic signals to learn

general-purpose features. Such a learning task is known as *self-supervised* learning and is very useful to learn rich features, which often require large amounts of annotated datasets.

Noroozi and Favaro [53] followed the principles of self-supervised learning and introduced Context-Free Networks for solving a Jigsaw puzzle as a pretext task. The network is trained to identify each tile as an object part and to assemble them in objects.

Cruz et al. [18] proposed a generic data-driven approach for learning visual permutations. They developed a framework based on the Siamese type of CNN named DeepPermNet.

Mena et al. [50]proposed a model that can learn permutation implicitly. Similarly to [18] they presented a model based on approximation of a non-differentiable permutation in terms of differentiable relaxations (Sinkhorn Network). The model uses a simple element-wise linear map for each of the N elements of the set to N positions and normalize by Sinkhorn operator [73].

Zhang et al. [93] improved [50] approach by introducing a new learning cost function based on pairwise comparisons, that allows to improve the learning from permutation considering local relations. An alternated optimization process seeks the correct permutation and a suitable cost matrix assessing pairwise relationships between objects.

Ru Li et al. [45] introduced JigsawGAN, a self-supervised GAN-based approach, that combines global semantic information and edge information of each piece, to solve 3x3 puzzle. The output of the model is then a permutation matrix of all the pieces.

3.3 Solving puzzle with missing border

Only few papers [58, 10, 45] addressed solving jigsaw puzzles when borders are missing.

Paumard et al. [58] tackled the 3x3 puzzle problem with a probabilistic model; to emulate the erosion, they randomly cropped a fragment inside each piece; then, given a central fragment, they used a neural network to predict the relative positions of the remaining fragments and computed the shortest path in the graph to reassemble the puzzle.

Bridger et al. [10] proposed a method to solve the puzzle with ruined regions; first, they recovered the missing parts using a GAN-based model and then reconstructed the image using greedy solver form [55]. Although the method works nicely, it is computationally intensive since it considers all the possible combinations of patches pairs and their relations.

Derech et al. [23] proposed to match overlapping fragments rather than searching valid continuities. To do so, they extrapolated the fragments and superposed the extrapolation, looking for a match. Then, they solved the puzzle one piece after another: they used the current reassembly to place the next fragment. They also considered a slight erosion of the fragments borders and tackled it by using inpainting techniques.

3.4 Puzzle-solving task

As we discussed above, several strategies for puzzle-solving have been proposed in the literature in recent years. Although these works approach the problem in a different way, any puzzle-solver procedure is usually composed of two components: the compatibility measure, which tackles the relationship among pieces and is the core component of any puzzle solver; the puzzle problem formulation (how we formulate and tackle the problem), that determines the solving technique.

Following this scheme, we present some popular compatibility measures used in the literature and two alternative problem formulations for the puzzle-solving task (quadratic optimization and graph matching problem).

3.4.1 Compatibility measure

Compatibility measure is a key component in any jigsaw puzzle solvers as it predicts the likelihood of two patches to be neighbors. In most cases the piece affinity is measured by computing the dissimilarity between the abutting boundary pixels of two adjacent pieces. Even this particular approach can be implemented in many different ways. The most commonly used compatibility measures include the sum of squared distances (SSD) proposed by Cho et al. [14], the so-called prediction-based compatibility (employing (Lp) q variants) proposed by Pomerantz et al. [61], the Mahalanobis gradient compatibility (MGC) proposed by Gallagher et al. [30] and its improved version proposed by Son [74]. Here we recap some of them.

Sum of square differences (SSD) dissimilarity

Given two pieces x_i, x_j and a relation R, their dissimilarity is computed by summing pixel difference of their (relevant to R) edges. For example, for puzzles with $K \times K$ and relation r (x_j is right to x_i), the measure is as

follows:

$$SSD_{p,q}(x_i, x_j, r) = (\sum_{k=1}^{K}\sum_{d=1}^{3}(|x_i(k, K, d) - x_j(k, 1, d)|)^p)^{\frac{q}{p}} \qquad (3.1)$$

Prediction based dissimilarity

Given two pieces x_i, x_j and a relation R, their dissimilarity is computed by comparing predictions of edges and the actual edges. For example, for puzzles with $K \times K$ and relation r (x_j is right to x_i), the measure is as follows:

$$Pred_{p,q}(x_i, x_j, r) = [\sum_{k=1}^{K}\sum_{d=1}^{3}(|2x_i(k, K, d) - x_i(k, K-1, d) - x_j(k, 1, d)|)^p$$
$$(|2x_j(k, 1, d) - x_J(k, 2, d) - x_i(k, K, d)|)^p]^{\frac{q}{p}},$$

Improved Mahalanobis Gradient Compatibility (MGC)

Improved Mahalanobis Gradient Compatibility (MGC) [74] consists of two components: similarly to MGC, first it considers pixel intensity changes crossing the border between two pieces; then it considers directional derivative information along the adjoining boundaries of the pieces. The compatibility measure between the right side of the piece x_i and the left side of the piece x_j is formulated as:

$$\Gamma_{LR}(x_i, x_j, r) = D_{LR}(x_i, x_j) + D_{RL}(x_j, x_i) + D'_{LR}(x_i, x_j) + D'_{RL}(x_j, x_i)$$

The first two terms penalize larger or smaller changes of the pixel value across the two pieces than expected

$$D_{RL}(x_j, x_i) = \sum_{s=1}^{S}(\Lambda^{ij}_{LR}(s) - E^{ij}_{LR}(s))V^{-1}_{iL}(\Lambda^{ij}_{LR}(s) - E^{ij}_{LR}(s))^T,$$

where:

$$\Lambda^{i,j}_{LR}(s) = x_j(s, 1) - x_i(s, S)$$

is a pixel intensity change across the boundary of the two pieces;

$$E^{i,j}_{LR}(s) = \frac{1}{2}x_i(s, S) - x_i(s, S-1) + x_j(s, 2) - x_j(s, 1)$$

is an expected change across the boundary of the two pieces, V_{iL} - is sample the covariance calculated from the following samples

$$(x_i(s,S) - x_i(s,S-1))|s = 1,2,...S$$

D'_{LR} is calculated similarly to D_{LR}, but replacing $x_i(u,v)$ with directional derivatives

$$\delta(u,v) = x_i(u,v) - x_i(u-1,v)$$

Comparison of compatibility functions

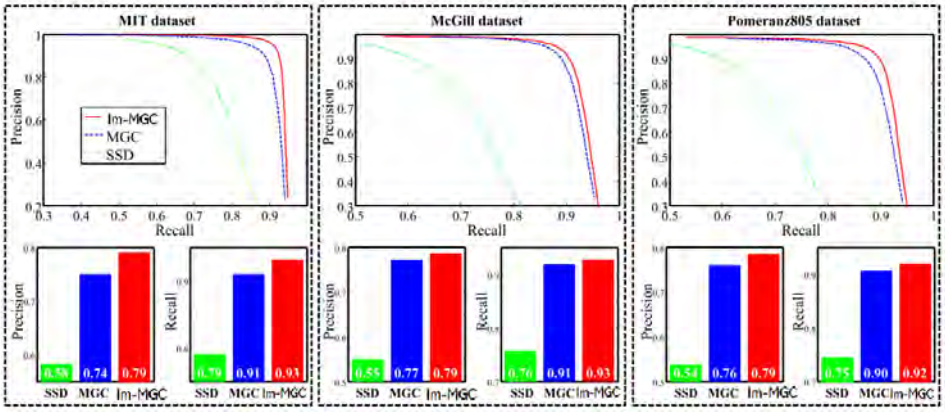

Figure 3.1: Pairwise compatibility measure performance comparison. The figures represent precision and recall curves for Sum of Squared Distance (SSD) [61], MGC [30] and Improved MGC measure [74], and precision and recall values when each side of the piece possesses a single neighbor which returns lowest dissimilarity for various datasets [74]

.

Son et al. in [74] evaluated the performance of pairwise compatibility measures by using *precision* (ratio of true positives to positives) and *recall* (proportion of true positives retrieved). In their analysis, they compared three popular dissimilarity measures: Sum of Squared Distance (SSD) [61], Mahalanobis Gradient Compatibility (MGC) [30], and Improved MGC [74] for three widely used benchmark datasets. Figure 3.1 presents the precision and recall curves. According to this analysis, the proposed Improved MGC [74] method reduces the precision and recall errors by 20 percent.

3.4.2 Problem formulation

Quadratic programming

The puzzle-solving task can be reduced to a quadratic optimization problem[2]. In particular, it consists of finding a permutation of the pieces that maximizes a global compatibility function.

In this formulation, a puzzle is formulated as a directed graph $G = (V, E)$, with $V = \Lambda$ and $E = E_H \cup E_V$, E_H and E_V denoting horizontally and vertically neighboring locations, respectively. For each pair of pieces (b_1, b_j), $i \neq j$, two non-negative compatibilities $C_{H_{i,j}}$ and $C_{V_{i,j}}$ are defined; these represent the compatibility of the two pieces being associated with locations connected by any horizontal edge $e \in E_H$ or vertical edge $e \in E_V$, regardless of the absolute positions of the pieces. All the compatibilities can be summarized in two compatibility matrices C_H and C_V.

Given a permutation represented as a permutation matrix $\mathbf{p} \in \mathcal{P} \subseteq [0,1]^{n^2 \times n^2}$, we evaluate its global compatibility as:

$$\varepsilon(\mathbf{p}) = \sum_{(i,j) \in E_H} (\mathbf{p}^\top \mathbf{C}_H \mathbf{p})_{ij} + \sum_{(i,j) \in E_V} (\mathbf{p}^\top \mathbf{C}_V \mathbf{p})_{ij} \tag{3.2}$$

Consider to "stretch" the permutation matrix $\mathbf{p}$ to obtain a vector $\mathbf{p}_{st} = (\mathbf{p}_{st_1}, \ldots, \mathbf{p}_{st_n})$, p_{st_i} being the i-th column of $\mathbf{p}$, then Equation 3.2 can be rewritten as:

$$\varepsilon(\mathbf{p}) = \mathbf{p}_{st}^\top \mathbf{A} \mathbf{p}_{st} = \sum_{i=1}^{n} \sum_{k=1}^{n} \sum_{l=1}^{n} \sum_{j=1}^{n} p_{ki} A_{(ki)(lj)} p_{lj} \tag{3.3}$$

where $A_{(ki)(lj)}$ is element (lj) of block (ki) of matrix $\mathbf{A}$, and $A_{(ki)}$ represent the compatibilities associated to the edge $(i, j) \in E$, thus it can be $\mathbf{C}_H$, $\mathbf{C}_V$ or $\mathbf{0}$ if $(i, j) \notin E$.

Graph matching

We can also formulate the puzzle task as a graph matching problem; this formulation is tightly related to the previous one.

Let $\pi : [n] \to [n]$ be a permutation function, with $[n] = \{1, \ldots n\}$. Furthermore, let $S \in \mathbb{R}^{n \times n}$ be a similarity/compatibility matrix. Then the function to maximize is the following:

$$\max_{\boldsymbol{\pi}} \sum_{i=1}^{n-1} S_{\pi(i), \pi(i+1)} \tag{3.4}$$

which in turn becomes:

$$\max_{\mathbf{p}\in\mathcal{P}} tr(\mathbf{L}\mathbf{p}^\top\mathbf{S}\mathbf{p}) \tag{3.5}$$

With $\mathbf{L} = \mathbf{U}^\top$ being a lower shift matrix, namely $u_{ij} = 1$ if $j = i + 1$, 0 otherwise.

The program can be rewritten in the following form:

$$\min_{\mathbf{p}\in\boldsymbol{\mathcal{P}}} \|\mathbf{U} - \mathbf{p}\mathbf{S}\mathbf{p}\|_F^2 \tag{3.6}$$

where it is made clear that we want our permutation to be as similar as possible as the graph represented by $\mathbf{U}$; the problem can be seen as a the graph matching problem.

Program 3.5 and 3.6 can be relaxed by searching in the multi-simplex $\mathcal{D} \subseteq (0,1)^{n\times n}$, transforming them respectively in an indefinite and a convex optimization problem. In [47] it is pointed out that, in general, while the indefinite formulation seems more difficult to solve, actually the found solution is nearer to that of the unrelaxed problem than that found with the convex formulation.

3.5 Our methods

In our paper *Khoroshiltseva at al.* [40] we propose to tackle the puzzle-solving problem as a problem of finding a consistent labeling that satisfies certain compatibility relations. The problem is solved using the classical relaxation labeling algorithm coupled with the Sinkhorn-Knop matrix normalization procedure [73], while adopting the Mahalanobis gradient compatibility function [30] to calculate the affinity of the parts.

In the second part of this research we extend our method [40] for the case of puzzle with eroded borders. Similarly to Bridger et al. [10], we tackle the puzzle problem with ruined regions; however, their work differs from ours in two crucial points: *i*) [10] fills in the gaps in the image by applying inpainting algorithm to each pair of patches for all possible transformations; instead, we recover the damaged borders of each single patch using image extension algorithm. That is more convenient from a computational point of view. *ii)* [10] uses a solver based on naive greedy placer; instead, we cast the problem as a consistent labeling problem [38], and solve the puzzle using the relaxation labeling algorithm that enjoys excellent theoretical properties [60].

Chapter 4

Consistent Labeling Problem

Consistent labeling problem is a class of problems widely studied in the computer vision and pattern recognition communities since the 1970s [65, 36]. The notion of a consistent labeling was formalized and culminated in a seminal paper by Hummel and Zucker [38] who, motivated by the theory of variational inequalities, developed a formal theory of consistency. The theory generalizes classical (boolean) constraint satisfaction problems to scenarios involving "soft" compatibility measures and probabilistic label assignments. Within this framework, under a certain symmetry assumption, consistent labelings also turn out to be equivalent to local solutions of a linearly constrained quadratic optimization problem.

The relaxation labeling procedure was derived heuristically by Rosenfeld, Hummel and Zucker [38] to solve certain constraint satisfaction problems. Although successfully applied in a variety of practical tasks, the lack of a satisfactory definition of consistency and of a justified relation with their model, inspired and motivated researchers to solve these severe drawbacks. To this scope, two main approaches have been used: on one hand probabilistic methods based on Bayesian analysis [59, 41, 16], on the other deterministic ones [80, 29]. While the firsts provided undoubtedly much insight into the understanding of the model, at the same time they suffered from some strong limitations in understanding its dynamical properties.

Deterministic approaches have better served to this scope, and thanks to the standard theory of consistency in labelling problems defined in [38] many advances have been achieved over years [41, 88]. In this context, inspired by [43, 27], Pelillo [60] definitively explained that, under certain symmetry restriction of the compatibility matrix, Rosenfeld et al.'s relaxation scheme monotonically increases the consistency function defined in [38]; moreover he

showed that most of the main dynamic properties of the algorithm continue to hold even if such requirement is relaxed.

4.1 Nonlinear Relaxation Labeling

In this section we recap some basic concepts of relaxation labeling processes.

Problem

Suppose we are given a set of objects $B = \{b_1, \ldots, b_n\}$ and a set of labels $\Lambda = \{\lambda_1, \ldots, \lambda_m\}$, the task is to assign a label to each object in B. Numerous real-world problems, typically discrete in nature, can be abstracted in this way.

"Soft" labeling assignment

Label assignments for object b_i can be represented by a probability distribution $\mathbf{p}_i$ over all possible labels. Formally, $\mathbf{p}_i \in \Delta^m$, where

$$\Delta^m = \left\{ \mathbf{x} \in \mathbb{R}^m \mid x_\lambda \geq 0 \quad \wedge \quad \sum_{\lambda=1}^{m} x_\lambda = 1 \right\} \tag{4.1}$$

is the standard m-dimensional simplex and $p_{i\lambda}$ is the probability of object b_i to "choose" label λ. Every vertex of Δ^m represents the case in which every object is exactly assigned to one label; the set Δ^* of all *unambiguous* labelling, is defined as:

$$\Delta^* = \{\mathbf{x} \in \Delta^m \mid x_{i\lambda} = 0 \; or \; 1, \; i = 1, ..., n, \; \lambda \in \Lambda\} \tag{4.2}$$

Aggregating all $\mathbf{p}_i$ vectors, the matrix

$$\mathbf{P} = \begin{bmatrix} \mathbf{p'}_1 \\ \vdots \\ \mathbf{p'}_n \end{bmatrix} \tag{4.3}$$

is a "soft" labeling assignment for all objects, residing in the multidimensional standard simplex $\Delta^{n \times m} = \Delta^m \times \cdots \times \Delta^m$, and thus may be represented as a labeling matrix of n rows and m columns.

Compatibility coefficients

An initial labeling assignment can originate from *local measurements* that capture the relevant features of individual objects when considered in isolation. Such local measurements typically provide imperfect labeling assignments, a state of affairs that *relaxation labeling processes* seek to improve progressively.

A major source of information that is utilized in such processes is *contextual information*, a prior that reflects the structure of the problem through compatibility relations about the assignment of labels in different objects. Contextual information is quantitatively expressed with a matrix of *compatibility coefficients*

$$\mathbf{R} = [r_{ij\lambda\mu}], \tag{4.4}$$

where $r_{ij\lambda\mu}$ measures the strength of compatibility between the hypotheses "b_i has label λ" and "b_j has label μ".

Contextual support

The compatibility model $\mathbf{R}$ is considered "contextual" because it naturally leads to measures of *contextual support*, i.e., how much the context supports the assignment of a particular label λ to object b_i. The "context" in this case is considered the labels assigned to all *other* objects, and following the classic relaxation labeling theory [38] it is defined as

$$q_{i\lambda} = \sum_{j,\mu} r_{ij\lambda\mu} p_{j\mu}. \tag{4.5}$$

where a high value of $q_{i\lambda}$ indicates that high-confidence neighbouring is "compatible" with λ on b_i, while a low value indicates that the high-confidence neighbouring is "incompatible" with λ. It is worth noting that low-confidence labels have low influence on the support measure.

Consistency

The consistent labeling assignments should be the "desired" goal of relaxation labeling processes. To this end, Hummel and Zucker [38] gave a formal definition of consistancy: a weighted labeling assignment $\mathbf{p}$ is said to be *consistent* if for all $\mathbf{v} \in \Delta^{n\times m}$

$$\sum_{\lambda}^{m} p_{i\lambda} q_{i\lambda} \geq \sum_{\lambda}^{m} v_{i\lambda} q_{i\lambda} \quad \forall i = 1, \ldots, n \tag{4.6}$$

Furthermore, if in 4.6 there is strict inequality, then $\mathbf{p}$ is said to be *strictly consistent*. Derived from the above definition the following theorem provides a useful characterization of consistent labeling, and an operation criterion to test the consistency of a labeling (see [60] for a proof.):

Theorem 1 *A labeling $p \in \Delta^{mn}$ is consistent if and only if for all $i = 1...n$ the following conditions hold:*

1) $q_{i\lambda} = c_i$, *whenever* $p_{i\lambda} > 0$

2) $q_{i\lambda} \leq c_i$, *whenever* $p_{i\lambda} = 0$

for some nonnegative constants $c_1, ..., c_n$.

