Seminar Report on

Understanding Synergy and Variations in Human vs Machine Cognition in Problem Solving

Sarthak Johnson Prasad (18CS10049)

Guided by : Prof. Aritra Hazra
Jointly Guided by : Prof. Partha Pratim Chakrabarti

Dept of Computer Science and Engineering
Indian Institute of Technology Kharagpur

Spring Semester, 2020-21
April 3, 2021

Table of Contents

Introduction

What is Problem Solving ?

Problem solving is referred to as the process of constructing and applying mental representations of problems to finding solutions to those problems that are encountered in nearly every context[1]. It consists of using generic or ad hoc methods in an orderly manner to find solutions to problems.[2]

Why Problem Solving ?

Problem solving skills are important both from a human and a machine's point of view. Humans learn how to solve simple problems from a very early age (learning to eat, make coordinated movements and communicate) – and as a person goes through life problem-solving skills are refined, matured and become more sophisticated (enabling them to solve more difficult problems). In the latter case, increasing efficiency in problem solving leads to a better machine.

Problem Solving : Human vs Machine

As quoted in *Human and Machine Problem Solving* by Gilhooly, K.J. "*Cognitive science is the study of intelligent systems, whether natural or artificial, and treats both organisms and computers as types of information-processing systems. Clearly, humans and typical current computers have rather different functional or cognitive architectures. Thus, insights into the role of cognitive architecture in performance may be gained by comparing typical human problem solving with efficient machine problem solving over a range of tasks*". These insights can help us build better machines/learners to solve sophisticated human problems. Thus, comparing similarities and dissimilarities between how a machine and a human approach a problem is important.

Why Jigsaw Puzzles ?

As per the Polya-Problem Solving Process [32], there are four steps in problem-solving, viz. understanding the problem, compiling a plan (dividing a plan), carrying out a plan (carrying out the plan) and checking back (looking back). Jigsaw puzzles are the perfect example of this. It uses motor cognition as well as the previous intelligence and structure. Apart from looking at a local orientation the solver also needs to take care of the global structure, which implies a deeper level of cognition.

Related Works

In this section we will look into the previous research on the problem we have chosen, i.e, Jigsaw Puzzles to have a greater insight on our problem and how various methods were devised to solve the same. The works are broadly divided into three parts mainly on the basis of shape of the pieces and how the sudden rise in popularity of Machine and Deep Learning influenced the solving process.

Works on Regular Shaped Jigsaw Puzzles

a) A Probabilistic Image Jigsaw Puzzle Solver[3]

It was one of the pioneer works of reconstructing an image from a set of square image patches, i.e, a jigsaw puzzle. A jigsaw puzzle problem has shown to be NP-complete when the pairwise affinity of the pieces is unreliable [1]. The paper focused on solving image jigsaw puzzles made up of square-shaped pieces. It also laid groundwork for addressing the patch-based image editing /synthesis problems in which image layout is a requirement.

Fig 1. Reconstructed images using the estimated local evidence.

It used a graphical model to solve the jigsaw problem, i.e. each patch location was considered to be the node in the graph and each patch was considered to be the node label. Thus, the whole problem got reduced to finding the most likely patch configuration in the graph (a probabilistic problem) which had already been addressed in Cho *et al.* [2].

Firstly, a number of pairwise patch compatibility metrics were explored and compared for the model. Five types of compatibility measures were compared, viz. dissimilarity-based compatibility, boosting-based compatibility, set-based compatibility, image statistics-based compatibility and the combination of a dissimilarity-based and image statistics-based compatibility as in Cho *et al.* [2]. It was observed that a dissimilarity-based compatibility metric is the most discriminative.

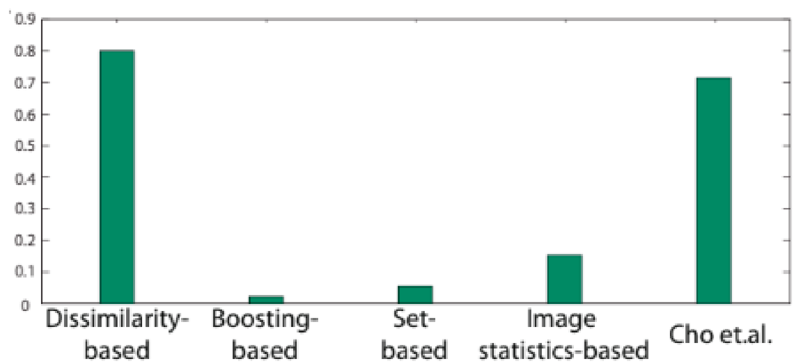


Fig 2. Types of Compatibility

Secondly, two strategies were evaluated to model the evidence term (which determines the image layout) in the graphical model, viz. dense-and-noisy evidence and sparse-and-accurate evidence. The first approach estimated a low resolution image from a set of patches. The second was to fix a small number of patches, called anchor patches, at their correct location, it served as the latter local evidence. Anchor patches were viewed as injected geometric information which was used to more accurately reconstruct images.

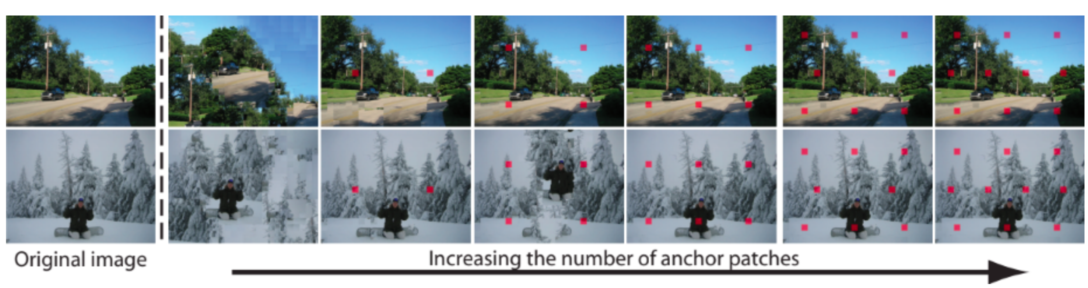


Fig 3. Figure shows two examples of reconstructed images with the latter local evidence.

