

DNN-Buddies: A Deep Neural Network-Based Estimation Metric for the Jigsaw Puzzle Problem

Dror Sholomon¹, Omid E. David¹, and Nathan S. Netanyahu^{1,2}

¹ Department of Computer Science, Bar-Ilan University, Ramat-Gan 52900, Israel
dror.sholomon@gmail.com, mail@omidavid.com, nathan@cs.biu.ac.il

² Center for Automation Research, University of Maryland, College Park, MD 20742
nathan@cfar.umd.edu

Abstract. This paper introduces the first deep neural network-based estimation metric for the jigsaw puzzle problem. Given two puzzle piece edges, the neural network predicts whether or not they should be adjacent in the correct assembly of the puzzle, using nothing but the pixels of each piece. The proposed metric exhibits an extremely high precision even though no manual feature extraction is performed. When incorporated into an existing puzzle solver, the solution's accuracy increases significantly, achieving thereby a new state-of-the-art standard.



Fig. 1: Jigsaw puzzle before and after reassembly using our DNN-Buddies scheme in an enhanced solver.

1 Introduction

Jigsaw puzzles are a popular form of entertainment, available in different variation of difficulty to challenge children, adults and even professional players. Given $n \times m$

different non-overlapping tiles of an image, the objective is to reconstruct the original image, taking advantage of both the shape and chromatic information of each piece. Despite the popularity and vast distribution of jigsaw puzzles, their assembly is not trivial computationally, as this problem was proven to be NP-hard [1] [8]. Nevertheless, a computational jigsaw solver may have applications in many real-world applications, such as biology [16], chemistry [25], literature [18], speech descrambling [27], archeology [2] [15], image editing [5], and the recovery of shredded documents or photographs [3, 7, 14, 17]. Regardless, as noted in [11], research of the topic may be justified solely due to its intriguing nature.

Recent years have witnessed a vast improvement in the research and development of automatic jigsaw puzzle solvers, manifested in both puzzle size, solution accuracy, and amount of manual human intervention required. In its most basic form, every puzzle solver requires some function to evaluate the compatibility of adjacent pieces and a strategy for placing the pieces as accurately as possible. Most strategies are greedy and rely heavily on some “trick” to estimate whether two pieces are truly adjacent (e.g. two pieces that are each the most compatible piece from all pieces to one another, four pieces that form a loop where each pair’s compatibility is above a threshold, etc). Such heuristics were dubbed an “estimation metric” in [20], as they allow estimating the adjacency correctness of two pieces without knowing the correct solution. The majority of recent works focused on devising elaborate, hand-crafted compatibility functions and high-precision estimation metrics.

Despite the proven effectiveness of neural networks in the field of computer vision, no attempt has been made to automatically devise a high-precision estimation metric for the jigsaw puzzle problem. This might be due to the highly imbalanced nature of the puzzle problem, as in each $n \times m$ puzzle, there are $\mathcal{O}(n \times m)$ matching piece-pairs and $\mathcal{O}(n^2 \times m^2)$ possible mismatching ones. In this paper we propose a novel estimation metric relying on neural networks. The proposed metric achieves extremely high precision despite the lack of any manually extracted features.

The proposed metric proves to be highly effective in real-world scenarios. We incorporated the metric in our GA-based solver, using no hand-crafted sophisticated compatibility measure and experimented with the currently known challenging benchmarks of the hardest variant of the jigsaw puzzle problem: non-overlapping, (28×28) square pieces (i.e. only chromatic information is available to the solver) where both piece orientation and puzzle dimensions are unknown. The enhanced solver proposed sets a new state-of-the-art in terms of the accuracy of the solutions obtained and the number of perfectly reconstructed puzzles.