Average local consistency

By properly weighting and combining the support of all labels to all objects, one can also quantify the average (or total) support of the assignment by the following formula:

$$A(\mathbf{p}) = \sum_{i,j} \sum_{\lambda,\mu} r_{ij\lambda\mu} p_{i\lambda} p_{j\mu}, \tag{4.7}$$

Hummel and Zucher defined this measure as *Average local consistency* in [38], where also proved that:

Theorem 2 *Suppose that the compatibility matrix R is symmetric (i.e., $r_{ij\lambda\mu} = r_{ji\mu\lambda}$ for all i, j, λ, μ). Then any local maximum $p \in \Delta^m$ of A is consistent.*

Put differently, a labeling assignment $\mathbf{p}$ that maximizes the average local consistency $A(\mathbf{p})$ represents a consistent labeling.

Update rule

The discussed above definition of consistency suggests us a way to adjust the labeling $\mathbf{p}$: a process that relaxes a given inconsistent assignment towards a more consistent one (and thus maximizes the average local consistency A) will intuitively care to increase $p_{i\lambda}$ when $q_{i\lambda}$ is high and decrease it when $q_{i\lambda}$ is low. Indeed, one of the best known update rules is defined by the following iterative procedure [65, 60]:

$$p_{i\lambda}(t+1) = \frac{p_{i\lambda}(t) q_{i\lambda}(t)}{\sum_{\mu} p_{i\mu}(t) q_{i\mu}(t)} \quad \forall i, \lambda \tag{4.8}$$

where the nominator formalizes this intuition and the denominator projects the result to the multi-simplex and guarantees that the updated vector still belongs to a probability space.

The relaxation algorithm can be viewed as a continuous mapping T of the assignment space onto itself. It starts out with $p^{(0)}$ and iteratively produces a sequence of points $p^{(0)}, p^{(1)}, p^{(2)}, ... \in \Delta^m$, where $p^{(t+1)} = T(p^{(t)}), t \geq 0$ The process continues until an equilibrium point is reached. This means that $T(p^{(t)}) = p^{(t)}$, for some t, or, equivalently,

$$q_{i\lambda} = c_i \quad \text{whenever} \;\; p_{i\lambda} > 0, \;\; i = 1...n, \;\; \lambda \in \Lambda, \tag{4.9}$$

for some nonnegative constants $c_i, ..., c_n$

Importantly, this update rule does not require the problematic choice of a step size and theoretical analysis has proven that under non-negativity and symmetry conditions on $\mathbf{R}$, it is guaranteed to converge to a consistent labeling and thus locally maximizes the $A(\mathbf{p})$. These important results derived from the Baum-Eagon Theorem and demonstrated by Pelillo in [60] as reported in the following.

4.2 The dynamics of nonlinear relaxation labelling process

To demonstrate that the relaxation labeling process converges to a consistent optimal solution Pelillo [60] offered a unified treatment of the algorithm and proved a list of proprieties that are strongly linked to the theory of consistency developed by Hummel and Zucker. More specifically, based on the Baum and Eagon theorem, he showed that, given a symmetric compatibility matrix, the algorithm possesses a Liaponov function which is precisely the measure of consistency defined by Hummel and Zucker.

Theorem 3 ***Baum-Eagon*** *Let $P(\mathbf{x})$ be a homogeneous polynomial in the variables $x_i(\lambda)$ with nonnegative coefficients, and let $\mathbf{x}$ be a point of the domain $\mathbb{K}$. Define the mapping $\mathcal{M}$ as follows:*

$$(\mathcal{M}(\mathbf{x}))_{i\lambda} = x_{i\lambda} \frac{\partial P(\mathbf{x})}{\partial x_{i\lambda}} / \sum_{\mu} x_{i\mu} \frac{\partial P(\mathbf{x})}{\partial x_{i\mu}} \tag{4.10}$$

Then $P(\mathcal{M}(\mathbf{x})) > P(\mathbf{x})$, unless $\mathcal{M}(\mathbf{x}) = \mathbf{x}$.

For a real-valued function f, a continuous mapping ϕ for which $f(\phi(\mathbf{x})) \geq (\mathbf{x})$ is called a *growth transformation* for ϕ.

Moreover Baum and Sell [ref 26] showed that $\mathcal{M}$ increases *homotopically*, that is:

$$P(\eta\mathcal{M}(\mathbf{x})) + (1-\eta)\mathbf{x}) \geq P(\mathbf{x}), \quad 0 \leq \eta \leq 1 \tag{4.11}$$

The theorem and this important result imply that an increment of the objective function is always guaranteed unless $\mathbf{x}(t+1) = \mathbf{x}(t)$; moreover, any points lying on the segment joining $\mathbf{x}(t+1)$ to $\mathbf{x}(t)$ is sufficient for its increment, independently from the step size.

Since the average local consistency is a homogeneous quadratic polynomial in the variables $p_{i\lambda}$ with non-negative, symmetric coefficients $r_{ij\lambda\mu}$

$$\frac{\partial A(\mathbf{p})}{\partial p_{i\lambda}} = 2q_{i\lambda} \tag{4.12}$$

the Baum-Eagon theorem can be directly applied to the non linear realization operator T, leading to the following result [60]:

Theorem 4 *The nonlinear relaxation operator T is a growth transformation for the average local consistency A, provided that compatibility coefficients are non-negative and symmetric.*

Put differently, the theorem 4 states that the nonlinear relaxation scheme strictly increases the average local consistency on each iteration:

$$A(\mathbf{p}^{t+1}) > A(\mathbf{p}^t), \;\; t = 0, 1, ... \tag{4.13}$$

until a fixed point is reached. Pelillo [60] also showed that the average local consistency can be viewed as a strict *Liaponov* function for the nonlinear operator T, implying the following result:

Theorem 5 *Supposing that $\mathbf{R}$ is symmetric, the unambiguous labeling assignment $\mathbf{e}$ is a stable equilibrium point for the nonlinear operator T and a local attractor consequently.*

In other words, the dynamic system converges to $\mathbf{e}$ whenever we start sufficiently close to it.

Although for puzzle-solving applications the compatibility matrix is always symmetric, for the sake of completeness, it is worth underlying that [60] showed that all the fundamental dynamic proprieties of relaxation labelling are retained even if such condition is relaxed, i.e. the unambiguous labelling is still an asymptotically stable equilibrium point for the nonlinear relaxation scheme T.

Chapter 5

Solving puzzle as Consistent Labeling problem

This chapter presents our approach to puzzle solving. In our formulation, the puzzle pieces are considered as a set of players (objects) and their possible positions as a set of strategies (labels); the puzzle problem is viewed as the problem of finding consistent labeling that satisfies certain compatibility relations, with an additional requirement for one-to-one correspondences between the puzzle's tiles and their positions. We solve the puzzle using classical relaxation labeling algorithm [60] that, starting from the uniform probability (barycentre point) distribution, progressively updates the assignment matrix till it converges to the consistent labeling, which in our case corresponds to a permutation matrix.

The chapter is organized as follows: in the first section 5.1, we review some background topics used in building up our puzzle-solving method. Section 5.2 presents our approach, more specifically the compatibility function, the Relaxation labeling puzzle solver; it then discusses the matrix balancing step and the method for approximation to the final permutation matrix. In section 5.3 we present some preliminary experiments to test the plausibility of the method, including experiments with oracle compatibility matrix and perturbation test. Finally, in section 5.4, we present the results of experiments with benchmark data sets, testing our method on puzzles of increasing size (256, 432, 810 pieces).

5.1 Preliminaries - terminology

In this section, we review some background topics that are used to build up our puzzle-solving approach: permutation matrix, doubly stochastic matrix, Sinkhorn-Knopp algorithm, Alternated projection method, Hungarian Algorithm.

5.1.1 Permutation matrix

In matrix theory, a permutation matrix is a binary square matrix that has exactly a single unit value in every row and column, and zeros elsewhere. These matrices are used to compactly represent permutations of elements in an ordered sequence. For instance, given an ordered sequence of n elements $S = \langle 1, ..., a_n \rangle$, any permutation $\pi : \{1, ..., n\}$ can be uniquely represented by a permutation matrix P_π.

These matrices have some useful properties: (1) permutation matrices are closed under multiplication, that is, the product of two permutation matrices is again a permutation matrix representing the combined permutation; (2) the inverse of a permutation matrix is equal to its transpose, $P^{-1} = P^T$ (orthogonality).

5.1.2 Doubly stochastic matrix

A non-negative matrix in which all its rows/columns sum to one, is said to be a row/column stochastic matrix. A matrix, that is simultaneously rows stochastic and column stochastic, is said to be doubly stochastic. Formally, an $n \times n$ matrix $A \in \mathbb{R}^{nxn}$ is said to be doubly stochastic if $A = (a_{ij})$ such that $a_{ij} \geq 0$ and $\sum_j a_{ij} = \sum_i a_{ij} = 1 \ \forall i, j = 1, ..., n$.

The permutation matrices are also doubly stochastic matrices. Furthermore, according to the Birkhoff-von Neumann theorem [7, 83] any doubly stochastic matrix is a convex combination of finitely many permutation matrices; in another words the set of $n \times n$ doubly stochastic matrices forms a convex hull of m $n \times n$ permutation matrices, with $m \geq (n-1)^2 + 1$ (also known as the Birkhoff polytope $\mathcal{B}_n$).

Thus, it is natural to see the doubly stochastic matrices as convex relaxations of permutation matrices. Approximating doubly stochastic matrices is a key problem in many applications as well as an crucial element of puzzle-solving algorithm proposed in this research. Next, we briefly present an efficient approach to carry out this task.

5.1.3 Sinkhorn-Knopp algorithm

The Sinkhorn-Knopp [73] theorem states that if A is a real nonnegative squared matrix, then there exist diagonal matrices D_1 and D_2 with strictly positive diagonal elements such that $D_1 A D_2$ is doubly stochastic. Sinkhorn-Knopp algorithm (also known as Sinkhorn Normalization) is a simple iterative procedure that takes as input a non-negative square matrix A and alternately normalizes all rows and all columns to sum to 1; the sequence of matrices obtained from this process converges to a doubly stochastic matrix:

$$\begin{aligned} S^0(\mathbf{A}) &= \exp(\mathbf{A}) \\ S^t(\mathbf{A}) &= T_c(T_r(S^{t-1}(\mathbf{A})) \\ S(\mathbf{A}) &= \lim_{t \to +\infty} S^t(\mathbf{A}) \end{aligned} \tag{5.1}$$

where $\mathbf{A}$ is a n-dimensional square matrix and T_c and T_r are respectively column- and row-wise normalization matrix operators.

Sinkhorn [73] presented the algorithm (1964) and proved that $S(\mathbf{A})$ must belong to the Birkhoff polytope, the set of doubly stochastic matrices. Knight [42] analysed the convergence of Sinkhorn-Knopp algorithm and stated that, for a matrix A with entries in $[1, V]$, $\mathcal{O}(V|\log_\epsilon|)$ iterations suffice to reach ϵ-near doubly stochasticity.

5.1.4 Alternating projection method

Alternating projection (also known as projections onto convex sets) is a method to find a point in the intersection of two closed convex sets. The Alternating projection algorithm solves the following problem: find $x \in \mathbb{R}^n$ such that $x \in C \cap D$, where C and D are closed convex sets.

The algorithm starts with an arbitrary value for x_k and then generates the sequence

$$x_{k+1} = \mathcal{P}_C(\mathcal{P}_D(x_k)) \tag{5.2}$$

where P_C and P_D denote projection on C and D respectively. If the intersection of C and D is non-empty, then the sequence generated by the algorithm will converge to some point in this intersection.

The Alternating projection was analysed by John von Neumann [82] for the special case when the sets are affine spaces; in this simplest case, assuming the intersection is non-empty, the iterates not only converge to a point in the intersection, but to the orthogonal projection of the point onto the intersection.

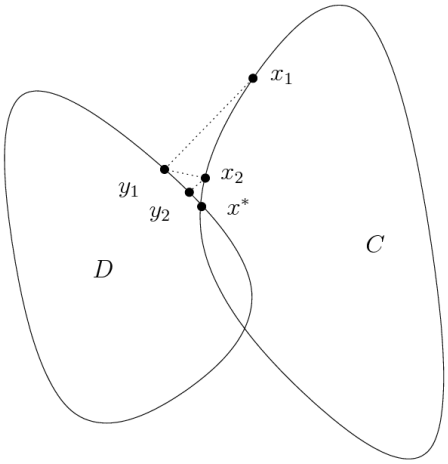

Figure 5.1: first few iterations of alternating projection algorithm; the sequences are converging to the point $x^* \in C \cap D$ [8].

5.1.5 Hungarian Algorithm

The Hungarian matching algorithm (sometimes called the Kuhn-Munkres algorithm), is used to find maximum-weight matchings in bipartite graphs, which is also known as the assignment problem.

Assignment problem. The problem instance has a number of agents (n) and a number of tasks (n). Any agent can be assigned to perform any task, incurring some cost. It is required to perform all tasks by assigning at most one agent to each task and at most one task to each agent. C is a non-negative matrix of size $n \times n$ (cost matrix), where the element in the i_{th} row and j_{th} column corresponds to the cost of performing the j_{th} type of task by the i_{th} agent. The goal is to match the tasks to the workers in such a way that the total labor input is minimized.

Algorithm. The Hungarian algorithm operates on the following two ideas:

- if the same number y is added to, or is subtracted from, all elements of any row or column of a cost matrix, the total cost will decrease by y, and an optimal assignment for the resulting cost matrix is also an optimal assignment for the original cost matrix.
- if there is a zero-cost solution, then it is optimal.

The algorithm looks for values to be subtracted from all elements of each row and each column such that all elements of the matrix remain non-negative, but a zero solution appears.

Assuming C is the $n \times n$ cost matrix the algorithm performs as follows:

1. Subtract a minimum value of every row

2. Subtract a minimum value of every column

3. Cover all zeroes with a minimum number of horizontal and vertical lines.

4. Check for optimality: if the min number of covering lines is n, the optimal assignment is found, else (the number of lines less than n) an optimal assignment is not found

5. (If an optimal assignment is not found). Find the smallest element not covered by any line; subtract this number from each uncovered line and add it to each covered column. Return to step 3.

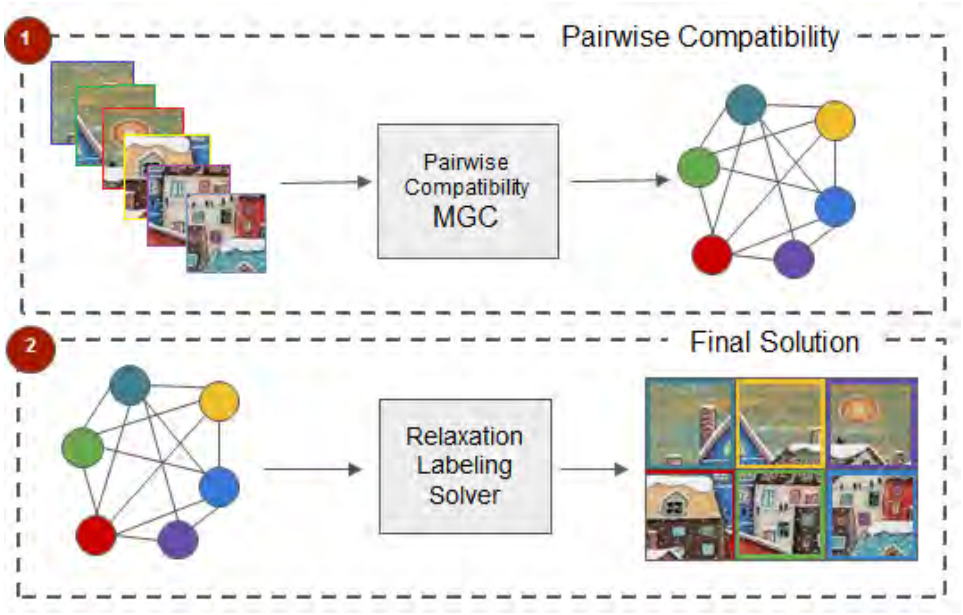

Figure 5.2: Pipeline of the algorithm. (1) We compute pairwise compatibility between all the patches using Mahalanobis Gradient Compatibility (MGC) [30]. (2) We use Relaxation Labeling to find a consistent labeling (positioning) of each piece.

5.2 Model

Here we present our method for puzzle-solving based on the Relaxation labeling algorithm. The pipeline of the procedure is presented in figure 5.2

We first calculate the compatibility map that is the core component of our puzzle-solving method, using Mahalanobis gradient compatibility (see subsection 5.2.1 for details). Then we solve the puzzle as a consistent labeling problem with additional one-to-one correspondence constraints, adopting, for this particular case, the classical relaxation labeling algorithm. Subsection 5.2.2 presents the details of the algorithm and matrix balancing methods. Last, we discuss the method for approximating the permutation matrix, that encodes the reordering of the pieces and represents a final solution of a puzzle.

5.2.1 Compatibility measure

The pairwise compatibility measure $C_{\mathcal{R}}(i,j)$ quantifies the affinity between pieces i and j when placed adjacent to each other in one of the spatial relationships $\mathcal{R} \in \{left, up, right, down\}$. This notion of compatibility can be extended naturally into a relaxation labeling compatibility after defining non-neighbor positions and self-comparison cases properly. We thus formalize $\mathbf{R}$ as follows:

$$r_{ij\lambda\mu} = \begin{cases} C_{\mathcal{R}}(i,j) & (i \neq j) \wedge (\lambda, \mu \text{ are adjacent locations in relation } \mathcal{R}) \\ 0 & \text{otherwise} \end{cases}. \tag{5.3}$$

As already mentioned, $C_{\mathcal{R}}(i,j)$ can be measured in numerous different ways. Similar to others [74, 61, 30, 55] we measure piece's affinity by computing the dissimilarity between the abutting boundary pixels of two adjacent pieces.

Dissimilarity score

The dissimilarity score of two pieces can be implemented in many different ways, here we adopt the improved Mahalanobis Gradient Compatibility (MGC) originally developed by Gallagher [30] and further improved by Son *et al.* [74], that considers both the color differences across pieces borders and the directional derivative differences along the borders. (see section 3.4.1 for details). The dissimilarity measure that we choose for our procedure is empirically demonstrated to be more reliable with respect to other measures proposed in the literature [74].