Lastly, three measures were introduced to evaluate the puzzle reconstruction accuracy: Direct comparison, Cluster comparison, Neighbour comparison and it was shown that the algorithm could reconstruct real images reliably.

b) Solving Square Jigsaw Puzzles with Loop Constraints[10]

This paper proposed a computational puzzle solver for non overlapping square-piece jigsaw puzzles. It characterised the puzzles into two categories, viz. Type 1 and Type 2. In Type 1 puzzles the orientation of every piece is known and only the location of each piece is unknown whereas in Type 2 puzzles orientation of each piece is also unknown. Many algorithms had already been devised to solve the Type 1 puzzles. This paper proposed an algorithm to solve both Type 1 and Type 2 puzzles with no anchor points and no information about the dimensions of the puzzles.

The proposed solver reached a precision of 100% or precision 1 (perfect reconstruction) given the dissimilarity metrics which we discussed in the paper above. It was shown that the precision of matching pairs is more likely to increase as pieces or small loops of pieces are assembled to higher order loops and eventually reaches 1. The solver was able to reconstruct the image jigsaw puzzle perfectly for any dimension greater than 4 or 5, the configurations were always correct – piece pair matches in the assemblies were all true positives.

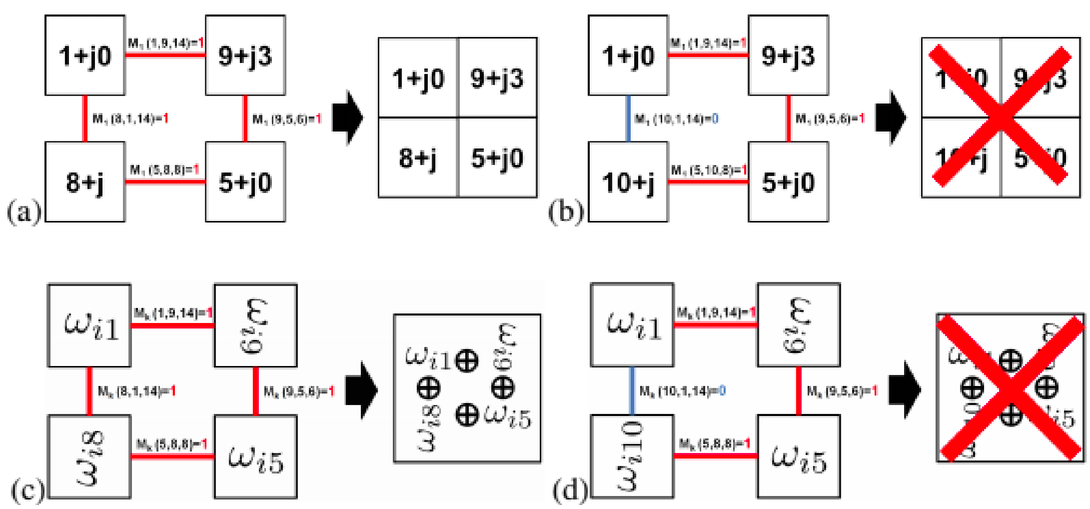


Fig 4. Recovering small loops of arbitrary order in the puzzle.

The solver outperformed state-of-the-art methods [3, 4, 5, 6, 7, 9] with the standard data sets [3, 8, 4]. For more challenging Type 2 puzzle setup, the solver reduced the error rate by up to 70% from the most accurate prior work [7]. In fact, it showed that the algorithm approached the theoretical upper bound in reconstruction accuracy on the data set from Cho *et al.* [2].

Finally, the robustness of the Type 2 category puzzle solving strategies to image noise were analysed and evaluated by adding Gaussian noise to the MIT dataset. Five experiments were conducted and the performance values were averaged. It was shown that the method outperforms Gallagher [7] as the noise was increased.

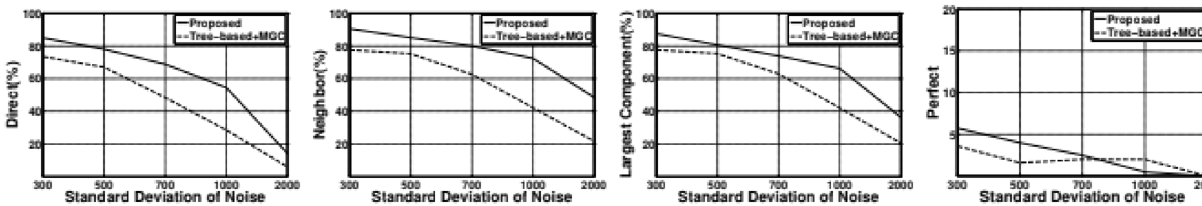
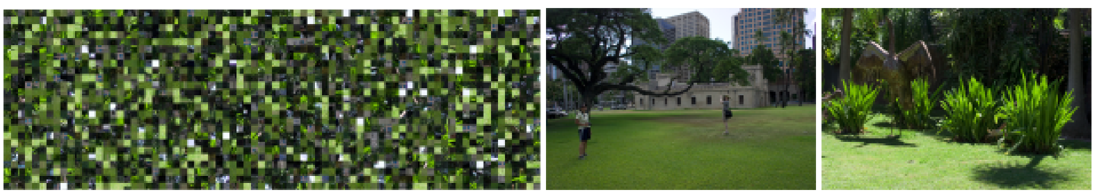
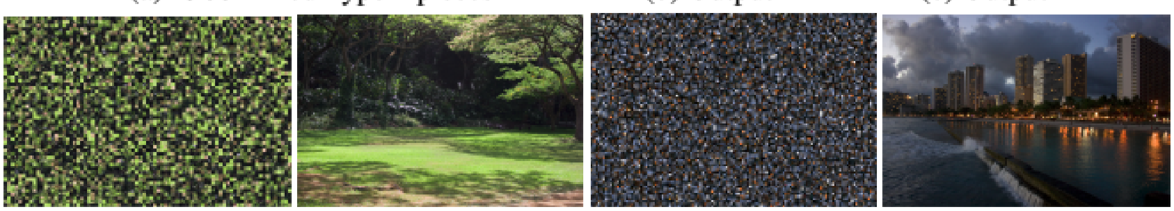


Fig. 5: Performance comparison in the presence of noise.