2 Previous Work

Jigsaw puzzles were first introduced around 1760 by John Spilsbury, a Londonian engraver and mapmaker. Nevertheless, the first attempt by the scientific community to computationally solve the problem is attributed to Freeman and Garder [9] who in 1964 presented a solver which could handle up to nine-piece problems. Ever since then, the research focus regarding the problem has shifted from shape-based to merely color-based solvers of square-tile puzzles. In 2010 Cho *et al.* [4] presented

a probabilistic puzzle solver that could handle up to 432 pieces, given some a priori knowledge of the puzzle. Their results were improved a year later by Yang *et al.* [26] who presented a particle filter-based solver. Furthermore, Pomeranz *et al.* [20] introduced that year, for the first time, a fully automated square jigsaw puzzle solver that could handle puzzles of up to 3,000 pieces. Gallagher [10] has further advanced this by considering a more general variant of the problem, where neither piece orientation nor puzzle dimensions are known. Son *et al.* [24] improved the accuracy of the latter variant using so-called “loop-constraints”. Palkin and Tal [19] further improved the accuracy and handled puzzles with missing pieces. Sholomon *et al.* [21] presented a genetic algorithm (GA)-based solver for puzzles of known orientation which was later generalized to other variants [22, 23].

2.1 Compatibility Measures and Estimation Metrics

As stated earlier, most works focus on the compatibility measure and an estimation metric. A compatibility measure is a function that given two puzzle piece edges (e.g. the right edge of piece 7 versus the upper edge of piece 12) predicts the likelihood that these two edges are indeed placed as neighbors in the correct solution. This measure applies to each possible pair of piece edges. The estimation metric, on the other hand, predict whether two piece edges are adjacent but may not apply to many possible pairs. Following is a more detailed review of the efforts made so far in the field.

Cho *et al.* [4] surveyed four compatibility measures among which they found dissimilarity the most accurate. Dissimilarity is the sum (over all neighboring pixels) of squared color differences (over all color bands). Assuming pieces x_i, x_j are represented in some three-dimensional color space (like RGB or YUV) by a $K \times K \times 3$ matrix, where K is the height/width of a piece (in pixels), their dissimilarity, where x_j is to the right of x_i , for example, is

$$D(x_i, x_j, r) = \sqrt{\sum_{k=1}^K \sum_{cb=1}^3 (x_i(k, K, cb) - x_j(k, 1, cb))^2}, \quad (1)$$

where cb denotes the color band.

Pomeranz *et al.* [20] also used the dissimilarity measure but found empirically that using the $(L_p)^q$ norm works better than the usual L_2 norm. Moreover, they presented the high-precision *best-buddy* metric. Pieces x_i and x_j are said to best-buddies if

$$\begin{aligned} \forall x_k \in Pieces, C(x_i, x_j, R_1) \geq C(x_i, x_k, R_1) \\ \text{and} \\ \forall x_p \in Pieces, C(x_j, x_i, R_2) \geq C(x_j, x_p, R_2) \end{aligned} \quad (2)$$

where *Pieces* is the set of all given image pieces and R_1 and R_2 are “complementary” spatial relations (e.g. if $R_1 = \text{right}$, then $R_2 = \text{left}$ and vice versa).

Gallagher [10] proposed yet another compatibility measure, called the *Mahalanobis gradient compatibility* (MGC) as a preferable compatibility measure to those

used by Pomeranz *et al.* [20]. The MGC penalizes changes in intensity gradients, rather than changes in intensity, and learns the covariance of the color channels, using the Mahalanobis distance. Also, Gallagher suggested using *dissimilarity ratios*. Absolute distances between potential piece edge matches are sometimes not indicative (for example in smooth surfaces like sea and sky), so considering the absolute score, divided by the second-best score available seems more indicative.

Son *et al.* [24] suggested “loop-constraints”, four or more puzzle piece edges where the compatibility ratio between each pair is in the top ten among all possible pairs of piece edges in the given puzzle. Palkin and Tal [19] proposed a greedy solver based on an L_1 -norm asymmetric dissimilarity and the best-buddies estimation metric.

3 DNN-Buddies

3.1 Motivation

We propose a novel estimation metric called “DNN-Buddies”. Our goal is to obtain a classifier which predicts the adjacency likelihood of two puzzle piece edges in the correct puzzle configuration.

Note that despite the exponential nature of the problem (as there are $\mathcal{O}((nm)!)$ possible arrangements of the pieces, taking into account rotations), the problem can be solved theoretically by assigning correctly, in a consecutive manner, $n \times m - 1$ piece-edge pairs. (This is reminiscent of finding a minimal spanning tree, as noted by [10].) Hence, the classifier’s precision is of far greater importance than its recall. A classifier with perfect precision and a recall of

$$\frac{n \times m - 1}{\text{all possible matches}} = \frac{n \times m - 1}{4 \times (n \times (m - 1) + (n - 1) \times m)} < \frac{1}{8} \quad (3)$$

might achieve a perfect solution by itself.