Form dissimilarity score to compatibility measure

With the dissimilarities obtained, we next convert them to normalized compatibility values by (a) dividing them by the K smallest dissimilarity and reflecting about 1, and (b) rectifying at zero as only positive scores are considered useful. Formally,

$$C_{\mathcal{R}}(i,j) = \max\left(1 - \frac{\Gamma_R(i,j)}{K_{min_{\mathcal{R}}}(i)}, 0\right) \tag{5.4}$$

where $\Gamma_R(i,j)$ is a dissimilarity score between two candidate pieces i and i placed adjacent to each other in spatial relationship $\mathcal{R} \in \{left, up, right, down\}$. $K_{min_{\mathcal{R}}}(i)$ is the K-min value of the dissimilarity between all other pieces in

relation $\mathcal{R}$ to piece i. The smaller the value of K, the more sparse $C_{\mathcal{R}}(i,j)$ becomes, leading to a sparser $\mathbf{R}$ matrix and, generally speaking, a more efficient relaxation labeling process.

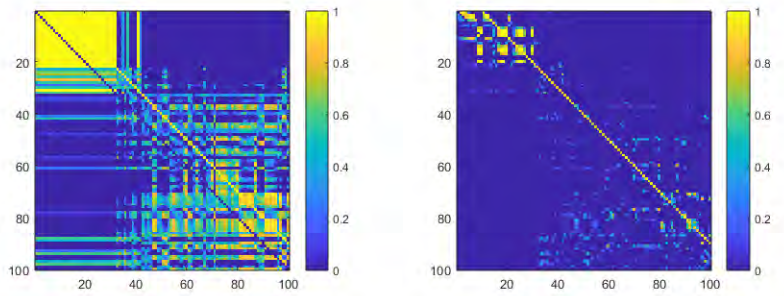

Figure 5.3: Sparsification effect of the compatibility matrix with K=50 and K=10 for puzzle of 100 pieces.

Best bodies sparsification

To further sparsify the compatibility matrix, we adopt the *best buddies* concept from Pomeranz *et al.* [61]. The procedure allows us to find the perfect pairwise matching by considering the ordinal position of the candidate in a ranking of other pieces. The Best Buddies relation is defined as the following [61]: two pieces are *best buddies* if both agree that the other piece is their most likely neighbor in a certain spatial relation. Two pieces i and j relation R_1 and opposite relation R_2 are best buddies if both hold:

$$\begin{cases} C_{R_1}(i,j) > C_{R_1}(i,k) & \forall k \neq j \\ C_{R_2}(j,i) > C_{R_2}(j,k) & \forall k \neq i \end{cases} \tag{5.5}$$

We set the compatibility of any two best buddies to perfect compatibility (i.e., 1), and zero the compatibility values of all other non-best-buddy matches concerning each of the two best buddies. We term this step the *sparsification* of the compatibility matrix and experiment both with and without it.

We observe that sparsification contributes to the computational efficiency of the solver, thus, high sparsification can seem encouraging. On another side, applying sparsification we risk cutting off true candidates: the BB sparsification is a rigid neighboring assignment and, in the case of false BB,

there is no way for the errors to be corrected by the algorithm, thus leading to the wrong solution.

5.2.2 Puzzle Solver

We cast jigsaw puzzle solving as a consistent labeling problem defined as above. Here, the set of objects B represents the puzzle pieces, the labels Λ are the relevant positions in the reconstruction plane, and the task is to assign a different position from Λ to each puzzle piece from B.

Note that in such an abstraction, the label-object representation can be easily exchanged to seek an assignment of a different piece to each possible position. In either case, not every assignment is admissible for puzzle solving, as one must seek a *permutation* matrix $\mathbf{p}$ that reorders the pieces to the correct positions.

Formally, the $\mathbf{P}$ is a soft assignment matrix, where each row $\mathbf{p}_i$ represents a probability distribution of the positions for a piece i and each column $\mathbf{p}_\lambda$ represents a probability distribution of the pieces for a position λ. Thus, $p_{i\lambda} \in \Delta^{n \times m}$ is the probability of piece i to choose position λ, and $\Delta^{n \times m}$ is the multi-simplex with

$$\Delta^m = \{\mathbf{p}_i \mid p_{i\lambda} \geq 0 \wedge \sum_\lambda p_{i\lambda} = 1\} \tag{5.6}$$

and

$$\Delta^n = \{\mathbf{p}_\lambda \mid p_{i\lambda} \geq 0 \wedge \sum_i p_{i\lambda} = 1\} \tag{5.7}$$

where $p_{i\lambda}$ is the probability of piece i to choose position λ. Thus the soft assignment matrix $\mathbf{P} = [p_{i\lambda}]$ is required to be a doubly stochastic matrix such that $\sum_\lambda p_{i\lambda} = \sum_i p_{i\lambda} = 1$.

Indeed, the relaxation labeling update rule from Eq. (4.8) guarantees that $\mathbf{P}$ is a stochastic matrix (i.e., rows sum to 1) but does not enforce the same constraint for its columns. Therefore, the optimization process can converge to a labeling that does not represent a permutation (producing an infeasible solution where the multiple pieces are assigned to the same position and vice versa).

To alleviate the problem and enforce one-to-one correspondence constraints, we endow the relaxation process with a matrix balancing step that encourages convergence into a permutation matrix. Two such steps are explored: Sinkhorn-Knopp (SK) normalization [73] and alternated projection algorithm [82].

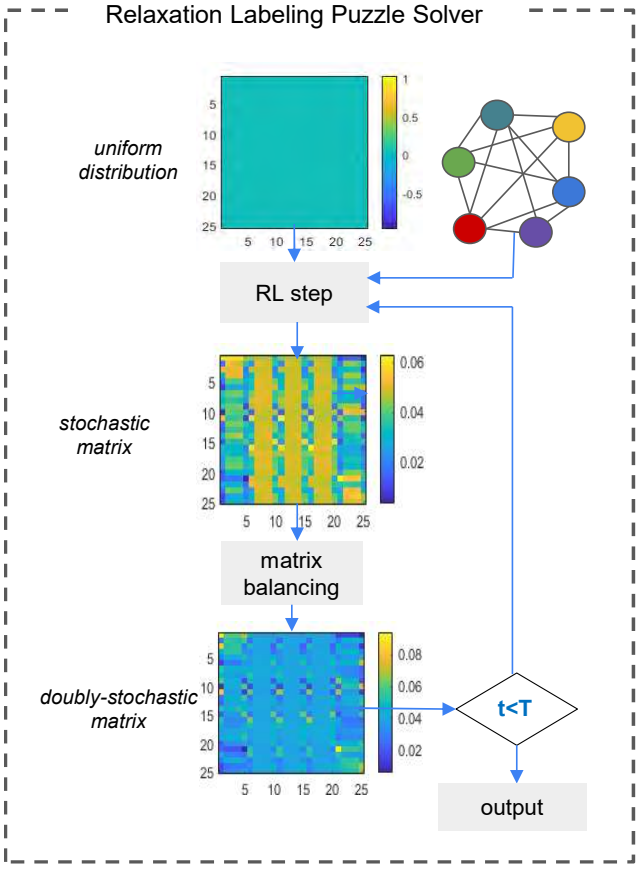

Figure 5.4: Relaxation Labeling Puzzle solver. The initial assignment is represented by an uniform distribution; on each step of solver the updated assignment (stochastic) matrix is normalised to its doubly stochastic version; in the last iteration the obtained doubly stochastic matrix is approximated to its closest permutation matrix that encodes the final solution.

Sinkhorn normalization

SK algorithm transforms a given non-negative square matrix to its related doubly stochastic version. This is done by alternately normalizing the rows and columns. Starting from an initial point $\mathbf{P}(0)$, the balancing is performed according to:

$$\mathbf{P}(t) = T_c(T_r(\mathbf{P}(t-1)) \tag{5.8}$$

where $\mathbf{P}$ is a n-dimensional square matrix and T_c and T_r are respectively column- and row-wise normalization matrix operators; the process converges to a doubly stochastic matrix. SK is incorporated in our algorithm as an additional balancing step applied after the update rule in each iteration. By this we encourage the relaxation labeling process to move in a space of doubly stochastic matrices (instead of stochastic) that pushes the convergence towards a permutation matrix.

As an alternative, we could apply the balancing step with different frequencies (i.e every 5 steps, 10 steps, etc). Intuitively, this would allow the algorithm to take its natural course, and the balancing step would only serve to correct the direction every now and then. Nevertheless, our experiments show that periodic balancing is not sufficient, and in most cases, the algorithm converges to a local minimum; figure 5.5 illustrates the behavior of the algorithm with different frequencies of balancing step in terms of average local consistency and related accuracy (defined as the "intermediate solution" produced by the permutation matrix approximated to a current assignment matrix).

Figure5.12 illustrates the "snapshot" of assignment matrix ($\mathbf{P}$) after 1, 2, 6, 11 and 17 iteration of RL solver comparing 4 versions of the algorithm: 1 - RL-solver without balancing, 2 - RL solver with balancing applied every 10 steps, 3 - with balancing applied every 5 steps; and 4 - RL solver with balancing applied every step.

Alternating projection

As we mention before the label-object (player-strategies) representations can be exchanged; thus, the assignment process can be seen from two points of view: assigning the position to the pieces and assigning the pieces to the position; these two processes occur contemporary and are expected to converge to a shared optimal point. This property suggests using *Alternating projection method*: sequential re-projection from piece-to-position assignment space to position-to-pieces assignment and vice-versa at every step of the relaxation labeling algorithm; doing this the algorithm converges to the

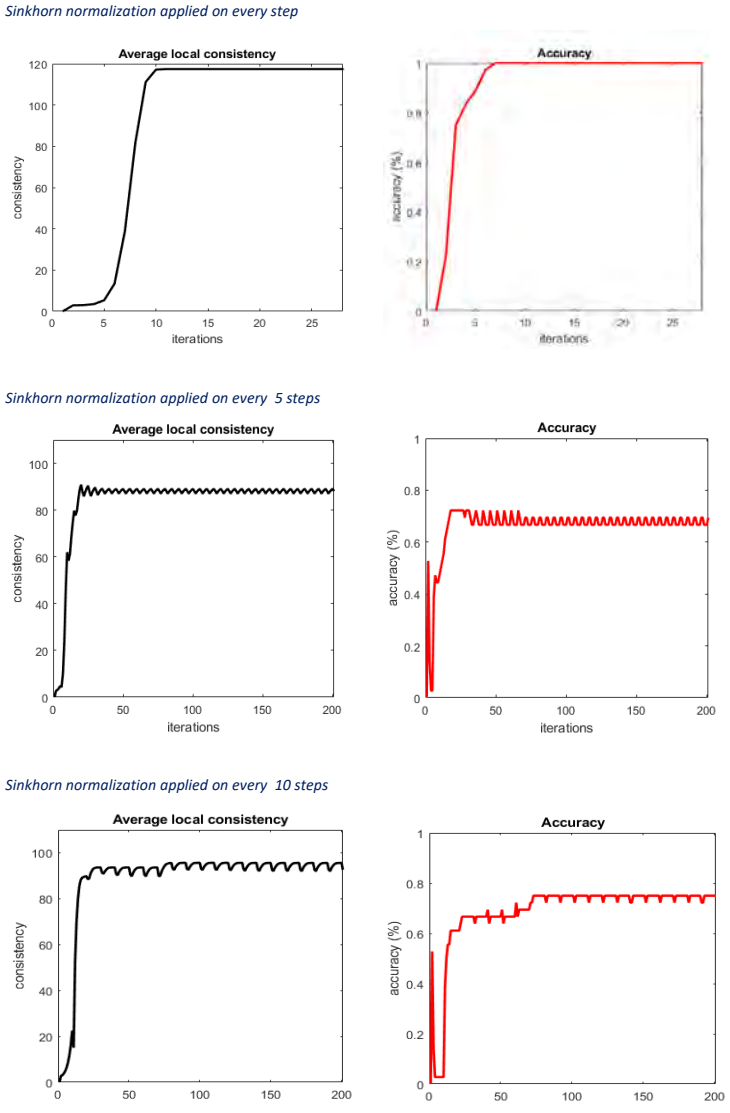

Figure 5.5: Comparison of the behaviour of the algorithm with different frequencies of matrix balancing, in terms of Average local consistency and direct accuracy; columns from left to right: (1) Sinkhorn balancing applied every step of Relaxation labeling puzzle solver, (2) Sinkhorn balancing applied every 5 steps, (3) Sinkhorn balancing applied every 10 steps. The experiment are done for a puzzle of 25 pieces.

point in the intersection of two spaces (that corresponds to the solution with respected one-to-one correspondence constraint).

Since the native relaxation labeling update rule normalizes the rows of the assignment matrix but not the columns, and since objects and labels in the puzzle solving abstraction are interchangeable, it is tempting to switch the role of pieces (objects, rows) and positions (labels, columns), thereby switching between row and column normalization every step of the process.

Computationally, this is done by alternately switching between two update rules, Eq. (4.8) and the following update rule:

$$p_{i\lambda}(t+1) = \frac{p_{i\lambda}(t)q_{i\lambda}(t)}{\sum_j p_{j\lambda}(t)q_{j\lambda}(t)} \quad \forall i, \lambda \tag{5.9}$$

and keep doing so until convergence. Similarly to the case with Sinkhorn normalization, one can be tempted to vary the frequency of the alternation (i.e. every 5, 10 steps instead of every step), thus allowing the algorithm to run several steps between the switching from one normalization mode to another. Figure 5.6 illustrates the results of the experiments varying the frequency of alternated projection (every step, every 5 steps, every 10 steps); the plots illustrate the behaviour of the algorithm in terms of average local consistency and related accuracy, for a puzzle of 25 pieces. Similarly to the same experiment with Sinkhorn normalization we observe better convergence of the algorithm when normalization is applied on every step.

Figure 5.13 illustrates the convergence of an assignment matrix to a permutation one, comparing the relaxation labeling algorithm without balancing with the one equipped with alternated projection (applied every 10, 5 steps and every step of the solver).

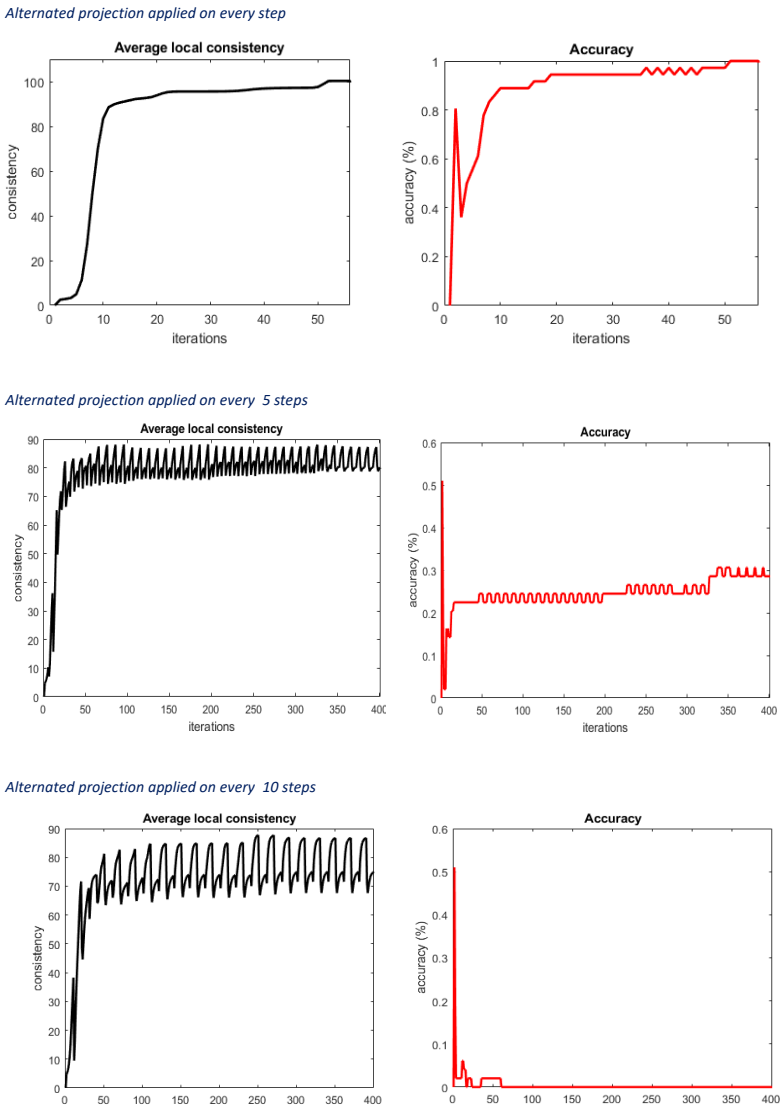

Figure 5.6: Comparison of the behaviour of the algorithm with different frequencies of matrix balancing, in terms of Average local consistency and direct accuracy; columns from left to right: (1) Alternated projection applied every step of Relaxation labeling puzzle solver, (2) Alternated projection applied every 5 steps, (3) Alternated projection applied every 10 steps. The experiment are done for a puzzle of 25 pieces.

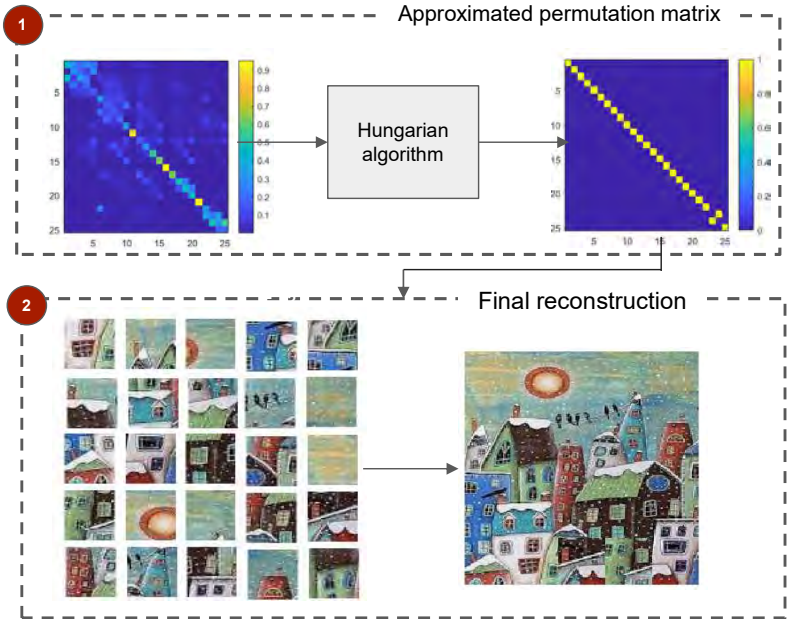

Figure 5.7: (1) Approximation of final solution to the closest permutation matrix by means of Hungarian algorithm; (2) final reconstruction of the puzzle from obtained permutation matrix.

5.2.3 Final reconstruction

In most cases we cannot expect that the algorithm converges to the permutation matrix. More often the output of the relaxation labeling algorithm is a doubly stochastic matrix that only partially represents the desired solution. In order to obtain the final solution that encodes the position of each piece on the assembling plane we have to approximate the predicted doubly stochastic matrix P to its closest permutation matrix P_{final}.