(a) 1900 mixed Type 2 pieces (b) Output 1 (c) Output 2



(d) 4108 Type 2 pieces (e) Output (f) 9801 Type 2 pieces (g) Output

Fig. 6: Reconstructions on mixed Type 2 puzzles and very large Type 2 puzzles.

c) Solving Multiple Square Jigsaw Puzzles with Missing Pieces[11]

This work was mainly inspired by Pomeranz's [4] work in which they introduced the first fully-automatic square jigsaw puzzle solver which is based on a greedy placer and on a novel prediction-based dissimilarity. However, this work didn't only improve upon the state-of-the-art results, it added an extra challenge in the form of missing pieces with unknown size and unknown orientation (Fig 7). Besides that, it also solved multiple jigsaw puzzles whose pieces were mixed together, where neither the size of the pieces were provided nor any a priori information regarding the missing pieces were given. This is shown in Fig 8.

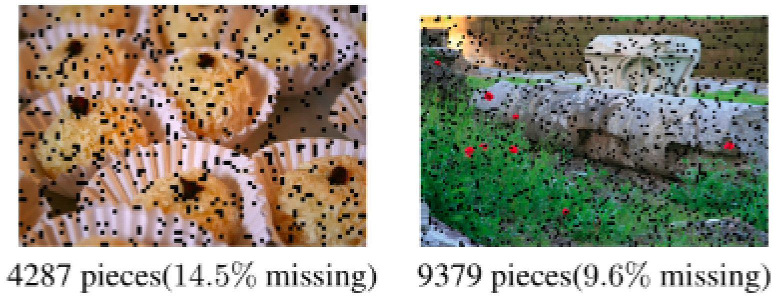


Fig 7. Accurate reconstructions of our solver, given jigsaw puzzles with missing pieces (black squares).

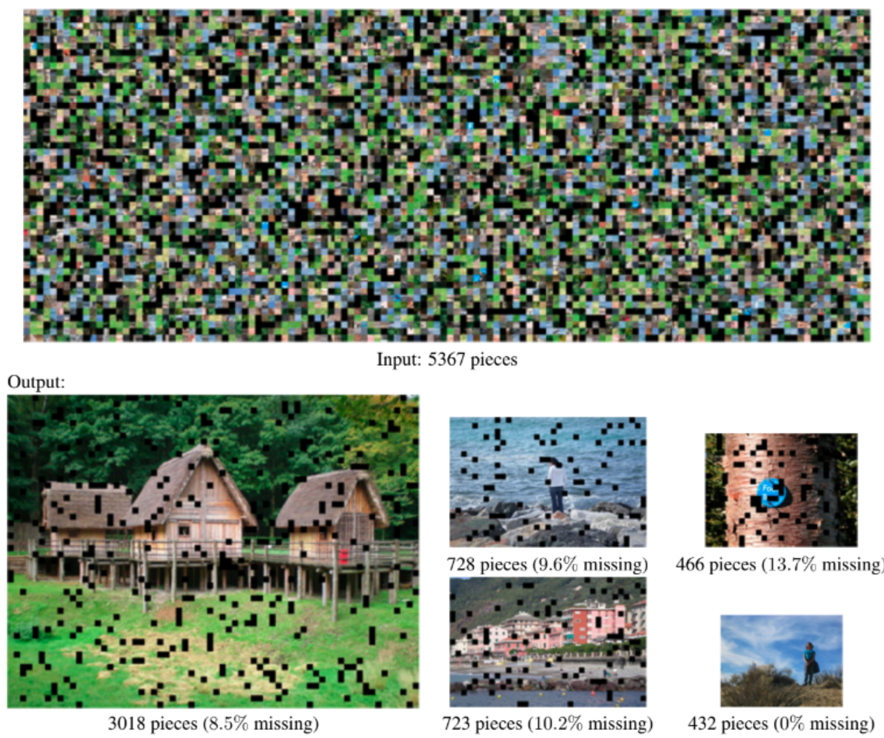


Fig 8. Solving multiple puzzles with mixed and missing pieces.

Like in [7, 4] the algorithm used was greedy. However they incorporated three main ideas which not only proved helpful in solving the standard puzzles but also supported the additional requirements. These were :

1) They proposed a more accurate, fast and reliable compatibility function which took advantage of both : the similarity between the pieces and the reliability of this similarity.

2) They took special care while choosing the first piece since the greedy solvers were extremely vulnerable to the initial placement.
3) Instead of choosing the best piece for a particular location, they selected the piece that minimises the likelihood of erring, irrespective of its location.

Due to these key decisions, the algorithm came out to be deterministic, i.e, it made random choices and were thus sensitive either to an initial random solution or to the selection of the first piece. Hence, it was only needed to run the algorithm once to solve the puzzle and outperformed the state-of-the-art methods both in efficiency and accuracy.

d) Solving Small-piece Jigsaw Puzzles by Growing Consensus[14]

In this paper they proposed a solver for challenging square-pieced jigsaw puzzles in which no prior information (piece orientation, anchor pieces and dimension of the puzzles) are known. They mainly focused their work on small jigsaw puzzles (mostly 14 X 14, 10 X 10 and 7 X 7) as for smaller puzzles the information contained in each piece gets reduced and the predefined compatibility measures become no longer reliable to configure the puzzles correctly.

A framework was chosen that estimated pairwise compatibility and simultaneously aligned these matched pieces by keeping track and making use of all the pairwise costs in the aligned sets. In virtue of this, the solver was able to reconstruct image puzzles with mixed pieces and no prior information provided. It also handled the case of missing fragments which has a major use in archaeology [12].