3.2 Challenges

A straight-forward solution might have been to train a neural network against matching-pairs vs. non-matching ones. However, the issue of a jigsaw puzzle piece matching is of an imbalanced nature. In each $n \times m$ puzzle, there are $\mathcal{O}(n \times m)$ matching pairs of piece edges and $\mathcal{O}(n^2 \times m^2)$ possible nonmatching ones. A thorough review on the challenges and tactics to avoid them can be found in [13].

The trivial approach of random or uninformed undersampling, i.e. randomly choosing the required number of nonmatching pairs leads to a low-precision and high-recall metric, the very opposite of the goal set beforehand. We believe that the reason for this shortcoming is that there exist many “easy-to-spot” mismatches but only a handful of “hard-to-spot” ones. Thus, we resort to informed undersampling, choosing a subset of “good” mismatching pairs according to some criterion. Nevertheless, we avoid using any manual feature selection or other sophisticated image-related means.

In the jigsaw puzzle domain, similarly to many other problem domains, the solver does not actually try to reassemble the original image (as this problem is

not mathematically defined), but rather tries solving a “proxy problem” which is to achieve an image whose global overall score between abutting-edges is minimal. Thus, we choose using the compatibility measure as the undersampling criterion.

3.3 Neural Network Training

For training and cross-validation, we use the 2,755 images of size 360×480 pixels from the IAPR TC-12 Benchmark [12]. Each image is first converted to YUV space followed by the normalization of each channel separately (via z -score normalization). Next, each (puzzle) image is divided to 12×17 tiles, where each tile is of size 28×28 pixels (as in all previous works); finally, we create a balanced set of positive and negative samples of puzzle-piece pairs, using informed undersampling as will be described below. In the end, we obtain a balanced set of 970,224 pairs overall.

To balance our dataset, we use the most basic compatibility score which is the dissimilarity between two piece-edges in the YUV color-space, as described in Eq. 1, as an undersampling criterion. For each puzzle piece edge $x_{i,j}$ ($i = 1..n \times m, j = 1..4$), we find its most compatible piece edge $x_{k1,l1}$ and its second most compatible piece edge $x_{k2,l2}$. If the pair of edges $x_{i,j} - x_{k1,l1}$ is indeed adjacent in the original image, we add this pair to the pool of positively-labeled samples and toss the pair $x_{i,j} - x_{k2,l2}$ to the pool of negatively-labeled samples. Otherwise, $x_{i,j} - x_{k1,l1}$ is added to the negatively-labeled samples and the other pair is discarded. The latter is done to avoid training the network on adjacent pieces which happen to be vastly different due to a significant change of the image scenery in the corresponding region. In other words, we restrict our interest to highly compatible piece edges that are indeed adjacent. Since this method leads to more negative samples than positive ones, we eventually randomly throw some negative samples to balance out the set.

From each image pair we extract the two columns near the edge, i.e. the column of abutting pixels in each edge and the one next to it. This results in an input of size $(28 \times 4 \times 3 =) 336$ pixels. We use a feed-forward neural network (FFNN) of five fully connected layers of size 336, 100, 100, 100, and 2. The output is a softmax layer containing two neurons. We expect (0, 1) for matching pairs and (1, 0) otherwise. The activation used in all layers is the rectified linear unit (ReLU) function, i.e. $f(x) = \max(0, x)$. Figure 2 depicts the network’s structure.

We trained the network in a supervised manner using *Stochastic Gradient Descent* that minimizes the negative log likelihood of the error for 100 iterations. The resulting network reaches 95.04% accuracy on the training set and 94.62% on a held-out test set.

All dataset preparation and network training was performed using Torch7 [6].