We solved the problem of the approximation of the permutation matrix as an assignment problem, using the Hungarian algorithm. In terms of Hungarian algorithm, the input *cost matrix* is represented by the final *soft assignment* matrix P of RL solver, while its output (*optimal assignment*) corresponds to the desired permutation matrix, which represents the final solution of the puzzle P_{final} (that assigns the puzzle pieces to their positions).

Figure 5.7 illustrates the procedure: (1) output of the relaxation label-

ing process, that represents "incomplete" solution, and closest permutation matrix obtained with of Hungarian algorithm, (2) results of assembling from obtained permutation matrix.

5.3 Preliminary study

5.3.1 Oracle compatibility

In order to explore the effectiveness of our solver, we first tested the proposed algorithm using a synthetically generated and ideal compatibility, dubbed *oracle compatibility*; it is defined as follows:

$$C_{\mathcal{R}}^{(oracle)}(i,j) = \begin{cases} 1 & \text{if } i,j \text{ are the correct neighbors in relation } \mathcal{R} \\ 0 & \text{otherwise} \end{cases} \tag{5.10}$$

We tested the relaxation labeling algorithm with this oracle compatibility on a puzzle of 540 pieces and examined its performance with and without balancing. All solvers have been executed with no-prior knowledge, i.e., using the barycenter of the standard multidimensional simplex as initial point. When balancing is used, we started the balancing algorithm after $t = 10$ iteration of relaxation labeling, to let the latter process first propagate the information without any constraint. The solvers always converged before reaching the maximum number of iterations set to $T = 200$.

Fig. 5.8 shows the behavior of the average local consistency function along the iterations. The experiment demonstrates that each proposed strategy reached a maximum and therefore a consistent labeling, corresponding to the correct permutation and the desired synthetic puzzle. As can be observed, endowing the procedure with either balancing components, significantly expedited the convergence by forcing the optimization process towards doubly stochastic matrix.

5.3.2 Perturbation test

In real-world puzzles, however, it is of course difficult to guarantee an ideal oracle compatibility, so we verified the robustness of the proposed solvers by perturbing all coefficients with additive Gaussian noise $\epsilon \sim \mathcal{N}(0, \sigma^2)$. Different experiments has been done with increasing values of σ, ranging in $(0, 0.2]$ for the experiments with and without sparsification technique. We applied the sparsification technique described in the previous section, setting $K = 5$.

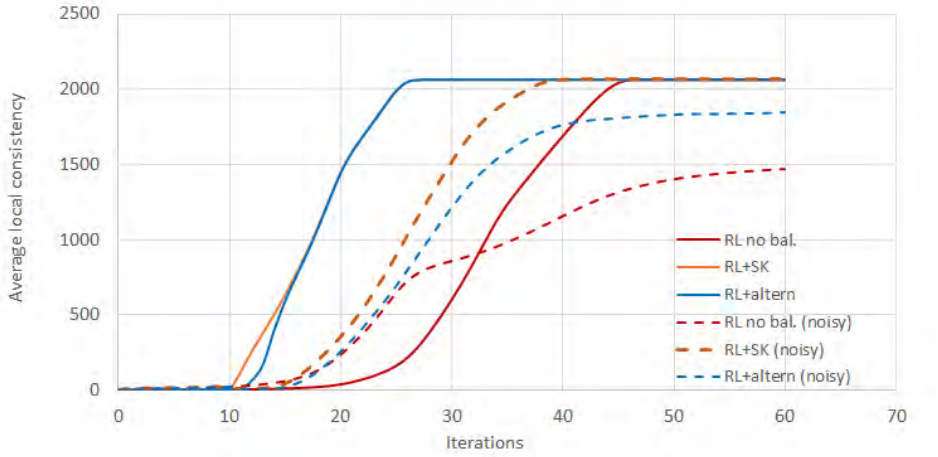

Figure 5.8: Average local consistency over the relaxation labeling iterations with oracle compatibilities (solid lines - ideal oracle compatibility; dashed lines - "noisy" oracle compatibility with $\sigma = 0.02$).

We assessed the performance of the three solvers using the *Direct Comparison* metric [14] which measures the ratio of pieces placed in correct position compared to the ground-truth. As Figure 5.9 shows, the performance drops immediately unless balancing is incorporated. Moreover, with sparsification, the performance persists even in the case of high levels of noise. This signifies the importance of this computational component. Such experiments show that the relaxation labeling scheme for consistent labeling is able to address the puzzle solving task quite effectively, and that the reliability of the compatibility function is a crucial aspect for the correctness of the solution.

5.4 Experiments with Natural Images

The quality of the compatibility computed for natural images depends on several factors that can be divided into two categories: the *image content*, that might make the assembling process particularly challenging, like the presence of repetitive patterns or large regions with homogeneous color; the *puzzle size* that, when large, may lead to less informative pieces increasing the ambiguity in the compatibility computation. For these reasons, the compatibility measure should be sensitive enough to capture small differences be-

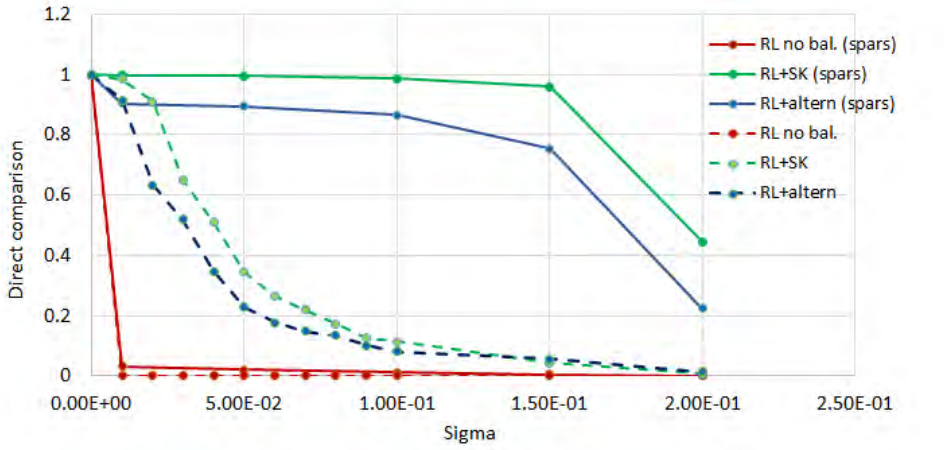

Figure 5.9: Drop in accuracy of our solvers due to the perturbation of oracle compatibility (solid lines - results with sparsification; dashed lines - results without sparsification).

tween pieces. The dissimilarity measure that we chose for our procedure (cf. Section 3.4.1) is empirically demonstrated to be more reliable with respect to other measures proposed in the literature [74].

As dissimilarity measures incorporate more pictorial information for larger piece sizes, we tested the algorithm on the up-scaled versions of widely used datasets. Scaling was done by applying bicubic interpolation with a scale factor of 2, resulting in doubling the number of pixels involved in each dissimilarity computation, without changing the puzzle size.

Datasets

We tested the algorithm on three datasets, each contains 20 images: 20 puzzles of 432 pieces from the *MIT dataset* presented by Cho *et al.* [14], 20 puzzles of 540 pieces from the *McGill dataset* and 20 puzzles of 805 pieces from *Pomeranz805* dataset, proposed by Pomeranz *et al.* [61].

All these datasets are widely used as benchmark to measure the performance of puzzle solver algorithms and contain several "problematic" images for which a good reconstruction is challenging. Some of the images contain horizontal and vertical lines that may be aligned with pieces edges and distort the compatibility; other images contain repetitive texture patterns (e.g., wall, grass) or homogeneous regions (e.g., sky, water) that may gen-

"Direct" accuracy	RL without balancing	RL with SK balancing	RL with alt. projections
432 pieces (MIT)	7.3%	89.9%	90.7 %
540 pieces (McGill)	11.7%	98.6%	98.1%
805 pieces (Pomeranz805)	13.7%	96.8%	96.7%
Mean	10.9%	95.0%	95.2%

Table 5.1: Reconstruction performance of puzzles from the MIT, McGill and Pomeranz805 datasets. *"Direct accuracy"* metrics.

erate numerous false positive compatibilities. In all datasets, piece size was scaled from 28×28 to 56×56 pixels, and the RGB color space was used in dissimilarity computations and all experiments refer to puzzles with known piece orientation.

Accuracy metrics

Aside from the Direct Comparison metric, we adopted two other measures to assess solvers performance: the *Neighbor Comparison* metric [14] that measures the ratio of correctly assigned neighbors in the solution, and the *Perfect Reconstruction* metric [30] that is a binary indicator of whether all pieces are in the correct position.

Results

As the results show in Tables 5.1, 5.2 and 5.3, the relaxation labeling algorithm (with balancing) can handle real-world compatibilities and solve real puzzles in most cases. At the same time, the datasets contain some images (such as images 1, 2, 4, 13, 15 in the MIT dataset) that generate multiple false compatibilities and for which, due to the presence of homogeneous tiles, the desired reconstruction is a challenging task. Such compatibilities then lead to ambiguous assignments with equal probability, hence affecting the quality of the solution obtained.

Analyzing the different versions of the puzzle solver it is evident that consistent labeling via relaxation labeling is a viable substrate for solving natural image jigsaw puzzles. However this requires either highly predictive compatibility, or more realistically, the incorporation of balancing in the update rule.

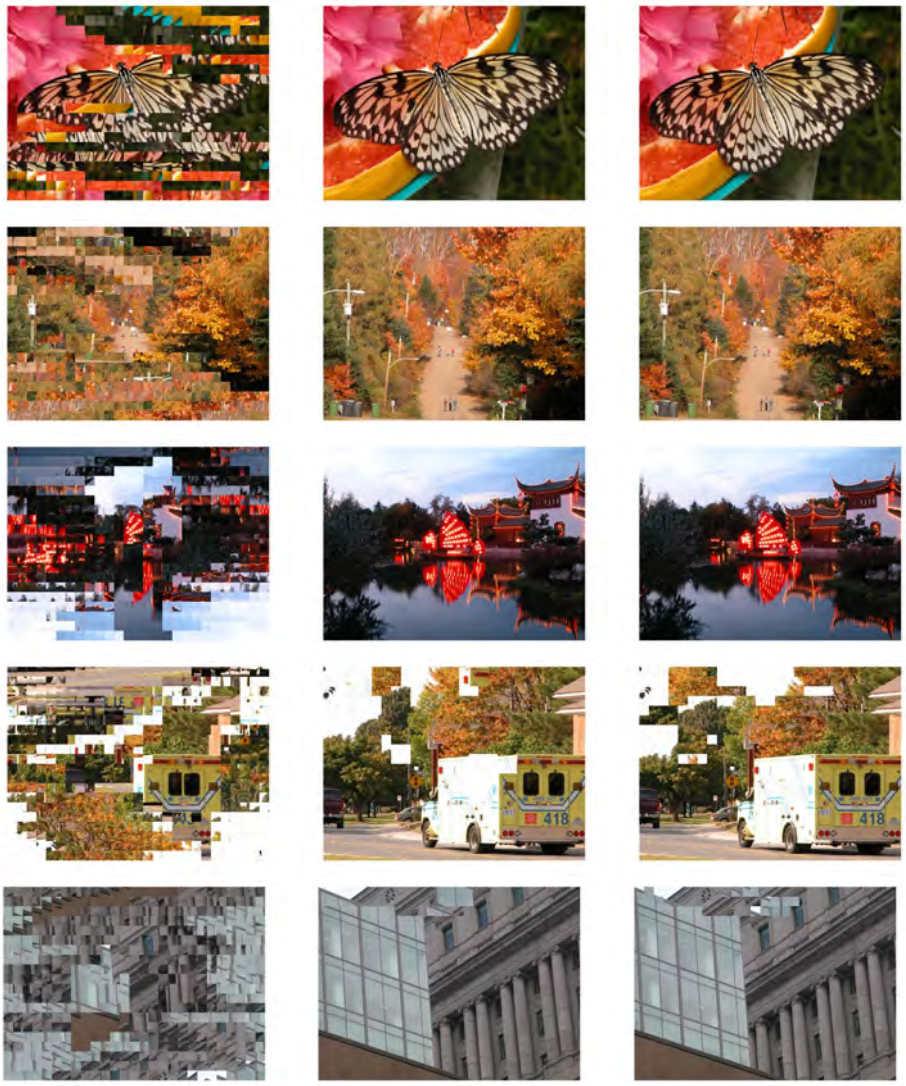

Figure 5.10: Qualitative reconstruction performance of five images: (left) results without balancing, (middle) results by SK balancing, (right) results by alternating projections.

"Neighbour" accuracy	RL without balancing	RL with SK balancing	RL with alt. projections
432 pieces (MIT)	7.3%	89.9%	90.7 %
540 pieces (McGill)	11.7%	98.6%	98.1%
805 pieces (Pomeranz805)	13.7%	96.8%	96.7%
Mean	15.5%	96.9%	96.5%

Table 5.2: Reconstruction performance of puzzles from the MIT, McGill and Pomeranz805 datasets. *"Neighbour accuracy"* metrics

"Perfect" accuracy	RL without balancing	RL with SK balancing	RL with alt. projections
432 pieces (MIT)	0%	75%	70%
540 pieces (McGill)	0%	75%	75%
805 pieces (Pomeranz805)	0%	80%	65%
Mean	0%	77%	70%

Table 5.3: Reconstruction performance of puzzles from the MIT, McGill and Pomeranz805 datasets. *"Perfect"* accuracy metrics - percentage of perfectly reconstructed puzzles in each of the datasets.

Analyzing the two versions of the algorithm (that differ in the balancing strategies) we can conclude that both versions are reliable and doing well for most of the datasets images

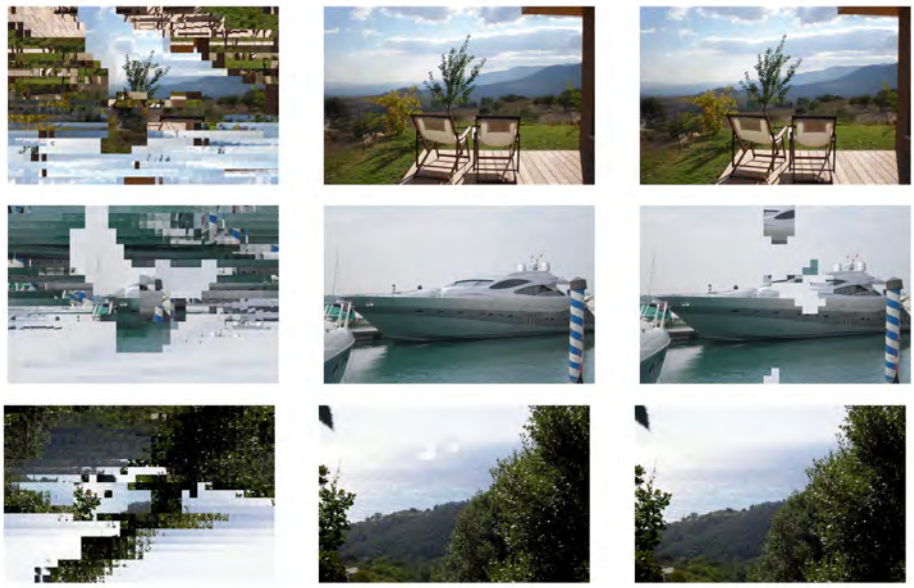

Figure 5.11: Qualitative reconstruction performance of three images from *McGill* dataset: (left) results without balancing, (middle) results by SK balancing, (right) results by alternating projections.

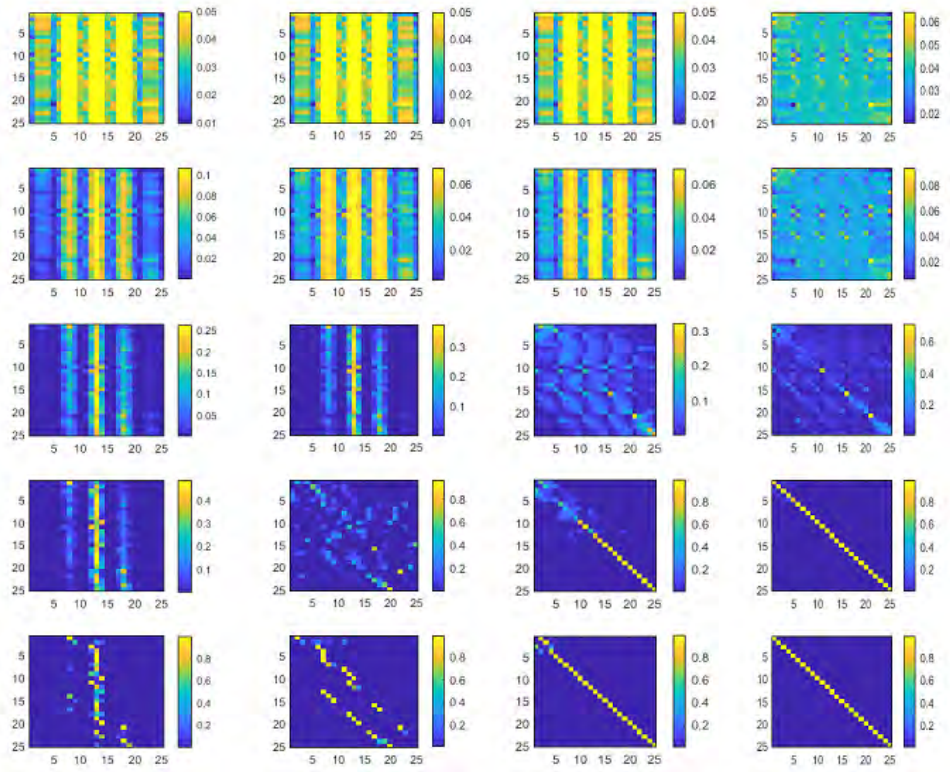

Figure 5.12: Comparison of the behaviour of the algorithm with different frequencies of matrix balancing, in terms of Assignment matrix; rows from top to down: the "snapshots" of assignment matrix after $1^{th}, 2^{nd}, 6^{th}, 11^{th}, 17^{th}$ iterations of the puzzle-solving algorithm; columns from left to right: (1) relaxation labeling algorithm without balancing, (2) Sinkhorn balancing applied every 10 steps, (3) Sinkhorn balancing applied every 5 steps, (4) Sinkhorn balancing applied every step of the puzzle-solver. The experiment are done for a puzzle of 25 pieces.