The biggest challenge in the proposed puzzle reconstruction strategy was that the consecutive pairwise matching must not include a single false pair since that may lead to an incorrect configuration. It was overcomed by incorporating two crucial steps in the matching-based assembly algorithm : an assembly strategy and a pairwise compatibility measure. They proposed a new objective for assembling puzzles, specifically maximising consensus (loop or grid) configurations.

They also improved the pairwise compatibility measure by adding derivative information along the boundary in the piece to the pairwise compatibility measure.

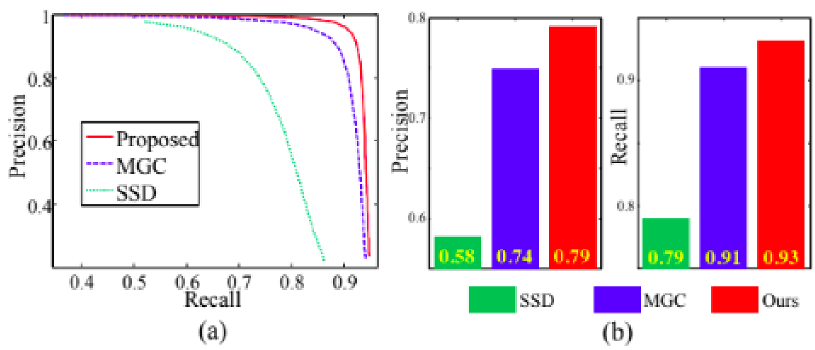


Fig 9. Pairwise compatibility measure performance.

It outperformed both Gallagher [7] and Son *et al.* [13] by a large margin in the very challenging MIT dataset (Fig 10). The proposed algorithm reduced assembly error by up to 75% compared with previous algorithms for the challenging unknown orientation puzzles from standard datasets where the size of pieces is small.

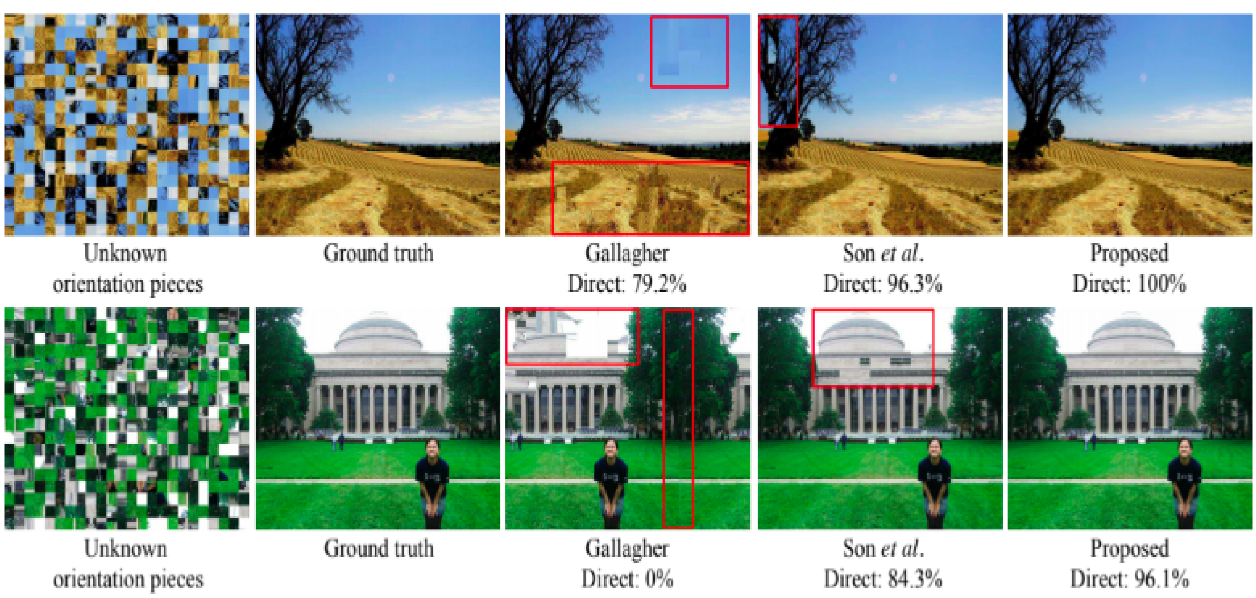


Fig 10. Qualitative reconstruction performance on unknown orientation piece puzzles.

Works on Irregular Shaped Jigsaw Puzzles

a) A Texture Based Matching Approach for Automated Assembly of Puzzles.[20]

It was one of the first works in the field of solving irregular shaped jigsaw puzzles and also including texture and color information into the general assembly problem. A jigsaw puzzle can be viewed as a reduced and restricted version of a general assembly problem. The first computerised solution to the puzzle was developed by Freeman [15] in which it successfully solved a 9-piece jigsaw puzzle. Other works [16, 17, 18] used feature based matching to solve the puzzle, these were relatively fast but failed to provide detailed matching of boundaries and overlapping regions. Prior to this paper, most works have ignored the texture and color information at the boundaries.

In the paper they designed a texture prediction algorithm that predicted the pixel values in a band outside the border of the pieces. Features obtained from the predicted texture outside a piece were correlated with original pictorial specifications of possible neighboring pairs. To achieve this they used image inpainting, a process of filling in missing data in a designated part of an image or a video from the surrounding area and texture synthesis methods (to create a new image with the same seed texture but of different shape to a sample region) for prediction.

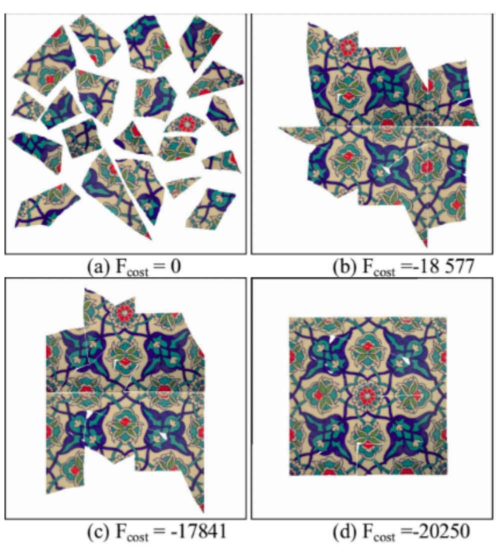


Fig 11. (a) Initial layout of the pieces (b,c,d) F_{cost} for 3 possible solutions.