4 Experimental Results

For each piece edge $x_{i,j}$ ($i = 1..n \times m, j = 1..4$), if its most compatible piece edge $x_{k,l}$ is classified positively using the DNN-Buddies network, we define $x_{k,l}$ to be $x_{i,j}$ ’s *DNN-buddy* piece edge. Note that each piece edge can have only a single DNN-buddy;

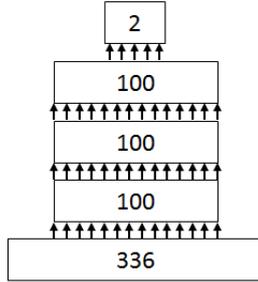


Fig. 2: Architecture of our DNN-Buddies scheme.

also, some pieces might not have a DNN-buddy at all (if the most compatible piece is not classified as one by the DNN-Buddies network).

First, we evaluate the precision of the proposed metric, i.e. how many DNN-buddies are indeed adjacent in the original image. Using the well known dataset presented by Cho *et al.* [4] of 20 432-piece puzzles, we obtained a precision of 94.83%.

Next, we incorporated the estimation metric (due to the proposed DNN-Buddies scheme) into the GA-based solver proposed by us previously [23]. Unfortunately, due to lack of space, no self-contained review of genetic algorithms and the proposed method can be included in this paper. Nevertheless, the modification required with respect to the existing GA framework is rather simple; if a DNN-buddy pair appears in one of the parents, assign this pair in the child. Figure 3 describes the modified crossover operator in the GA framework according to the above (see Step 2, which includes the new DNN-buddy phase).

Until $(n - 1)$ relative relations are assigned **do**

1. Try assigning all *common* relative relations in the parents.
 2. Try assigning all *DNN-buddy* relative relations in the parents.
 3. Try assigning all *best-buddy* relative relations in the parents.
 4. Try assigning all existing *most-compatible* relative relations.
 5. Try assigning *random* relative relations.
-

Fig. 3: Crossover overview

We ran the augmented solver on the 432-piece puzzle set and on the two additional datasets proposed by Pomeranz *et al.* [20] of 540- and 805- piece puzzles. We evaluated our results according to the *neighbor comparison* which measures the fraction of correct neighbors and the number of puzzles perfectly reconstructed for each set.

Table 1 presents the accuracy results of the same solver with and without the DNN-Buddies metric. For each dataset we achieve a considerable improvement in the overall accuracy of the solution, as well as the number of perfectly reconstructed puzzles. Moreover, our enhanced deep neural network-based scheme appears to outperform the current state-of-the-art results, as it yields accuracy levels of 95.65%, 96.37% and 95.86%, which surpass, respectively, the best results known of 95.4% [19], 94.08% and 94.12% [23].

# of Pieces	GA		Our (GA + DNN-Buddies)	
	Neighbor	Perfect	Neighbor	perfect
432	94.88%	11	95.65%	12
540	94.08%	8	96.37%	11
805	94.12%	6	95.86%	8

Table 1: Comparison of our accuracy results with and without the new DNN-Buddies estimation metric.

5 Conclusions

In this paper we presented the first neural network-based estimation metric for the jigsaw puzzle problem. Unlike previous methods, no manual feature crafting was employed. The novel method exhibits high precision and when combined with a real-world puzzle solver, it significantly improves the solution’s accuracy to set a new state-of-the art standard.

References

1. Altman, T.: Solving the jigsaw puzzle problem in linear time. *Applied Artificial Intelligence an International Journal* 3(4), 453–462 (1989)
2. Brown, B., Toler-Franklin, C., Nehab, D., Burns, M., Dobkin, D., Vlachopoulos, A., Doumas, C., Rusinkiewicz, S., Weyrich, T.: A system for high-volume acquisition and matching of fresco fragments: Reassembling Theran wall paintings. *ACM Transactions on Graphics* 27(3), 84 (2008)
3. Cao, S., Liu, H., Yan, S.: Automated assembly of shredded pieces from multiple photos. In: *IEEE International Conference on Multimedia and Expo*. pp. 358–363 (2010)
4. Cho, T., Avidan, S., Freeman, W.: A probabilistic image jigsaw puzzle solver. In: *IEEE Conference on Computer Vision and Pattern Recognition*. pp. 183–190 (2010)
5. Cho, T., Butman, M., Avidan, S., Freeman, W.: The patch transform and its applications to image editing. In: *IEEE Conference on Computer Vision and Pattern Recognition*. pp. 1–8 (2008)
6. Collobert, R., Kavukcuoglu, K., Farabet, C.: Torch7: A matlab-like environment for machine learning. In: *BigLearn, NIPS Workshop*. No. EPFL-CONF-192376 (2011)
7. Deever, A., Gallagher, A.: Semi-automatic assembly of real cross-cut shredded documents. In: *ICIP*. pp. 233–236 (2012)