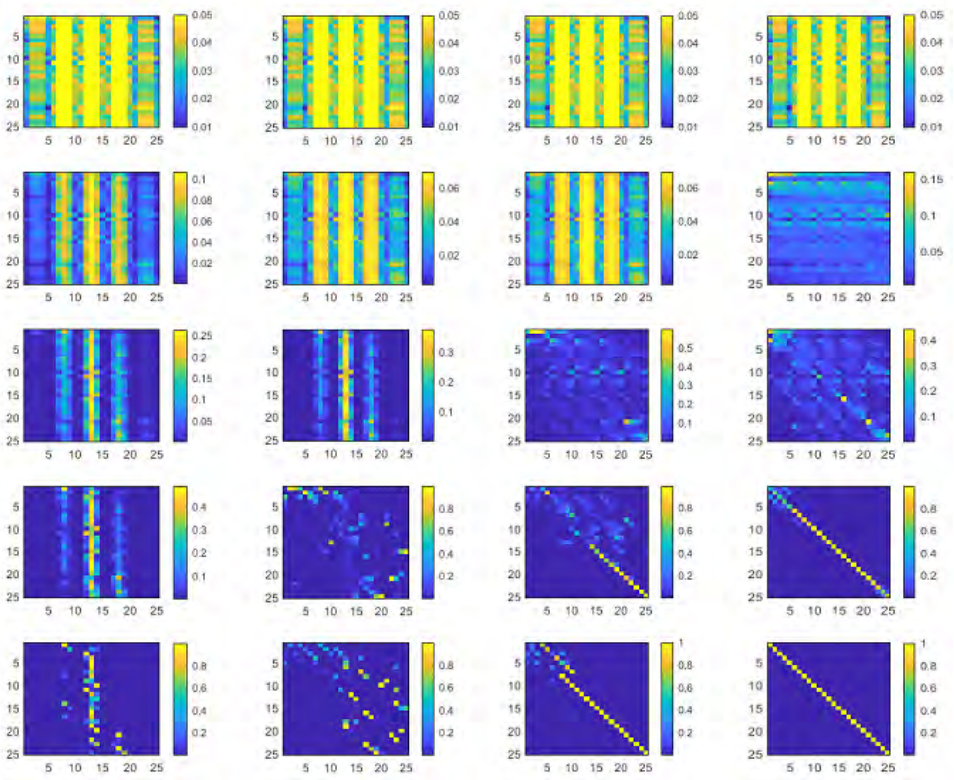

Figure 5.13: Comparison of the behaviour of the algorithm with different frequencies of matrix balancing, in terms Assignment matrix; in rows: the "snapshots" of assignment matrix after $1^{th}, 2^{nd}, 6^{th}, 11^{th}, 17^{th}$ iterations of the puzzle-solving algorithm; columns from left to right: (1) relaxation labeling algorithm without balancing; (2) Alternated projection applied every 10 steps; (3) Alternated projection applied every 5 step; (4) Alternated projection applied every step of the puzzle-solver. The experiment are done for a puzzle of 25 pieces.

Chapter 6

Solving Jigsaw Puzzles with Eroded Boundaries

6.1 Introduction

This chapter presents the extension of our relaxation labeling puzzle-solving approach for a more complex task, namely finding a solution when pieces are missing or eroded. Many real-world problems, such as recovering ancient documents and broken artifacts [23], can be seen as jigsaw puzzles with missing information (boundaries or entire pieces). This task has been only partially explored in the last year due to its complexity [10, 58].

To simulate the erosion in the puzzle we create gaps between pieces. The gaps produce interruption in the color and the line continuation between patches, and make the color gradient-based compatibility function unusable or, anyway, highly inaccurate. To alleviate this problem, we adopt an image extension technique; the idea is to extend the patches borders to cover the eroded parts in the picture with synthetically generated pixels. Image inpainting and extension are broadly studied in computer vision, and various techniques were proposed [77, 92, 17, 54, 5]. We consider that the image extension model is more suitable for our task, as we want to extend the images outside the original border rather than filling missing parts inside of each patch. The GAN-based model for image extension proposed in [77] shows impressive results, hence we adopt their model for our procedure. First, we recover the eroded borders of each patch by extending it in all directions, then we compute the pairwise compatibility on repaired patches; finally, we apply the solver [40] to reconstruct the image.

Related works

Only few papers addressed the task of solving jigsaw puzzles when border are missing. Paumard et al. [58] tackled the 3x3 puzzle problem with a probabilistic model; given a central fragment, they used a neural network to predict the relative positions of remaining fragments and computed the shortest path in the graph to reassemble the puzzle. Bridger et al [10] proposed efficient method to solve the puzzle with ruined regions; first they recovered the missing parts using GAN-based model and then reconstructed the image using greedy solver form [55]. Although the method works nicely, it uses a lot of information to generate the missing border, since it considers all the possible combinations of pairs and their location. Ru Li et al [45] introduced JigsawGAN, a self-supervised GAN-based approach, that combines global semantic information and edge information of each piece; the output of the model is then a permutation matrix of all the pieces. They applied such approach to solve $3x3$ puzzle problems.

In this research, similarly to Bridger et al. [10], we tackle the problem of the puzzle with ruined regions; however their method differs from ours in two crucial points: 1) Bridger [10] filled in the gaps in the image by applying inpainting algorithm to each pair of patches for all possible transformations; we, instead, propose recovering the damaged borders by generating the missing pixels all around the patch; hence we apply the image extension algorithm to each single patch, that is lighter from a computational point of view. 2) Differently from [10] who used a naive greedy placer, we cast the problem as a consistent labeling problem, formalized by Hummel and Zucker in [38], and solve the puzzle using relaxation labelling algorithm, that enjoys nice theoretical properties [60].

The chapter is organized as follow: in section 6.2 we preset a short overview of generative adversarial net and in particular Boundless model for image extension; section 6.3 describes our model and sections 6.3.1, 6.3.2 discus the image extension model and the compatibility computation respectively; in section 6.3.3 we recipe our puzzle solver. Finally, in section 6.5, we discuss the experiments and present our results.

6.2 Generative Adversarial Net (GAN)

Generative Adversarial Networks (GANs) [33, 94], are an approach to generative modeling using deep learning methods, such as convolutional neural networks. Generative modeling is an unsupervised learning task in machine learning trained for automatically discovering and learning the regularities in input data. GANs are a smart way to train a generative model by framing the problem as a supervised learning problem. It consists of two sub-models: the *generator* model that we train to generate new examples, and the *discriminator* model that classifies examples as either real (because drawn from the domain) or fake (because generated). The two models are trained together in a zero-sum adversarial game, until the discriminator model is fooled (in 1 times), meaning the generator model is generating plausible examples.

GANs are a rapidly changing field, they attract much attention due to their ability to generate realistic examples across a range of problem domains, such as image-to-image translation tasks (translating photos of summer to winter or day to night, etc) and generating photorealistic photos of objects, scenes, and people in such realistic way, that even humans cannot tell are fake. Generative Adversarial Networks have achieved impressive results in image generation [22, 63], image editing [96], and representation learning [22, 49]. Recent methods adopt the same idea for conditional image generation applications, such as text2image [64], image inpainting [57], and future prediction[48], as well as to other domains like videos[84] and 3D models [89]. The key to GANs' success is the idea of an adversarial loss that forces the generated images to be, in principle, indistinguishable from real images.

The GAN model architecture involves two models: a generator model that generates new examples and a discriminator model that classifies whether generated examples are real or fake. Generative adversarial networks are based on a game-theoretic scenario where the generator network competes against an adversary (discriminator). The generator network is generating samples, while its adversary (the discriminator network) tempting to distinguish between examples drawn from the training data and examples drawn from the generator. Figure 6.1 schematically illustrates the typical architecture of GAN.

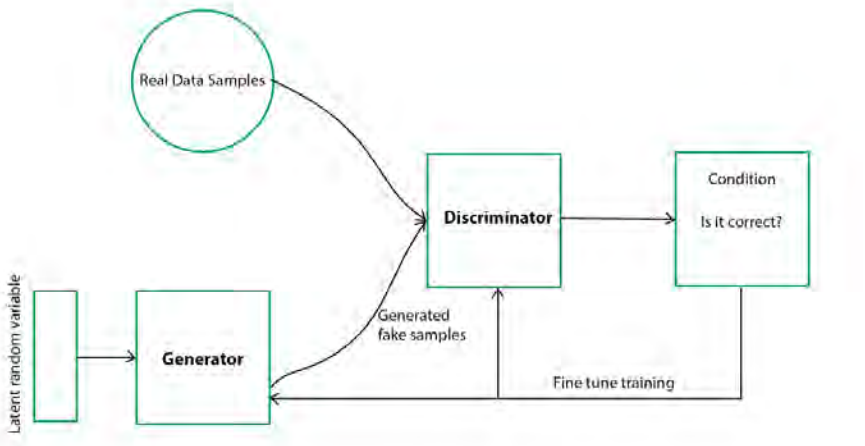

Figure 6.1: Schematic architecture of GAN

The GANs are formulated as a minimax game, where the Discriminator is trying to minimize its reward $V(D, G)$ and the Generator is trying to minimize the Discriminator's reward or in other words, maximize its loss. It can be mathematically described by the formula below:

$$\min_G \max_D V(D, G) \tag{6.1}$$

$$V(D, G) = \mathbb{E}_{x \sim p_{data}(x)}[\log D(x)] + \mathbb{E}_{z \sim p_z(z)}[\log(1 - D(G(z))] \tag{6.2}$$

where G is Generator, D is discriminator, $p_{data}(x)$ is the distribution of real data, $P(z)$ is the distribution of the generator, x is sample from $p_{data}(x)$, z is sample from $P(z)$, $D(x)$ Discriminator network, $G(z)$ Generator network.

Boundless GAN model for image extension

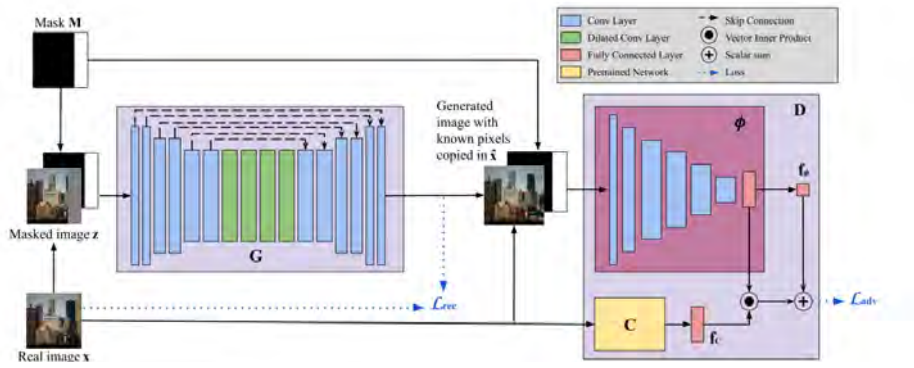

Figure 6.2: Boundless Model Architecture: this architecture is used for all our models.From Teterwak et al. [77]

The Boundless model [77] is developed for image extension task, that is, to fill in the image content outside the original boundaries. Unlike other inpainting algorithms [77, 92, 17, 54, 5], the extension region is expected to match the original area on structural, textual, and semantic levels. The model has the Wasserstein GAN framework composed of a generator network and a simultaneously trained discriminator.

Generator has the fully convolutional encoder-decoder architecture from [92] with additional skip connections between the non-dilated layers and instance normalization after each generator level. All the layers use gated convolutions and an ELU activation function; the final layer clips its outputs to the range $[1, 1]$.

The generator takes as input the image to be extended with the unknown region masked in the image; the output is a fully generated image. Before switching to the discriminator, the known region in the generated image is replaced with the original pixels from the input image.

Discriminator is a deep network, which transforms a generated sample into a single scalar that determines whether the generator's output is a plausible extension of its input. The overrating of the known part of the generated image with the original pixels creates a "seam" between the original and the generated parts of the image; if there is any abrupt change along the seam, the image is classified as false. The discriminator is designed "to be conditioned on the specific generator input"; the discriminator output is defined

as:

$$D(x^*, M, x) = f_\phi(\phi(x^*, M)) + \langle \phi(x^*, M), f(C(x)) \rangle \tag{6.3}$$

The model is trained via combination of reconstruction loss and adversarial loss. The reconstruction loss optimizes for image agreement; it is implemented as an l_1 loss imposed on the full output of $\mathbf{G}$ and defined as following:

$$\mathcal{L}_{rec} = \|\mathbf{x} - \mathbf{G}(\mathbf{z}, \mathbf{M})\|_1 \tag{6.4}$$

The adversarial loss refines the coarse prediction; here the Wasserstein GAN hinge loss is used [79] and defined as:

$$\mathcal{L}_{adv,G} = \mathbb{E}_{x \sim P_X(x)}[-\mathbf{D}(\hat{\mathbf{x}}, \mathbf{M}, \mathbf{x})] \tag{6.5}$$

The total loss on the generator is defined:

$$\mathcal{L}_{total} = \mathcal{L}_{rec} + \lambda \mathcal{L}_{adv,G} \tag{6.6}$$

Boundless is the first model where the GANs are effectively used to learn image extensions. The model is endowed with a stabilization scheme, based on using semantic information from a pre-trained deep network, to modulate the behavior of the discriminator of GAN. This semantic conditioning encourages the generated content to semantically match the target image.

The authors [77] demonstrate that the model achieves strong results on image extension with coherent semantics, plausible textures and colors. Moreover, the model also produces promising results for large extrapolations (up to 3 times the width of the original image).

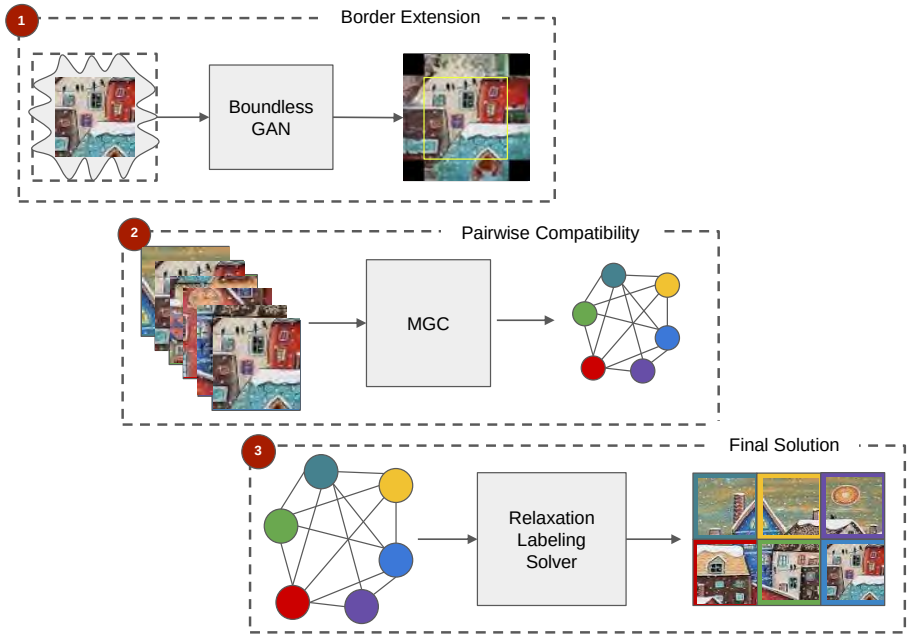

Figure 6.3: Pipeline of the algorithm. (1) Given a patch, we extend its borders using Boundless GAN [77]. (2) We exploited the generated borders and compute pairwise compatibility between all the patches using Mahalanobis Gradient Compatibility (MGC) [30]. (3) Relaxation Labeling is then used to find a consistent labeling (positioning) of each piece.

6.3 JiGAN Model for puzzle-solving

In this section, we introduce JiGAN, our GAN-based approach to solve jigsaw puzzles. Suppose we are given N images, that represent the patches of the puzzle; the borders of the patches are eroded implying gaps between parts in the puzzle. The goal is to reassemble the original image or, saying differently, assign a position in a 2-dimensional grid assemble plane to each patch of the puzzle. As in previous works, we assume that the patches are of the same size, the orientation is known, and the gaps created by eroded borders are of the same regular size.

Our model is illustrated in Figure 6.3 and is based on the three following key ideas: 1) extending the eroded patches border using a GAN model; 2) computing dissimilarity score for each pair of patches and transforming

dissimilarity scores into the matrix of compatibility coefficients; 3) given the compatibility map, running the relaxation labeling puzzle solver and reconstructing the image.

6.3.1 Border Extension

The various methods for compatibility computation, discussed in previous works [14, 61, 74, 55], are normally based on the color gradient and the continuation of the edge, and perform well for puzzles without erosion. However, the gaps created by erosion make any of these functions inaccurate and unreliable. In order to compute pairwise similarities between patches, hence reconstructing the final puzzle, we need to fill in the gap resulting from the eroded borders. For this reason we first repair the eroded edges by generating the band of new pixels all around the given patch. To do this we use an image extension technique called Boundless [77]. The idea is to extrapolate the image of the patch in all directions, to cover the void created by the erosion.

The Boundless is a GAN-based model tailored to extend the image content along any direction, i.e. to fill the image content outside the original boundaries. For our task, we use the pre-trained model on Places [95] provided by Google[1]. The limitation of the model is that it is trained to extend the image in one direction (right). In order to extend the images of the puzzle pieces all around, we pass each piece through the generator four times rotating it by 90° sequentially.

Formally, given the $\tilde{i}$-th piece of a puzzle, its extended version is denoted by

$$i = \Phi(\tilde{i}, \beta, \theta) \tag{6.7}$$

where β is the percentage of image extension, and $\Phi(...)$ is the Boundless model parametrized by θ. Once the damaged borders get repaired, we can use the reconstructed patches to calculate the patch compatibility.

6.3.2 Compatibility of patches

The compatibility measure quantifies the affinity between pieces and predicts the likelihood of two patches to be neighbors. We measure the piece affinity by computing the dissimilarity between the abutting boundary pixels of two

[1]Pretrained model from TensowrflowHub https://www.tensorflow.org/hub/tutorials/boundless

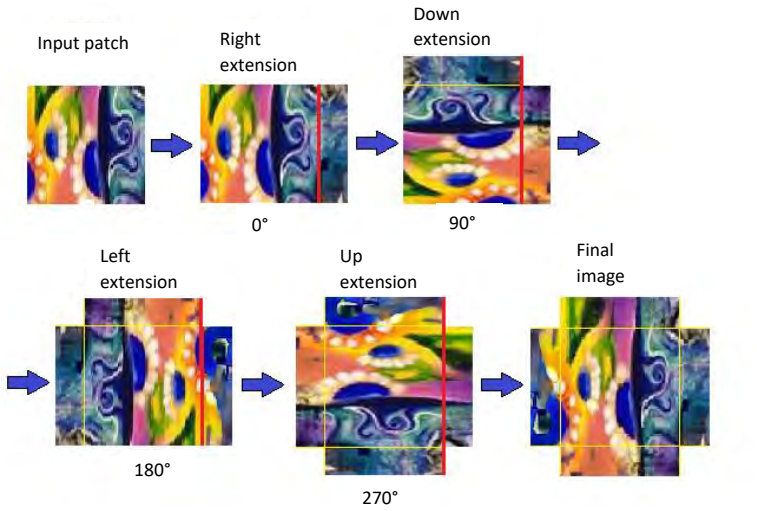

Figure 6.4: Border extension procedure: we rotate the patch in order to extend the image in all directions (right, up, left and down)

adjacent pieces; to this end, we adopt the *Mahalanobis Gradient Compatibility* (MGC) developed by Gallagher [30] and further improved by Son et al. [74]. MGC considers both the color differences across pieces borders and the directional derivative differences along the borders.