Fig 12. Solution obtained from mixture of two different sets of tiles.

Further, they derived feature values in both the original fragment and the extended region. Then a combination of the feature and confidence values was used to generate an affinity measure of corresponding pieces. The matching pieces and achievement of the assembly was established by optimising the affinity measure.

They further used the FFT Shift Theory [19] to find the solution that will maximize the correlation between the predicted parts of a piece and other pieces which is indeed the reconstructed image. The work was however done only on 2D puzzles.

(b) A Graph-based Optimization algorithm for fragmented image reassembly.[28]

This paper was the first to incorporate both the geometry information (used to reassemble the fragments by matching their boundary curves [23–26]) and color information (used to predict the adjacency relationship of the fragments and guide the matching [20-22]) in the reassembly computation. This made the process more efficient, accurate, and reliable. The paper proposed a 3-step algorithm as follows :

1) **Pairwise matching** : The proposed integrated algorithm (i) extracted the border of each and every fragment and represented it as a curve contour, (ii) clustered such a curve contour into multiple segments based on both the color and the geometry and (iii) matched the segments to suggest the potential alignment between alignments. It resulted in a set of possible matching between identified pairs of fragments (both correct and incorrect) ones.

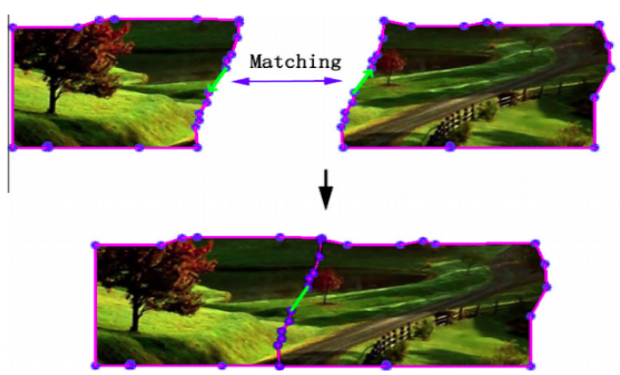


Fig 13. Fragment matching induced by cluster matching.

2) **Groupwise matching** : The pairwise matching produces many redundant matches which were intentionally generated in order to tolerate any erroneous adjacency identification due to noise and local minima. These false matches can be filtered out using a groupwise matching. This is proven to be NP-Hard. Instead of using the best-fit strategy, they formulated this composition as a graph-based searching problem which demonstrated better results.

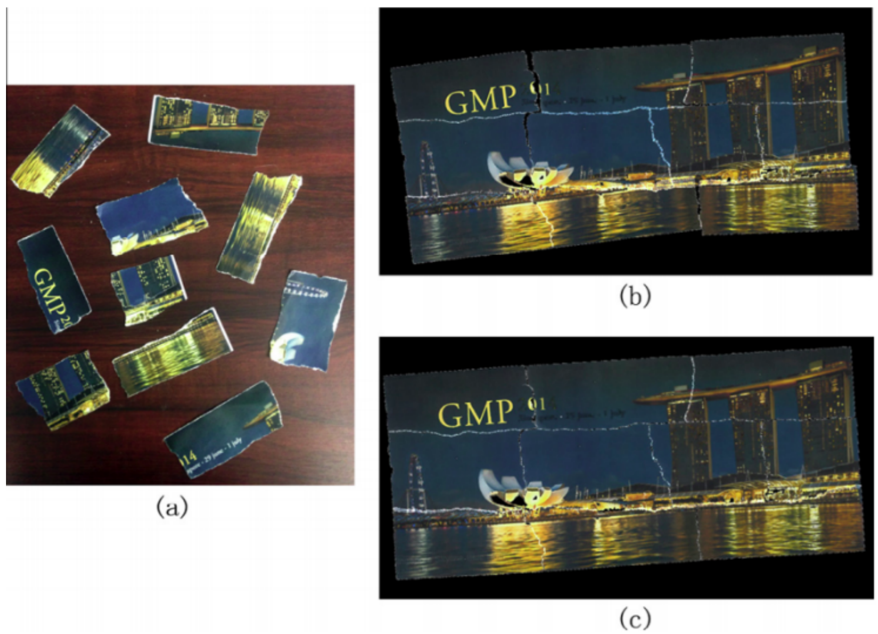


Fig 14. (a) GMP image fragments in 10 pieces, (b) initial reassembly results, and (c) final reassembly results.

3) **Composition refinement via graph optimization** : Groupwise matching can compose the fragments globally. However, the result is dependent on the composition sequence, and the alignment errors will be accumulated which may even cause a failure in the reassembly. They implemented a graph optimisation algorithm based on the framework of [27] to fix this. With the derived analytic derivatives information, the optimization converged to a local minimum efficiently. It was very effective and accurate on real world based image datasets.

Works in ML/DL Era

(a) Jigsaw Puzzle Solving Using Local Feature Co-occurrences in Deep Neural Networks[33]

Inspired by the setup proposed by Doersch *et al.* [29] this paper aimed at solving the jigsaw puzzle using a Convolutional Neural Network. Given an image of, they extracted same-sized 9 squared 2D tiles in a randomised grid pattern, visual features extracted using a CNN and lastly, combined and fed to a classifier to predict the correct relative position of one fragment with the other. Also for experimenting, two datasets were used : original ImageNet dataset proposed in [29] and a new MET dataset which contained 14000 images from the Metropolitan Museum of Art.