8. Demaine, E., Demaine, M.: Jigsaw puzzles, edge matching, and polyomino packing: Connections and complexity. *Graphs and Combinatorics* 23, 195–208 (2007)
9. Freeman, H., Garder, L.: Apictorial jigsaw puzzles: The computer solution of a problem in pattern recognition. *IEEE Transactions on Electronic Computers EC-13*(2), 118–127 (1964)
10. Gallagher, A.: Jigsaw puzzles with pieces of unknown orientation. In: *IEEE Conference on Computer Vision and Pattern Recognition*. pp. 382–389 (2012)
11. Goldberg, D., Malon, C., Bern, M.: A global approach to automatic solution of jigsaw puzzles. *Computational Geometry: Theory and Applications* 28(2-3), 165–174 (2004)
12. Grubinger, M., Clough, P., Müller, H., Deselaers, T.: The IAPR TC-12 benchmark: A new evaluation resource for visual information systems. In: *International Workshop OntoImage*. vol. 5, p. 10 (2006)
13. He, H., Garcia, E.A.: Learning from imbalanced data. *Knowledge and Data Engineering, IEEE Transactions on* 21(9), 1263–1284 (2009)
14. Justino, E., Oliveira, L., Freitas, C.: Reconstructing shredded documents through feature matching. *Forensic science international* 160(2), 140–147 (2006)
15. Koller, D., Levoy, M.: Computer-aided reconstruction and new matches in the forma urbis romae. *Bullettino Della Commissione Archeologica Comunale di Roma* pp. 103–125 (2006)
16. Marande, W., Burger, G.: Mitochondrial DNA as a genomic jigsaw puzzle. *Science* 318(5849), 415–415 (2007)
17. Marques, M., Freitas, C.: Reconstructing strip-shredded documents using color as feature matching. In: *ACM symposium on Applied Computing*. pp. 893–894 (2009)
18. Morton, A.Q., Levison, M.: The computer in literary studies. In: *IFIP Congress*. pp. 1072–1081 (1968)
19. Paikin, G., Tal, A.: Solving multiple square jigsaw puzzles with missing pieces. In: *Computer Vision and Pattern Recognition, 2015 IEEE Conference on*. pp. 4832–4839. IEEE (2015)
20. Pomeranz, D., Shemesh, M., Ben-Shahar, O.: A fully automated greedy square jigsaw puzzle solver. In: *IEEE Conference on Computer Vision and Pattern Recognition*. pp. 9–16 (2011)
21. Sholomon, D., David, O.E., Netanyahu, N.S.: A genetic algorithm-based solver for very large jigsaw puzzles. In: *IEEE Conference on Computer Vision and Pattern Recognition*. pp. 1767–1774 (2013)
22. Sholomon, D., David, O.E., Netanyahu, N.S.: A generalized genetic algorithm-based solver for very large jigsaw puzzles of complex types. In: *AAAI Conference on Artificial Intelligence*. pp. 2839–2845 (2014)
23. Sholomon, D., David, O.E., Netanyahu, N.S.: Genetic algorithm-based solver for very large multiple jigsaw puzzles of unknown dimensions and piece orientation. In: *ACM Conference on Genetic and Evolutionary Computation*. pp. 1191–1198 (2014)
24. Son, K., Hays, J., Cooper, D.B.: Solving square jigsaw puzzles with loop constraints. In: *European Conference on Computer Vision 2014*, pp. 32–46. Springer (2014)
25. Wang, C.S.E.: Determining molecular conformation from distance or density data. Ph.D. Thesis, Massachusetts Institute of Technology (2000)
26. Yang, X., Adluru, N., Latecki, L.J.: Particle filter with state permutations for solving image jigsaw puzzles. In: *IEEE Conference on Computer Vision and Pattern Recognition*. pp. 2873–2880 (2011)
27. Zhao, Y., Su, M., Chou, Z., Lee, J.: A puzzle solver and its application in speech descrambling. In: *WSEAS International Conference Computer Engineering and Applications*. pp. 171–176 (2007)