Once the pairwise dissimilarity scores are calculated for each pair of pieces in all possible neighboring relationships (right, up, left, down), we convert them to normalized compatibility values, as follows:

$$C_{\mathcal{R}}(i,j) = \max\left(1 - \frac{\Gamma_R(i,j)}{K_{min_{\mathcal{R}}}(i)}, 0\right) \tag{6.8}$$

where $K_{min_{\mathcal{R}}}(i)$ is the K-min value of the dissimilarity between all other pieces in relation $\mathcal{R}$ to piece i. The smaller the value of K, the more sparse $C_{\mathcal{R}}(i,j)$ becomes, leading to a more efficient relaxation labeling process.

6.3.3 Puzzle Solver

We cast jigsaw puzzle solving as a consistent labeling problem, the detailed explanation is given in Section 4. In our formulation, the puzzle pieces are considered as a set of objects and their possible positions as a set of

labels. The puzzle problem is hence viewed as the problem of finding consistent labeling that satisfies certain compatibility relations, with an additional requirement for one-to-one correspondences between the puzzle's tiles and their positions. We solve the puzzle using classical relaxation labeling algorithm [60] that, starting from the uniform probability (barycentre point) distribution, progressively updates the assignment matrix till it converges to the consistent labeling, which in our case corresponds to a permutation matrix.

The set of objects B represents the puzzle pieces, the labels Λ are the positions in the reconstruction plane (hence $m = n$), and the task is to assign a different position from Λ to each puzzle piece from B.

The $\mathbf{P} \in \Delta^{n \times m}$ is a soft assignment matrix (where each row represents a probability distribution of the positions for a piece and each column represents a probability distribution of the pieces for a position), $\Delta^{n \times m}$ is the multi-simplex with

$\Delta^m = \{\mathbf{p}_i \mid p_{i\lambda} \geq 0 \wedge \sum_\lambda p_{i\lambda} = 1\}$ and $\Delta^n = \{\mathbf{p}_\lambda \mid p_{i\lambda} \geq 0 \wedge \sum_i p_{i\lambda} = 1\}$,

where $p_{i\lambda}$ is the probability of piece i to choose position λ. Thus $\mathbf{P} = p_{i\lambda}$ is doubly stochastic matrix such that $\sum_\lambda p_{i\lambda} = \sum_i p_{i\lambda} = 1$.

The relaxation labeling update rule guarantees that $\mathbf{P}$ is a stochastic matrix (i.e., rows sum to 1) but does not enforce the same constraint for its columns. Therefore, the optimization process can converge to a labeling that does not represent a permutation (producing a solution with multiple pieces assigned the same position and vice versa).

To alleviate this problem and enforce one-to-one correspondence constraints, we endow the relaxation process with matrix balancing algorithm, adopting Sinkhorn-Knopp (SK) normalization [73]. SK algorithm transforms a given non-negative square matrix to its related doubly stochastic version, by alternately normalizing the rows and columns. SK is incorporated in our algorithm as an additional balancing step in each iteration.

6.4 Experiments & Results

6.4.1 Datasets

We assess the validity of our approach considering two benchmarks. First, we test our method on a large dataset of small (synthetic) images. Following JigsawGAN [45] we create our collection of 1600 images randomly picked up from PACS dataset [44]. Our collection is divided into 4 object categories (elephant, guitar, person, house), each of which covers 4 image styles (paintings, photos, cartoons, and sketches). Each of 1600 images is cut into 72x72 pixels size pieces generating a 9-pieces puzzle (3x3).

For the second test, we apply our method to three datasets [14, 61], widely used as performance benchmarks; each contains 20 images of increasing size. We cut the images into equal size pieces, generating puzzles of 70, 88, and 150 pieces (for the 1st, 2nd, and 3rd data sets respectively).

6.4.2 Accuracy metrics

To evaluate the performance of the algorithm we adopt three accuracy measures, widely used in literature: *Direct Comparison* metric, which measures the ratio of pieces placed in the correct position; the *Neighbor Comparison* metric that measures the ratio of correctly assigned neighbors in the solution; the *Perfect Reconstruction* metric represents the ratio of perfectly solved puzzles, where *perfectly solved* means that all pieces are placed in the correct position.

6.4.3 Experiments

We performed experiments on the two aforementioned benchmarks considering the three different metrics and an increasing level of border erosion, $\beta \in \{0\%, 7\%, 14\%\}$. We compare the results of our JiGAN approach to our previous model without GAN, namely Relaxation labeling puzzle-solver (RL) [40]. Concerning [10], although the idea is similar to ours, their model involves much more information (all possible pairing and rotation of puzzle's pieces), thus a direct comparison would not be fair. We do not report the performances of JiGAN when there is no border erosion ($\beta = 0\%$) since the result are the same as the one of RL [40].

Experiments with PACS dataset (small puzzles)

Using the PACs dataset, we conduct two types of experiments: first, we generate 3x3 puzzles without any gap between pieces and run the relaxation labeling (RL) solver [40]; second, to simulate the erosion of the boards, we generate the puzzles with gaps between pieces with two different levels of erosion 7% and 14% gaps. We compare two methods: the RL algorithm without the image extension step, and our JiGAN procedure that involves the completion of the eroded border. Both procedures are applied for two-levels of border erosion, 7%, and 14% of the size of the piece.

"Direct" accuracy	*no gap* **RL**	*no gap* **JiGAN**	*7% gap* **RL**	*7% gap* **JiGAN**	*14% gap* **RL**	*14% gap* **JiGAN**
house	92%	-	57%	**74%**	41%	**60%**
elephant	88%	-	51%	**74%**	30%	**54%**
guitar	83%	-	42%	**65%**	26%	**48%**
person	90%	-	56%	**72%**	40%	**58%**
Mean	88%	-	50%	**70%**	32%	**53%**

Table 6.1: Reconstruction performance of puzzles from the PACS datasets. *"Direct accuracy"* metrics.

"Perfect" accuracy	*no gap* **RL**	*no gap* **JiGAN**	*7% gap* **RL**	*7% gap* **JiGAN**	*14% gap* **RL**	*14% gap* **JiGAN**
house	90%	-	46%	**64%**	26%	**42%**
elephant	86%	-	41%	**64%**	16%	**36%**
guitar	77%	-	33%	**49%**	13%	**27%**
person	89%	-	51%	**65%**	28%	**43%**
Mean	85%	-	41%	**60%**	19%	**35%**

Table 6.2: Reconstruction performance of puzzles from the PACS datasets. *"Perfect accuracy"* metrics.

Table 6.1 and table 6.2 show the results of puzzle reconstruction in terms of direct comparison accuracy measure and perfect reconstruction ratio. Figures 6.6, 6.7 and 6.8 illustrate some qualitative results for reconstruction of small puzzles with different levels of erosion. It can be seen that, for the case without gaps, our solver performs well in all categories. While in the cases with erosion, the performance of the solver algorithm decreases as the level of erosion increases. However, the image extension step is beneficial to puzzle reconstruction with respect to the algorithm without extension.

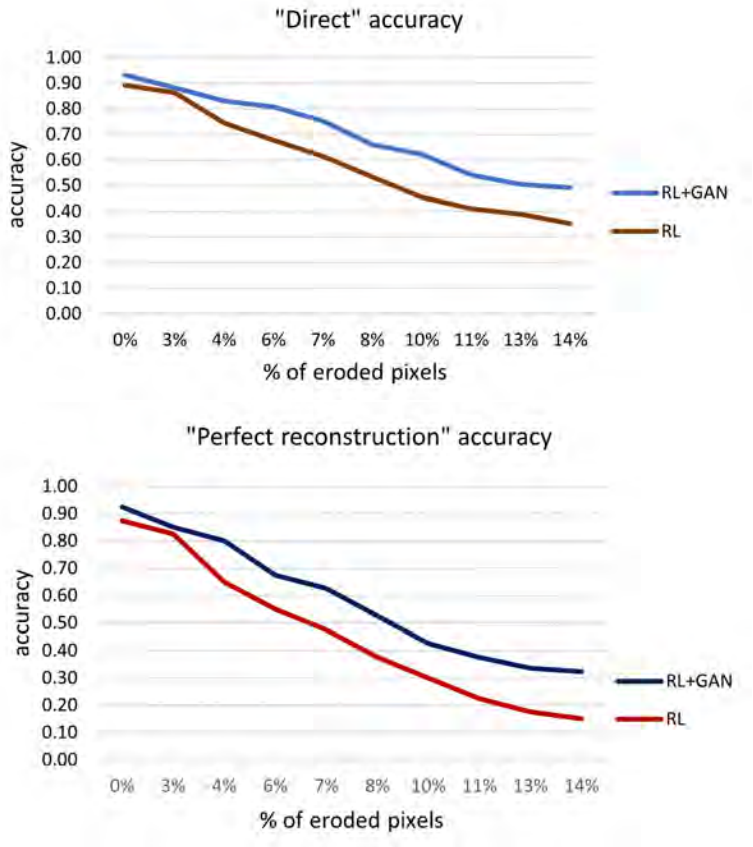

Figure 6.5: JiGAN (blue) vs RL (red) models: average Direct (a) and Perfect (b) accuracy then increasing the erosion gaps β.

Nevertheless, the performance of the model degrades with larger gaps and negatively influences the accuracy of the solver. To further investigate this degradation effect, we perform additional experiments by gradually increasing the erosion gaps (1%, 3%, ..., 14%) and observing the accuracy of the algorithm with and without extension steps. Figure 6.5 illustrates the performance of the solver applied to 400 randomly selected puzzles with different levels of erosion. As expected, the larger the erosion, the less accurate the results.

It is also worth noticing that the performance of the solver could be enhanced in absence of sketch images in a dataset. The puzzles generated from sketch images are particularly challenging to solve as they contain scarce

lines and no color information, leading to false high compatibility scores between all parts of the puzzle. In fact, we notice that 60% of the puzzles from the sketch category fail to be reconstructed (with a direct accuracy score equal to 0); this negatively influences the final results, in terms of averages.

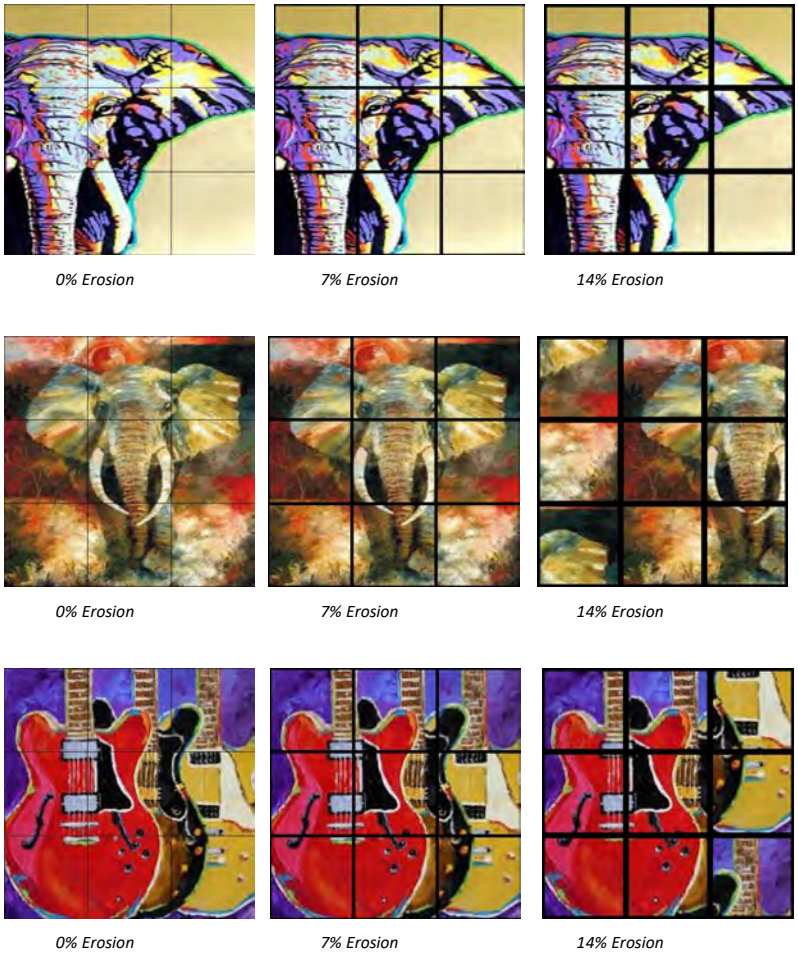

Figure 6.6: Qualitative results for small puzzles from Pacs dataset (elephant, guitar) with 0%, 7%, 14% erosion of piece size

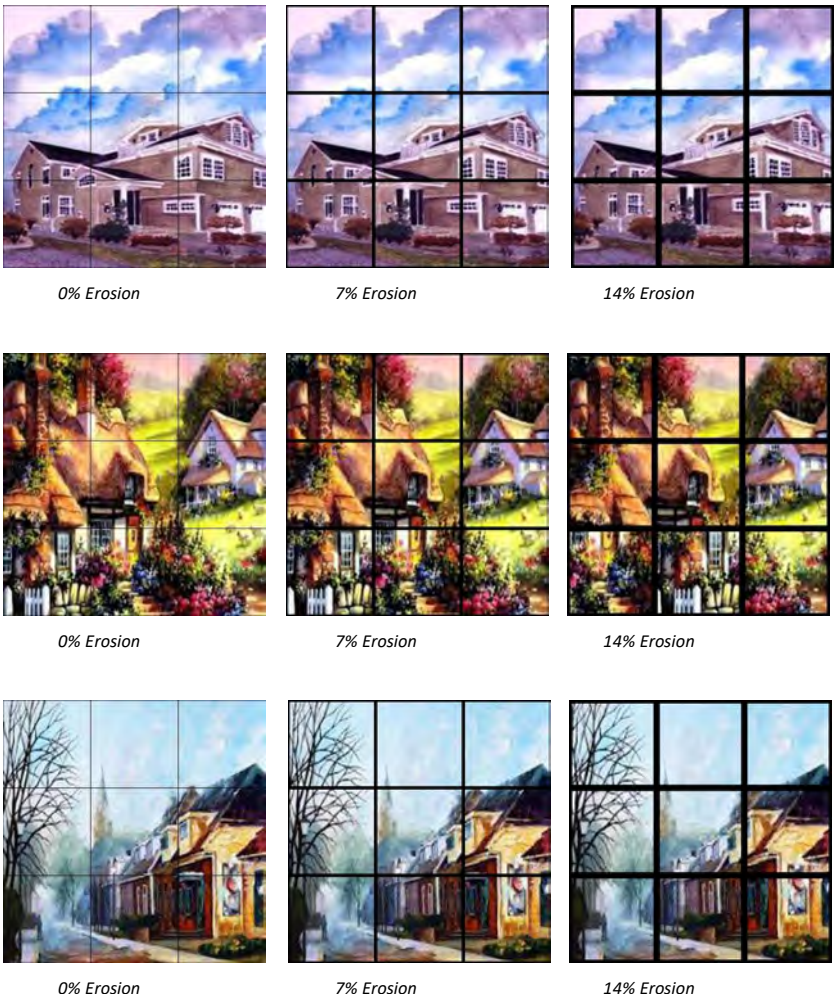

Figure 6.7: Qualitative results for small puzzles from Pacs dataset (house) with 0%, 7%, 14% erosion of piece size

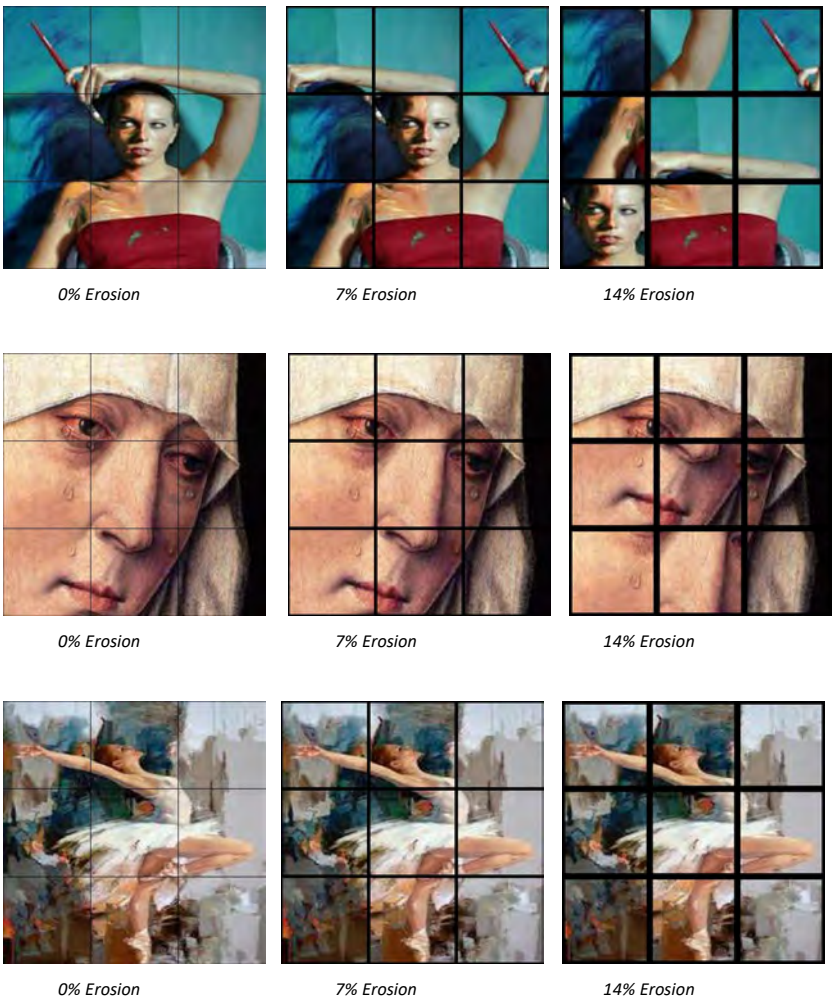

Figure 6.8: Qualitative results for small puzzles from Pacs dataset (person) with 0%, 7%, 14% erosion of piece size

Experiments with Benchmark datasets

For further evaluation, we apply our method to the large puzzles generated from the three benchmark datasets. As before, we conduct two experiments applying erosion of 7% and 14% of piece size. Tables 6.3 and 6.4 shows the results of the RL solver run without reconstruction of the eroded border and the results of the puzzle solver after the GAN image extension algorithm is applied.

Similarly to the case with small puzzles, the larger the erosion gaps, the lower the accuracy of the puzzle solution. The performance of the GAN model gradually degrades with the larger area of generated pixels. However, applying the inpainting algorithm significantly increases the accuracy of puzzle reconstruction concerning the results of the solver without image extension.