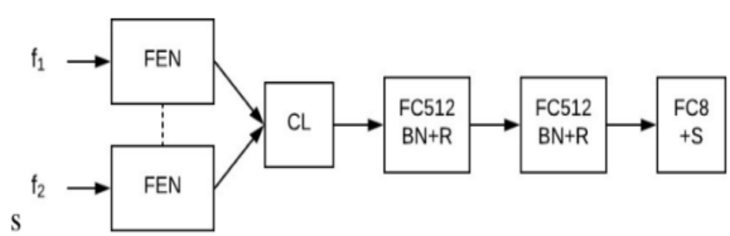
The proposed method can be divided into three steps :

1) **Feature Extraction Network** : CNN was used for feature extraction. The architecture was inspired by VGG and composed of a sequence of 3 × 3 convolutions followed by batch-normalizations [30], ReLU activations [31] and max-poolings. Each fragment was of size 96 X 96 with pixels varying between -1 and 1. The final layer consisted of a fully connected layer that took in account the spatial layout of the input fragments.

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

Fig 15. Architecture of the Feature Extraction Network

Fig 16. Full network architecture.



2) **Combination Layer** : In order to predict the relative position of fragments with respect to each other, features of both fragments were extracted using the same extraction network. These were then combined using a combination layer and further processed by a neural network. Instead of a linear combination of fragment features as proposed in [29], they used Kronecker product in the combination layer. This enabled them to explicitly model the co-occurrences between features of both fragments with a tradeoff of increase in parameters. The output of the full network consisted of a fully connected layer with k neurons followed by a Softmax activation, corresponding to the probabilities of the k possible relative locations.

3) **Puzzle Solving** : For each fragment, the probabilities of assignment were computed using the full network. This resulted in an 8×8 matrix where each row corresponded to a fragment and each column to a possible location. Solving the puzzle problem hence corresponded to an assignment problem in which we have to pick 8 values from the matrix (only one per row/column) such that their sum is maximized. A greedy algorithm was proposed which iteratively picks the maximum value and removes its corresponding row and column.

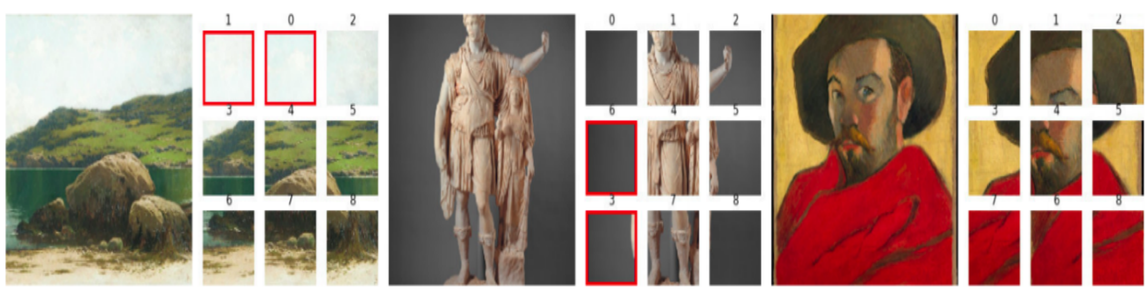


Fig 17. Examples of reconstructions using the greedy algorithm on images taken from the MET dataset. The red outlined fragments are misplaced.

The proposed robust deep learning method outperformed the state-of-the-art methods by 25%. Using the greedy algorithm, they were able to achieve perfect reconstruction 28.8% of the time and the proportion of correct fragments were measured to be 68.8%. The only limitation was the fixed and small size of the fragments of jigsaw puzzle in interest.

(b) JigsawNet: Shredded Image Reassembly using Convolutional Neural Network and Loop-based Compositions[40]

This paper proposed a novel algorithm to reassemble an arbitrarily shredded image to its original status. Most reassembly pipelines consist of two stages as discussed above : local matching stage and global assembly stage. Prior to this work, most of the methods to reassemble the fragments on a local stage used handcrafted features or heuristic methods to attach scores to each pair of fragments. Instead of this, they built a deep CNN to evaluate pairwise compatibility between fragment pairs. The performance was further improved by adding two technical components to the design : (i) the transfer of the CNN calculation attention on stitching regions, and (ii) an adaptive boosting training procedure for solving the data imbalance problem. For the global composition, the previous greedy strategy was modified to form two loop closure based graph searching algorithms.

The proposed work can be divided into three parts :

1) **Pairwise Alignment Candidate Extraction** : Since before matching nobody knows if the fragments in comparison can be matched or not, hence they computed matching between each pair of fragments. They adopted the pairwise matching computation strategy used in [28], which reduces the matching to a partial curve matching problem. The algorithm consisted of four steps : (i) Using a revised RDP algorithm [34] they approximated the noisy fragment boundary contour into a polygon whose each line segment has similar color; (ii) Matched each fragment pair by iteratively estimating all the possible segment-to-segment matches; (iii) Refined those good segment-to-segment matches using an ICP algorithm; and finally (iv) Evaluated the pairwise matching score by calculating the volume of well aligned pixels. Since they compared between each pair, the set of possible alignments produced contained both incorrect and correct pairs.

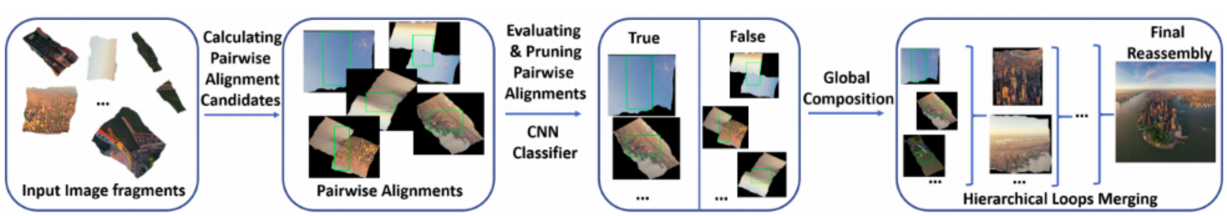


Fig 18. Image reassembly algorithm pipeline.

2) **Pairwise Compatibility Measurement** : Since pairwise alignment extraction produces a set of alignments containing majorly the incorrect pairs, they created a reliable compatibility evaluator to examine the candidates, filter out the incorrect ones and keep the most likely ones. Instead of the banal handcrafted and heuristic methods they took use of a classifier using a CNN network. The network was trained to identify whether the stitching of an image fragment pair under a specific pairwise alignment is correct or not and later boosted to improve performance and solve the data imbalance problem : *between-class imbalance* [35, 36, 37, 38] and *within-class imbalance* (solved using *AdaBoost* [39]). The network architecture was as shown below :

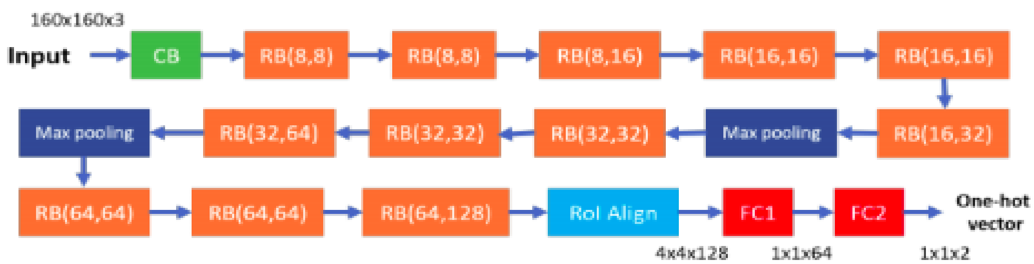


Fig 19. The convolutional neural network architecture. *CB* is the convolutional block. *RB(r, h)* is the residual block with depth of input *r* and depth of output *h*.

3) **Global Composition** : Even after designing a better pairwise compatibility measurement method, many misalignments due to local ambiguity were sometimes inevitable. These could only be handled from a global perspective. In this paper instead of using the popular greedy approaches they devised two new algorithms to enforce the global loop closure constraints and prune incorrect pairwise alignments, viz. Greedy Loop Closing (GLC) and Hierarchical Loop Merging (HLM). The former conducts the greedy selection in level of loops while the latter decides after hierarchical merging operations. HLM was also shown to be more robust than GLC in solving complicated puzzles.

Fragments | Tsam *et al.*[22] | Zhang *et al.*[28] | CNN + BF | CNN + GLC | CNN + HLM

Fig 20. Reassembly results on the BGU data. Fragments are 100 in no.

Limitations

We discussed various algorithms developed over the years to solve the Jigsaw Puzzle problem. However there exist some limitations either in the solvers or how we can use them to understand the synergy and variations in human and machine approach to solve a problem. Firstly, there exist some open jigsaw problems yet to be solved mainly *Multiple Solution Jigsaw Puzzles* and *Jigsaw Sudoku Puzzles*. A multiple solution jigsaw puzzle as the name suggests can have multiple solutions all following the same symmetry, color and texture. Such a case has not been handled by any of the works described above where we could come with certain kinds of multiple solutions of jigsaw puzzles. In case of the latter one, we have pieces of a solved sudoku puzzle as fragments. This can also lead to multiple solution as described in the figure below :

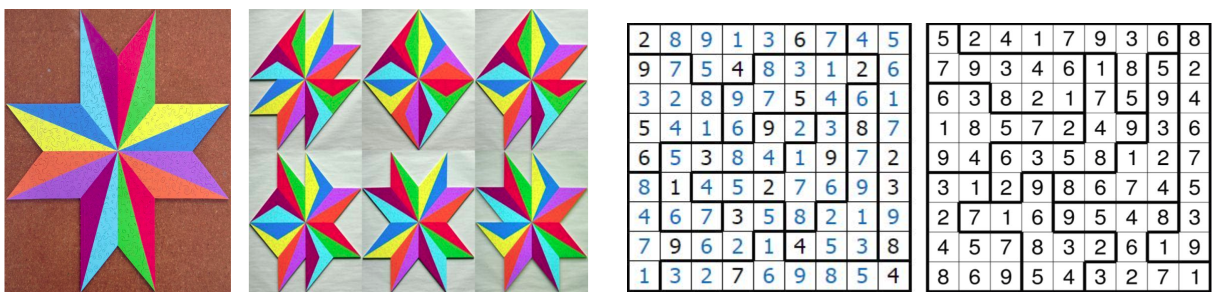


Fig 21. (a) Multiple solution jigsaw puzzle (b) A jigsaw sudoku puzzle with multiple solutions.

Secondly, abstract images like pink sky, blue desert or reflection of a scene in a very still lake,etc. which require deeper cognition and understanding remains to be modelled. Lastly, the current solvers only focus on solving the puzzle but the broader picture of aiding to learning remains not visible.

Goals of Study

The goal is to develop a model that would help compare and contrast the approaches of machine and human. This can help us study the similarities and dissimilarities on a deeper level and thus aid to problem solving. Further work is mainly divided into three parts : (i) Apart from Jigsaw puzzles we would like to also take in use of [Sliding puzzles](). We plan to build a web framework for the same where we plan to compare a human, a heuristic algorithm and a naive Convolutional neural network based solver (ii) We would also like to incorporate 3D jigsaw puzzles to study human cognition in depth. And lastly (iii) devising a better global composition method to increase efficiency of JigsawNet [40].

References

[1] E. D. Demaine and M. L. Demaine. Jigsaw puzzles, edge matching, and polyomino packing: Connections and complexity. *Graphs and Combinatorics*, 23, 2007.

[2] T. S. Cho, M. Butman, S. Avidan, and W. T. Freeman. The patch transforms its applications to image editing. In *IEEE CVPR*, 2008.

[3] Cho, T.S., Avidan, S., Freeman, W.T.: A probabilistic image jigsaw puzzle solver. In: *CVPR* (2010).

[4] Pomeranz, D., Shemesh, M., Ben-Shahar, O.: A fully automated greedy square jigsaw puzzle solver. In: *CVPR* (2011).

[5] Yang, X., Adluru, N., Latecki, L.J.: Particle filter with state permutations for solving image jigsaw puzzles. In: *CVPR* (2011).