"Direct" accuracy	*no gap* **RL**	*no gap* **JiGAN**	*7% gap* **RL**	*7% gap* **JiGAN**	*14% gap* **RL**	*14% gap* **JiGAN**
70 pieces (MIT)	97%	-	22%	**51%**	11%	**32%**
88 pieces (McGill)	99%	-	23%	**59%**	7%	**31%**
150 pieces (Pomeranz805)	99%	-	12%	**38%**	6%	**15%**
Mean	98%	-	19%	**49%**	8%	**26%**

Table 6.3: Reconstruction performance of puzzles from the MIT, McGill and Pomeranz805 datasets. *"Direct accuracy"* metrics.

"Neighbour" accuracy	*no gap* **RL**	*no gap* **JiGAN**	*7% gap* **RL**	*7% gap* **JiGAN**	*14% gap* **RL**	*14% gap* **JiGAN**
70 pieces (MIT)	97%	-	46%	**66%**	35%	**45%**
88 pieces (McGill)	99%	-	46%	**65%**	30%	**40%**
150 pieces (Pomeranz805)	99%	-	41%	**54%**	28%	**33%**
Mean	98%	-	45%	**62%**	31%	**39%**

Table 6.4: Reconstruction performance of puzzles from the MIT, McGill and Pomeranz805 datasets. *"Neighbour accuracy"* metrics.

Figures 6.11 and 6.12 illustrate some qualitative results for reconstruction of big puzzles with different levels of erosion. It can be seen that without erosion we obtain the perfect reconstruction in most of the cases; for images with 7% of erosion gap, the overall result is good, however, the images have some errors most of which are minor and negligible to human eyes.

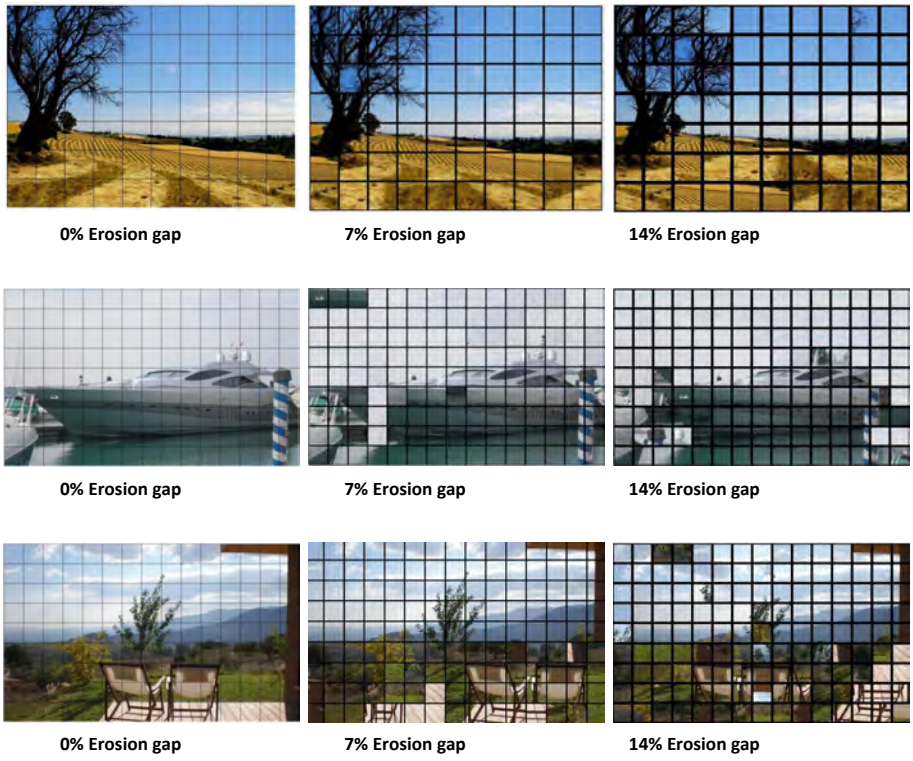

Figure 6.9: Qualitative results for big puzzles from Benchmark dataset (0%, 7%, 14% erosion of piece size)

As it can be expected, the results of reconstruction of images with 14% of erosion are less accurate than those with 7% of erosion. Though in some examples the misplaced patches make it difficult the perception the image; in other cases, the reconstruction results are acceptable for the human eye.

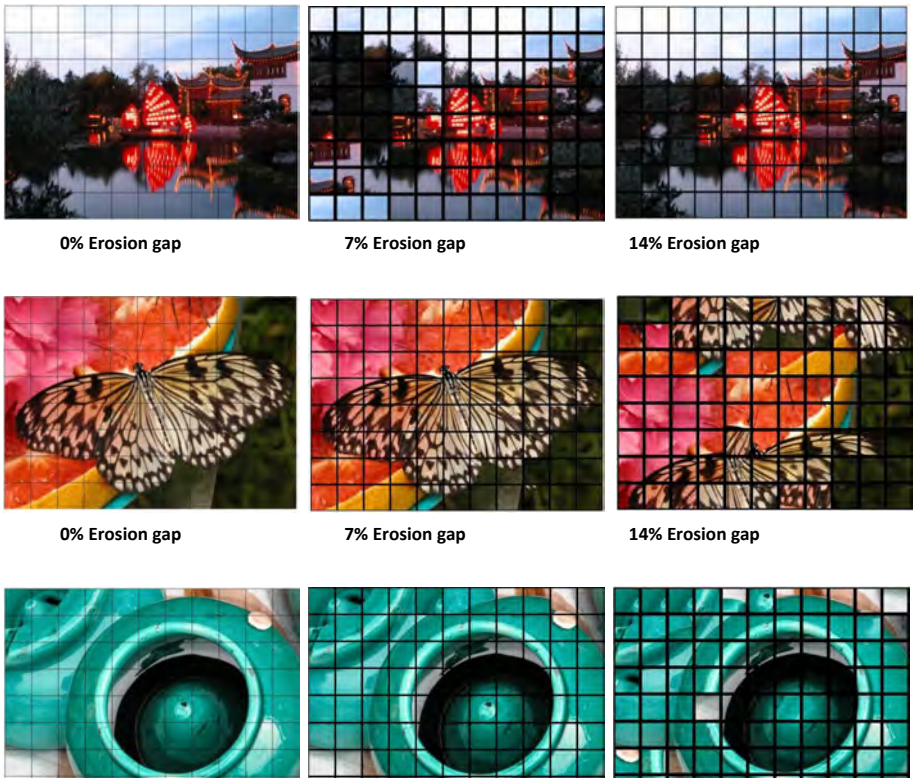

Figure 6.10: Qualitative results for big puzzles from Benchmark dataset (0%, 7%, 14% erosion of piece size)

6.5 Experiments with Fresco dataset

The motivation of this thesis is the reconstruction of the broken archaeological artifacts and in particular re-paring the ruined frescoes. For this reason, we also wanted to test our methods on a syntactical dataset of wall-painted images.

The texture of such images is different from the natural photos. When the images are hand-painted, the same color nuance and same brushstroke patterns can be presented all over the picture. That creates difficulties in extrapolating the patches in the case of a ruined border. Moreover, the spurious colors presented in some paintings are difficult to foresee, and the image extension model can fail in extrapolating these parts. Ideally, the GAN model must be retrained for every particular painting style. But, beyond being time-consuming, it is often infeasible, as the authorship of the ruined fresco is unknown and the data is not available.

However it is, we decided to test our Gan-based model and puzzle-solving method on a syntactical dataset of wall-painted images. To this end, we create our fresco puzzle dataset by selecting 20 examples from the DAFNE dataset of famous frescos [25]. The chosen artworks have been painted by famous Italian artists such as Giotto, Masaccio, Piero Della Francesca, and Michelangelo (14th - 18th century). For each selected fresco we cropped a fragment of size 1000x1000 pixels and shuffled each image into 100 patches (100x100 pixels). As previously, for each image we generate 3 types of puzzles: first without erosion and two with gaps between pieces, emulating two different levels of erosion 7% and 14%. We conduct two experiments: first, we test our relaxation labeling puzzle solver without the image extension step; second, we test the JiGan method (which first extrapolates the eroded border and then applies the RL solver on the repaired patches).

Table 6.5 and 6.6 present the result of the experiments in terms of the "Direct" and "Neighbour" accuracy.

It can be seen that the quality of reconstruction of images drops significantly with the level of degradation of the border: 43% of accuracy (for 7% of erosion) vs 17% of accuracy (for 14% of erosion). That basically confirms that the image extension model can be highly inaccurate for these types of images in extrapolating further from the border. Interestingly, the results of the reconstruction without gaps are perfect for all images in the dataset.

The figure 6.11 and 6.12 present some qualitative results of reconstruction of the fresco image with different levels of the erosion of the border. It can be seen that images with 14% of erosion often contain numerous misplaced patches. However, even if the accuracy of final reconstruction is not

high, the partially reconstructed segments of the fresco can be helpful as initial input for further reconstruction in human-computer interaction mode. Partially reconstructed elements of fresco can give to the archaeologists the idea of the whole composition of the painting, and after fixing some correctly reconstructed sectors, those can be used as anchors for the further run of puzzle-solving algorithm.

"Direct" accuracy	*no gap*	*7% gap*	*14% gap*
RL puzzle solve	100%	29%	10%
JiGAN puzzle solver	**100%**	**43%**	**17%**

Table 6.5: Reconstruction performance of puzzles from the Fresco dataset. *"Direct accuracy"* metrics.

"Neighbour" accuracy	*no gap*	*7% gap*	*14% gap*
RL puzzle solver	100%	47%	28%
JiGAN puzzle solver	**100%**	**56%**	**33%**

Table 6.6: Reconstruction performance of puzzles from the Fresco dataset. *"Neighbour accuracy"* metrics.

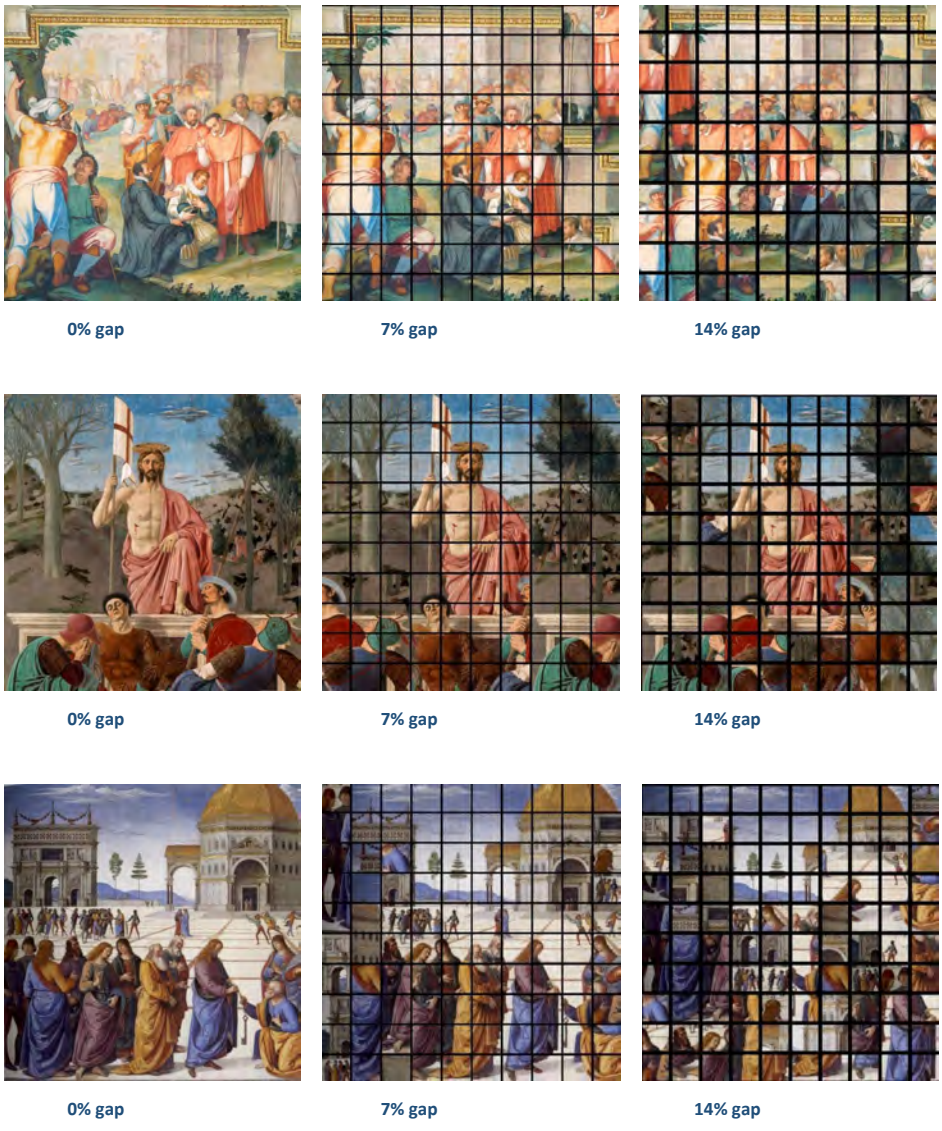

Figure 6.11: Qualitative results for puzzles from *Fresco* dataset (0%, 7%, 14% erosion of piece size)

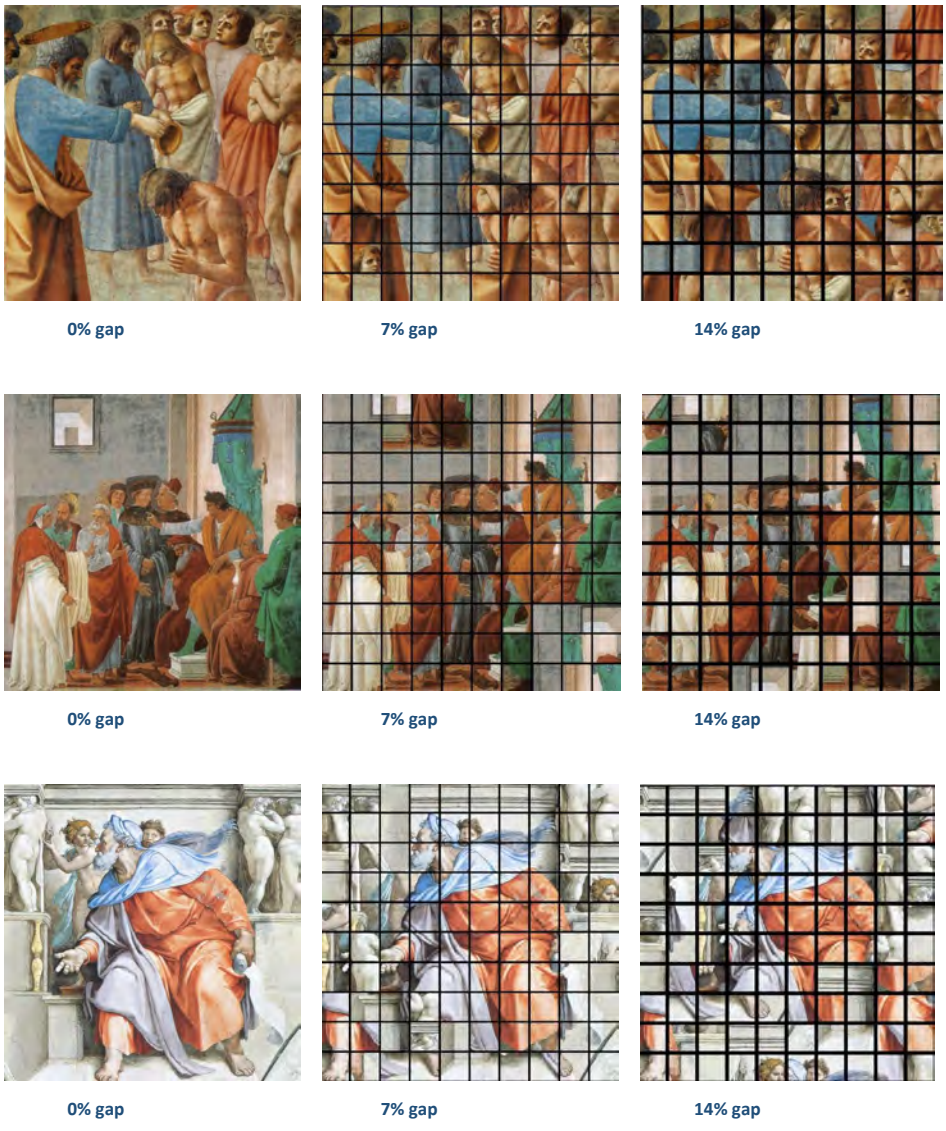

Figure 6.12: Qualitative results for puzzles from *Fresco* dataset (0%, 7%, 14% erosion of piece size)

Chapter 7

Conclusions

The idea to work on puzzle-solving comes thanks to its intimate relation to the problem of fresco reconstruction in archeology, as the reconstruction of fractured artifacts can be seen as a particularly challenging case of puzzle-solving.

In the first stage of the puzzle-solving project, we devise a novel and theoretically robust method to solve jigsaw puzzles. In our formulation, the puzzle problem is abstracted as a consistent labeling problem and is solved using a relaxation labeling algorithm, endowed with matrix balancing mechanisms to enforce one-to-one correspondence. The results of preliminary experiments attest the validity of the approach.

In the second stage of the research project, we extend our previous puzzle-solving method to handle the challenging task of solving puzzles with ruined borders. We see indeed that the previous methods, based on the compatibility calculated on the color gradient across the edges, solve effectively the puzzles without gaps, but their performance immediately drops in the presence of erosion gaps.

To overcome such problem, we introduce the idea of repairing the damaged patches using the GAN model for image extension. We apply the extension procedure on each patch separately, thus avoiding computationally expensive inpainting for all combinations in pairs. We use a GAN-based image extension model to generate the missing pieces around each patch, then we calculate the compatibility between the repaired patch and apply the puzzle-solving algorithm.

We show that the combination of solving algorithm and deep learning model can be a viable solution to reconstruct a puzzle with ruined regions. Our two-steps procedure produces better results compared to the previous

method. However, the quality of the final reconstruction depends on the level of degradation: the larger the erosion gap, the worse the result. Nevertheless, the overall results with a moderate level of erosion are generally acceptable to human eyes.

Although, the theory and proprieties of relaxation labeling have been deeply studied for the case of stochastic labeling only, its application on puzzle-solving confirms its validity also in the case of double stochastic matrix. However, it would be useful and interesting further research in this direction, especially to understand deterministically the behavioral proprieties of relaxation labeling in the presence of doubly stochastic constraint. Moreover, a natural extension enforces the balancing step by directly embedding it in the update rule, devising optimization and relaxation labeling schemes that are more suitable for puzzle solving.