[6] Fernanda A. Andal, Gabriel Taubin, S.G.: Solving image puzzles with a simple quadratic programming formulation. In: *Conference on Graphics, Patterns and Images* (2012).

[7] Gallagher, A.C.: Jigsaw puzzles with pieces of unknown orientation. In: *CVPR* (2012).

[8] Olmos, A., Kingdom, F.A.A.: A biologically inspired algorithm for the recovery of shading and reflectance images (2004).

[9] Sholomon, D., David, O., Netanyahu, N.: A genetic algorithm-based solver for very large jigsaw puzzles. In: *CVPR* (2013).

[10] K.Son, J. Hays, and D. Cooper. Solving square jigsaw puzzles with loop constraints. In *ECCV*, 2014.

[11] G. Paikin and A. Tal. Solving multiple square jigsaw puzzles with missing pieces. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

[12] A. Willis and D. B. Cooper. Computational reconstruction of ancient artifacts. *IEEE Signal Processing Magazine*, pages 65–83, 2008.

[13] K. Son, J. Hays, and D. Cooper. Solving square jigsaw puzzles with loop constraints. In *Computer Vision ECCV* 2014, Lecture Notes in Computer Science. Springer International Publishing, 2014.

[14] K. Son, J. Hays, D. B. Cooper et al., "Solving small-piece jigsaw puzzles by growing consensus," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition,* 2016, pp. 1193–1201.

[15] H. Freeman and L. Gardner, "A pictorial Jigsaw puzzles: The computer solution of a problem in pattern recognition", *IEEE Trans. Electron. Computers*, 13(1964) , pp. 118-127.

[16] D. Goldenberg, C. Malon, and M. Bern, "A global approach to automatic solution of jigsaw puzzles", In *Proc. of Conf. on Computational Geometry*, 2002, pp. 82-87.

[17] M.G. Chung, M.M. Fleck, and D.D. Forsyth, "Jigsaw Puzzle Solver Using Shape and Color", In *Proc. of ICSP*, 1998.

[18] A.D. Kosiba, P.M. Devaux, S. Balasubramanian, T.L. Gandhi, and R. Kasturi, "An automatic jigsaw puzzle solver", In *Proc. Int. Conf. Pattern Recognition*, 2001.

[19] G.Wolberg, S.Zokai, "Robust image registration using log-polar transform", In *Proc. of IEEE ICIP*, 2000.

[20] M. S. Sagiroglu and A. Ercil, "A texture based matching approach for automated assembly of puzzles," in *18th International Conference on Pattern Recognition (ICPR '06)*, vol. 3, 2006, pp. 1036–1041.

[21] F. Amigoni, S. Gazzani, S. Podico, A method for reassembling fragments in image reconstruction, in: *Proceedings of ICIP*, vols. 3,2, 2003, pp. III-581–4.

[22] E. Tsamoura, I. Pitas, Automatic color based reassembly of fragmented images and paintings, *IEEE Trans. Image Process.* 19 (2010) 680–690.

[23] W. Kong, B. Kimia, On solving 2d and 3d puzzles using curve matching, in: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, vol. 2, 2001, pp. II-583– II-590.

[24] H. da Gama Leitao, J. Stolfi, A multiscale method for the reassembly of two-dimensional fragmented objects, IEEE Trans. PAMI 24 (2002) 1239–1251.

[25] E. Justino, L.S. Oliveira, C. Freitas, Reconstructing shredded documents through feature matching, Forensic Sci. Int. 160 (2006) 140–147.

[26] L. Zhu, Z. Zhou, D. Hu, Globally consistent reconstruction of ripped-up documents, IEEE Trans. Pattern Anal. Mach. Intell. 30 (2008) 1–13.

[27] R. Kummerle, G. Grisetti, H. Strasdat, K. Konolige, W. Burgard, g2o: a general framework for graph optimization, in: *IEEE ICRA,* 2011, pp. 3607–3613.

[28] K. Zhang and X. Li, "A graph-based optimization algorithm for fragmented image reassembly," *Graphical Models*, vol. 76, no. 5, pp. 484–495, 2014.

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

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

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

[32] D. Brijlall, ― Exploring The Stages of Polya's Problem Solving Model during Collaborative Learning: A Case of Fractions, *Int. J. Educ. Sci.*, vol. 11, no. 3, pp. 291– 299, 2015

[33] M.-M. Paumard, D. Picard, and H. Tabia, "Jigsaw puzzle solving using local feature co-occurrences in deep neural networks," *arXiv preprint arXiv:1807.03155*, 2018

[34] D. H. Douglas and T. K. Peucker, "Algorithms for the reduction of the number of points required to represent a digitized line or its caricature," *Cartographica: The International Journal for Geographic Information and Geovisualization*, vol. 10, no. 2, pp. 112–122, 1973.

[35] H. He and E. A. Garcia, "Learning from imbalanced data," *IEEE Transactions on knowledge and data engineering*, vol. 21, no. 9, pp. 1263–1284, 2009.

[36] M. Buda, A. Maki, and M. A. Mazurowski, "A systematic study of the class imbalance problem in convolutional neural networks," *arXiv preprint arXiv:1710.05381*, 2017.

[37] C. Elkan, "The foundations of cost-sensitive learning," in *International joint conference on artificial intelligence*, vol. 17, no. 1. Lawrence Erlbaum Associates Ltd, 2001, pp. 973–978.

[38] M. D. Richard and R. P. Lippmann, "Neural network classifiers estimate bayesian a posteriori probabilities," *Neural computation*, vol. 3, no. 4, pp. 461–483, 1991.

[39] Y. Freund, R. Schapire, and N. Abe, "A short introduction to boosting," *Journal-Japanese Society For Artificial Intelligence*, vol. 14, no. 771-780, p. 1612, 1999.

[40] Le, Canyu & Li, Xin. (2019). JigsawNet: Shredded Image Reassembly Using Convolutional Neural Network and Loop-Based Composition. *IEEE Transactions on Image Processing*. PP. 1-1. 10.1109/TIP.2019.2903298.