Furthermore in light of our empirical results, it appears clear that the compatibility measure, being the crucial point of the solver, must be improved. To this end, we believe that the use of automatic feature extractors such as neural networks, with the recent advent of self-training techniques, represents an interesting branch of research to be explored.

Bibliography

[1] On the automatic assemblage of arbitrary broken solid artefacts. Image and Vision Computing **21**(5), 401–412 (2003). https://doi.org/https://doi.org/10.1016/S0262-8856(03)00008-8

[2] Andaló, F.A., Taubin, G., Goldenstein, S.: PSQP: puzzle solving by quadratic programming. IEEE TPAMI **39**(2), 385–396 (2017)

[3] Baraldi, L., Cornia, M., Grana, C., Cucchiara, R.: Aligning text and document illustrations: Towards visually explainable digital humanities. In: 2018 24th International Conference on Pattern Recognition (ICPR). pp. 1097–1102 (2018)

[4] Barker, M., Mazarico, E., Neumann, G., Zuber, M., Haruyama, J., Smith, D.: A new lunar digital elevation model from the lunar orbiter laser altimeter and selene terrain camera. Icarus **273**, 346 – 355 (2016)

[5] Barnes, C., Shechtman, E., Finkelstein, A., Goldman, D.B.: PatchMatch: A randomized correspondence algorithm for structural image editing. ACM Transactions on Graphics (Proc. SIGGRAPH) **28**(3) (Aug 2009)

[6] Benito-Calvo, A., Crittenden, A.N., Livengood, S.V., Sánchez-Romero, L., Martínez-Fernández, A., de la Torre, I., Pante, M.: 3d 360° surface morphometric analysis of pounding stone tools used by hadza foragers of tanzania: A new methodological approach for studying percussive stone artefacts. Journal of Archaeological Science: Reports **20**, 611 – 621 (2018)

[7] Birkhoff, G.: Three observations on linear algebra. Univ. Nac. Tacuman, Rev. Ser. A **5**, 147–151 (1946)

[8] Boyd, S.P., Dattorro, J.C.: Alternating projections

[9] Brandão, S., Marques, M.: Hot tiles: A heat diffusion based descriptor for automatic tile panel assembly. In: Hua, G., Jégou, H. (eds.) ECCV Workshops. vol. 9913, pp. 768–782. Springer (2016)

[10] Bridger, D., Danon, D., Tal, A.: Solving jigsaw puzzles with eroded boundaries (2019)

[11] Carneiro, G., da Silva, N.P., Del Bue, A., Costeira, J.P.: Artistic image classification: An analysis on the printart database. In: European Conference on Computer Vision. pp. 143–157. Springer (2012)

[12] Carraggi, A., Cornia, M., Baraldi, L., Cucchiara, R.: Visual-semantic alignment across domains using a semi-supervised approach. In: The European Conference on Computer Vision (ECCV) Workshops (September 2018)

[13] Chen, L., Chen, J., Zou, Q., Huang, K., Li, Q.: Multi-view feature combination for ancient paintings chronological classification. J. Comput. Cult. Herit. **10**(2), 7:1–7:15 (Mar 2017)

[14] Cho, T.S., Avidan, S., Freeman, W.T.: A probabilistic image jigsaw puzzle solver. In: Proc. CVPR. pp. 183–190 (2010)

[15] Chollet, F.: Xception: Deep learning with depthwise separable convolutions. CoRR **abs/1610.02357** (2016)

[16] Christmas, W.J., Kittler, J., Petrou, M.: Structural matching in computer vision using probabilistic relaxation. IEEE Trans. Pattern Anal. Mach. Intell. **17**, 749–764 (1995)

[17] Clevert, D.A., Unterthiner, T., Hochreiter, S.: Fast and accurate deep network learning by exponential linear units (elus) (2016)

[18] Cruz, R.S., Fernando, B., Cherian, A., Gould, S.: Deeppermnet: Visual permutation learning. CoRR **abs/1704.02729** (2017)

[19] Deever, A., Gallagher, A.: Semi-automatic assembly of real cross-cut shredded documents. In: Proc. ICIP. pp. 233–236 (2012)

[20] Demaine, E.D., Demaine, M.L.: Jigsaw puzzles, edge matching, and polyomino packing: Connections and complexity. Graphs Comb. **23**(Suppl. 1), 195–208 (2007)

[21] Deng, J., Dong, W., Socher, R., Li, L., Kai Li, Li Fei-Fei: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. pp. 248–255 (June 2009)

[22] Denton, E.L., Chintala, S., Szlam, A.D., Fergus, R.: Deep generative image models using a laplacian pyramid of adversarial networks. In: NIPS (2015)

[23] Derech, N., Tal, A., Shimshoni, I.: Solving archaeological puzzles. CoRR **abs/1812.10553** (2018)

[24] Disser, A., Dillmann, P., Leroy, M., L'Héritier, M., Bauvais, S., Fluzin, P.: Iron supply for the building of metz cathedral: New methodological development for provenance studies and historical considerations. Archaeometry **59**(3), 493–510 (2017)

[25] Dondi, P., Lombardi, L., Setti, A.: Dafne: A dataset of fresco fragments for digital anastylosis. Pattern Recognit. Lett. **138**, 631–637 (2020)

[26] Doyon, L.: On the shape of things: A geometric morphometrics approach to investigate aurignacian group membership. Journal of Archaeological Science **101**, 99 – 114 (2019)

[27] Elfving, T., Eklundh, J.O.: Some properties of stochastic labeling procedures. Comput. Graph. Image Process. **20**, 158–170 (1982)

[28] Elgammal, A., Kang, Y., Leeuw, M.D.: Picasso, matisse, or a fake? automated analysis of drawings at the stroke level for attribution and authentication. In: AAAI. pp. 42–50. AAAI Press (2018)

[29] Faugeras, O.D., Berthod, M.: Improving consistency and reducing ambiguity in stochastic labeling: An optimization approach. IEEE Transactions on Pattern Analysis and Machine Intelligence **PAMI-3**, 412–424 (1981)

[30] Gallagher, A.C.: Jigsaw puzzles with pieces of unknown orientation. In: Proc. CVPR. pp. 382–389 (2012)

[31] Gallwey, J., Eyre, M., Tonkins, M., Coggan, J.: Bringing lunar lidar back down to earth: Mapping our industrial heritage through deep transfer learning. Remote Sensing **11**(17) (2019)

[32] Gonthier, N., Gousseau, Y., Ladjal, S., Bonfait, O.: Weakly supervised object detection in artworks. In: Leal-Taixé, L., Roth, S. (eds.)

Computer Vision – ECCV 2018 Workshops. pp. 692–709. Springer International Publishing, Cham (2019)

[33] Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems. vol. 27. Curran Associates, Inc. (2014), https://proceedings.neurips.cc/paper/2014/file/5ca3e9b122f61f8f06494c97b1afccf3-Paper.pdf

[34] Gultepe, E., Conturo, T.E., Makrehchi, M.: Predicting and grouping digitized paintings by style using unsupervised feature learning. Journal of Cultural Heritage **31**, 13 – 23 (2018)

[35] Gur, S., Ben-Shahar, O.: From square pieces to brick walls: The next challenge in solving jigsaw puzzles. In: ICCV. pp. 4029–4037 (2017)

[36] Haralick, R.M., Shapiro, L.G.: The consistent labeling problem: Part I. IEEE TPAMI **1**(2), 173–184 (1979)

[37] Huang, Q., Flöry, S., Gelfand, N., Hofer, M., Pottmann, H.: Reassembling fractured objects by geometric matching. ACM Trans. Graph. **25**, 569–578 (2006)

[38] Hummel, R.A., Zucker, S.W.: On the foundations of relaxation labeling processes. IEEE TPAMI **5**(3), 267–287 (1983)

[39] Karayev, S., Trentacoste, M., Han, H., Agarwala, A., Darrell, T., Hertzmann, A., Winnemöller, H.: Recognizing image style (2014)

[40] Khoroshiltseva, M., Vardi, B., Torcinovich, A., Traviglia, A., Ben-Shahar, O., Pelillo, M.: Jigsaw puzzle solving as a consistent labeling problem. In: Computer Analysis of Images and Patterns. pp. 392–402. Springer International Publishing (2021)

[41] Kittler, J., Hancock, E.R.: Combining evidence in probabilistic relaxation. Int. J. Pattern Recognit. Artif. Intell. **3**, 29–52 (1989)

[42] Knight, P.A.: The sinkhorn–knopp algorithm: Convergence and applications. SIAM Journal on Matrix Analysis and Applications **30**(1), 261–275 (2008)

[43] Levy, M.: A new theoretical approach to relaxation, application to edge detection. [1988 Proceedings] 9th International Conference on Pattern Recognition pp. 208–212 vol.1 (1988)

[44] Li, D., Yang, Y., Song, Y.Z., Hospedales, T.M.: Deeper, broader and artier domain generalization. In: Proceedings of the IEEE International Conference on Computer Vision (ICCV) (Oct 2017)

[45] Li, R., Liu, S., Wang, G., Liu, G., Zeng, B.: Jigsawgan: Self-supervised learning for solving jigsaw puzzles with generative adversarial networks. CoRR **abs/2101.07555** (2021)

[46] Lopes, F., Lima, A., de Matos, A.P., Custódio, J., Cagno, S., Schalm, O., Janssens, K.: Characterization of 18th century portuguese glass from real fábrica de vidros de coina. Journal of Archaeological Science: Reports **14**, 137 – 145 (2017)

[47] Lyzinski, V., Fishkind, D.E., Fiori, M., Vogelstein, J.T., Priebe, C.E., Sapiro, G.: Graph matching: Relax at your own risk. IEEE Trans. Pattern Anal. Mach. Intell. **38**(1), 60–73 (2016)

[48] Mathieu, M., Couprie, C., LeCun, Y.: Deep multi-scale video prediction beyond mean square error. CoRR **abs/1511.05440** (2016)

[49] Mathieu, M., Zhao, J.J., Sprechmann, P., Ramesh, A., LeCun, Y.: Disentangling factors of variation in deep representation using adversarial training. ArXiv **abs/1611.03383** (2016)

[50] Mena, G., Belanger, D., Linderman, S., Snoek, J.: Learning latent permutations with gumbel-sinkhorn networks (2018)

[51] Mensink, T., van Gemert, J.: The rijksmuseum challenge: Museum-centered visual recognition (2014)

[52] Miller, D.A., Zucker, S.W.: Copositive-plus Lemke algorithm solves polymatrix games. Oper. Res. Lett. **10**, 285–290 (1991)

[53] Noroozi, M., Favaro, P.: Unsupervised learning of visual representations by solving jigsaw puzzles. CoRR **abs/1603.09246** (2016)

[54] van den Oord, A., Kalchbrenner, N., Kavukcuoglu, K.: Pixel recurrent neural networks (2016)

[55] Paikin, G., Tal, A.: Solving multiple square jigsaw puzzles with missing pieces. In: Proc. CVPR. pp. 4832–4839 (2015)

[56] Papaioannou, G., Karabassi, E.A., Theoharis, T.: Reconstruction of three-dimensional objects through matching of their parts. IEEE Transactions on Pattern Analysis and Machine Intelligence **24**(1), 114–124 (2002). https://doi.org/10.1109/34.982888

[57] Pathak, D., Krähenbühl, P., Donahue, J., Darrell, T., Efros, A.A.: Context encoders: Feature learning by inpainting. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 2536–2544 (2016)

[58] Paumard, M., Picard, D., Tabia, H.: Deepzzle: Solving visual jigsaw puzzles with deep learning and shortest path optimization. CoRR **abs/2005.12548** (2020)

[59] Peleg, S.: A new probabilistic relaxation scheme. IEEE Transactions on Pattern Analysis and Machine Intelligence **PAMI-2**(4), 362–369 (1980). https://doi.org/10.1109/TPAMI.1980.4767035

[60] Pelillo, M.: The dynamics of nonlinear relaxation labeling processes. J. Math. Imag. Vis. **7**(4), 309–323 (1997)

[61] Pomeranz, D., Shemesh, M., Ben-Shahar, O.: A fully automated greedy square jigsaw puzzle solver. In: Proc. CVPR. pp. 9–16 (2011)

[62] Prieto, A.J., Silva, A., de Brito, J., Macías-Bernal, J.M., Alejandre, F.J.: Multiple linear regression and fuzzy logic models applied to the functional service life prediction of cultural heritage. Journal of Cultural Heritage **27**, 20 – 35 (2017)

[63] Radford, A., Metz, L., Chintala, S.: Unsupervised representation learning with deep convolutional generative adversarial networks. CoRR **abs/1511.06434** (2016)

[64] Reed, S.E., Akata, Z., Mohan, S., Tenka, S., Schiele, B., Lee, H.: Learning what and where to draw. CoRR **abs/1610.02454** (2016)

[65] Rosenfeld, A., Hummel, R.A., Zucker, S.W.: Scene labeling by relaxation operations. IEEE Trans. Syst. Man & Cybern. **6**, 420–433 (1976)

[66] Sabatelli, M., Kestemont, M., Daelemans, W., Geurts, P.: Deep transfer learning for art classification problems. In: Computer Vision - ECCV 2018 Workshops - Munich, Germany, September 8-14, 2018, Proceedings, Part II. pp. 631–646 (2018)

[67] Saleh, B., Elgammal, A.M.: Large-scale classification of fine-art paintings: Learning the right metric on the right feature. CoRR **abs/1505.00855** (2015)

[68] Sharafi, S., Fouladvand, S., Simpson, I., Alvarez, J.A.B.: Application of pattern recognition in detection of buried archaeological sites based on analysing environmental variables, khorramabad plain, west iran. Journal of Archaeological Science: Reports **8**, 206 – 215 (2016)

[69] Shen, X., Efros, A.A., Aubry, M.: Discovering visual patterns in art collections with spatially-consistent feature learning. CoRR **abs/1903.02678** (2019)

[70] Sholomon, D., David, O.E., Netanyahu, N.S.: A generalized genetic algorithm-based solver for very large jigsaw puzzles of complex types. In: Proc. AAAI. pp. 2839–2845 (2014)

[71] Silburt, A., Ali-Dib, M., Zhu, C., Jackson, A., Valencia, D., Kissin, Y., Tamayo, D., Menou, K.: Lunar crater identification via deep learning. Icarus **317**, 27 – 38 (2019)

[72] Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. CoRR **abs/1409.1556** (2014)

[73] Sinkhorn, R., Knopp, P.: Concerning nonnegative matrices and doubly stochastic matrices. Pacific J. Math. **21**(2), 343–348 (1967)

[74] Son, K., Hays, J., Cooper, D.B.: Solving square jigsaw puzzle by hierarchical loop constraints. IEEE TPAMI **41**(9), 2222–2235 (2018)

[75] Strezoski, G., Worring, M.: Omniart: Multi-task deep learning for artistic data analysis. CoRR **abs/1708.00684** (2017)

[76] Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 2818–2826. IEEE Computer Society, Los Alamitos, CA, USA (jun 2016)

[77] Teterwak, P., Sarna, A., Krishnan, D., Maschinot, A., Belanger, D., Liu, C., Freeman, W.T.: Boundless: Generative adversarial networks for image extension (2019)

[78] Thomas, C., Kovashka, A.: Artistic object recognition by unsupervised style adaptation. CoRR **abs/1812.11139** (2018)

[79] Tran, D., Ranganath, R., Blei, D.: Hierarchical implicit models and likelihood-free variational inference. In: Guyon, I., Luxburg, U.V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R. (eds.) Advances in Neural Information Processing Systems. vol. 30. Curran Associates, Inc. (2017), https://proceedings.neurips.cc/paper/2017/file/6f1d0705c91c2145201df18a1a0c7345-Paper.pdf

[80] Ullman, S.: Relaxation and constrained optimization by local processes. Computer Graphics and Image Processing **10**(2), 115–125 (1979). https://doi.org/https://doi.org/10.1016/0146-664X(79)90045-5, https://www.sciencedirect.com/science/article/pii/0146664X79900455

[81] van der Vaart, W.V., Lambers, K.: Learning to look at lidar: The use of r-cnn in the automated detection of archaeological objects in lidar data from the netherlands. Journal of Computer Applications in Archaeology **2**(1), 31–40 (2019)

[82] Von Neumann, J.: On rings of operators. reduction theory. Annals of Mathematics **50**(2), 401–485 (1949)

[83] Von Neumann, J.: A certain zero-sum two-person game equivalent to the optimal assignment problem. Contributions to the Theory of Games **2**(0), 5–12 (1953)

[84] Vondrick, C., Pirsiavash, H., Torralba, A.: Generating videos with scene dynamics. In: NIPS (2016)

[85] Wang, H., He, Z., Huang, Y., Chen, D., Zhou, Z.: Bodhisattva head images modeling style recognition of dazu rock carvings based on deep convolutional network. Journal of Cultural Heritage **27**, 60 – 71 (2017)

[86] Wang, M., Deng, W.: Deep visual domain adaptation: A survey. Neurocomputing **312**, 135 – 153 (2018)

[87] Wilber, M.J., Fang, C., Jin, H., Hertzmann, A., Collomosse, J., Belongie, S.J.: Bam! the behance artistic media dataset for recognition beyond photography. CoRR **abs/1704.08614** (2017)

[88] Wilson, R.C., Hancock, E.R.: Hierarchical discrete relaxation. In: Perner, P., Wang, P., Rosenfeld, A. (eds.) Advances in Structural and Syntactical Pattern Recognition. Springer Berlin Heidelberg, Berlin, Heidelberg (1996)

[89] Wu, J., Zhang, C., Xue, T., Freeman, W.T., Tenenbaum, J.B.: Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. CoRR **abs/1610.07584** (2016)

[90] Xie, S., Girshick, R.B., Dollár, P., Tu, Z., He, K.: Aggregated residual transformations for deep neural networks. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR) pp. 5987–5995 (2017)

[91] Xu, Z., Wilber, M.J., Fang, C., Hertzmann, A., Jin, H.: Beyond textures: Learning from multi-domain artistic images for arbitrary style transfer. CoRR **abs/1805.09987** (2018)

[92] Yu, J., Lin, Z., Yang, J., Shen, X., Lu, X., Huang, T.: Free-form image inpainting with gated convolution (2019)

[93] Zhao, F., He, X., Zhang, Y., Lei, W., Ma, W., Zhang, C., Song, H.: A jigsaw puzzle inspired algorithm for solving large-scale no-wait flow shop scheduling problems. Appl. Intell. **50**(1), 87–100 (2020)

[94] Zhao, J.J., Mathieu, M., LeCun, Y.: Energy-based generative adversarial networks. In: ICLR (2017)

[95] Zhou, Lapedriza, Khosla, Oliva, Torralba: Places: A 10 million image database for scene recognition **40** (Jun 2018). https://doi.org/10.1109/tpami.2017.2723009

[96] Zhu, J., Krähenbühl, P., Shechtman, E., Efros, A.A.: Generative visual manipulation on the natural image manifold. CoRR **abs/1609.03552** (2016)