Vignette - Solving Jigsaw Puzzles

Bhavi, Mayank Mor, Priyansh Desai, Rushiraj Gadhvi, Soham Petkar

April 1, 2024

1 Objective

Puzzle Reconstruction serves more than just a recreational activity. It poses a complex problem-solving challenge with implications spanning various disciplines, from archaeology and art to computational biology and beyond. Our project aims to develop a novel Deep Learning approach for the reconstruction of jigsaw puzzles by exploiting the capabilities of Vision Graph Neural Networks (ViG). ViG stands out for their ability to model relational data, making them ideal for the task of jigsaw puzzle reconstruction. By representing each piece of a puzzle as a node in a graph, ViG can be trained to model complex inter-node relationships. Through our work, we aim to use various Transformer, Encoder, and ViG models to develop a novel methodology to contribute to the research of learning visual embeddings and reconstructing Jigsaw puzzles.

2 Dataset

No universally accepted standard dataset has been established for the jigsaw problem, leading each research paper to employ its own dataset for evaluation purposes.

Through our approach, we discovered that low-quality images are not suitable for benchmark models such as the U2Net background removal model and might hinder the feature learning process. It was evident that achieving satisfactory results with U2Net necessitates a minimum image dimension of 120x120. We have chosen to utilize the Animal-10N dataset due to its collection of medium-resolution images and diverse array of animal classes. This dataset contains approximately 2,000 to 5,000 images per class, offering abundant data for experimentation. Furthermore, we have decided to take the recent work of [2] on Jigsaw-ViT as our baseline for comparison. Example of Animals-10 Dataset is shown in Figure 1. The current sample is after image is resized at 120x120.

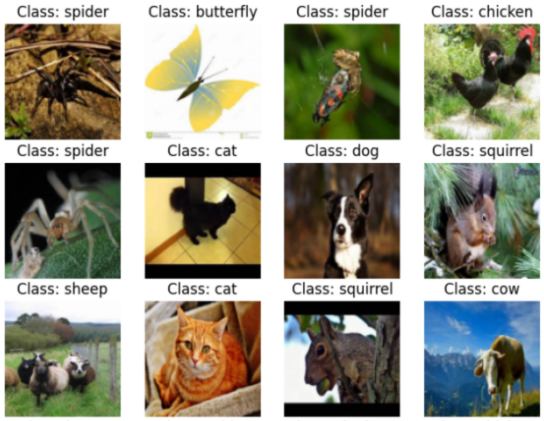

Figure 1: Animals-10 Data Samples

2.1 Data Preparation

As shown in Figure 2, our image is first subjected to segmentation by an advanced neural network, such as U2Net, which isolates the subjects from their background, effectively creating a saliency map. Next, the segmented image is divided into a 3x3 tile grid, partitioning the subjects into nine distinct segments. Our rationale behind the removal of the background is so that the model learns only about the main subject/object in the image, We believe that this coupled with the overall shuffled image(with the background

present) graph nodes would be beneficial input features for understanding the correct feature orientation in the image. The final step displayed here involves shuffling these tiles, resulting in a disordered array of image segments. The selection of shuffling the pieces would be done using the the maximal Hamming distance [16], and we assign an index to each entry.

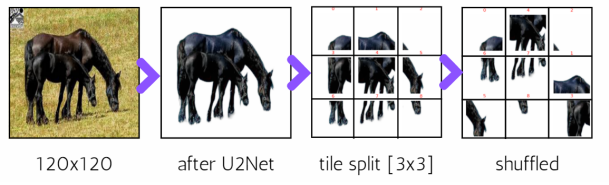

Figure 2: Data Preparation

2.2 Data Split

For this project, the dataset is divided into 70% training , 15% validation and 15% test sets. The training set is used to train the model, the validation set is used to tune hyperparameters and monitor the model's performance during training, and the test set is used to evaluate the final performance of the trained model.

2.3 Labels and Data

The input data consists of the 120x120 RGB image. The labels are integer values corresponding to the class of each image, ranging from 0 to 8, representing the 9 different categories mentioned above. An array of 9 positions is passed containing information regarding which passed tile belongs where.

3 Introduction

Puzzle reconstruction presents a fascinating yet challenging area within the domain of computer vision. Amongst its many uses, jigsaw reconstruction plays a crucial role in the accurate reassembly of fragmented historical artworks [1, 18] and artifacts [10], facilitating the study of our cultural heritage as well as the reconstruction of shredded documents [15] and photographs [13]. Additionally, its applications extend into the biological sphere, such as in genome reconstruction [14].

Though solving jigsaw puzzles has been proven to be an NP-complete problem [4], several algorithms have been proposed to tackle different types of puzzles through the years. Gallagher [7] classifies puzzles into three categories - Type 1 puzzles are characterized by the translation of pieces without rotation, Type 2 puzzles represent the category where pieces undergo both translation and rotation whereas Type 3 puzzles involve the rotation of pieces without translation. However, we limit ourselves to only Type 1 puzzles for now.

The first such algorithm was introduced by Freeman and Garder [6], developed to solve uniformly colored 9-piece puzzles, the model focuses exclusively on the geometric shape of the individual pieces. This was achieved through methods aimed at characterizing and classifying the puzzle pieces, selection and rearrangement of the pieces based on compatibility, determination of the likelihood of fit, and finally by careful evaluation of the puzzle assembly process. Moreover, they build upon their research in [3] where instead of treating this as a constraint satisfaction problem, they developed a probabilistic solver based on a graphical model to achieve puzzle reconstruction. This greedy algorithm acted as an initial benchmark for solving square-piece jigsaw puzzles.

Expanding on Freeman and Garder, subsequent approaches were proposed to solve the Type 1 puzzles. Many also included variants like handling puzzles with eroded boundaries [17, 19]. Yu *et al.* [21] propose a novel Linear Program based approach. Their method, centered around an LP assembly strategy, outperforms conventional greedy algorithms by using simultaneous pairwise matches to globally position pieces. This approach decreases the likelihood of getting stuck in local minima and improves robustness to mismatches in pairwise matches. Bridger *et al.* [17] proposed a novel GAN-based approach to solve puzzles having eroded boundaries. They first inpaint eroded boundaries between puzzle pieces and then use the quality of the inpainted area to classify pieces as neighbors. Their uniqueness lies in the fu-

sion of the same GAN discriminator for both inpainting and classification. For inpainting, it generates the missing pixels in the boundaries using the known pixels, whereas for neighbor classification it learns to discriminate the inpainted gaps of neighboring pieces from those of non-neighbours. Thus, after training, it can compute the pairwise dissimilarity between two pieces. Lastly, they use the greedy placement method based on the pairwise-dissimilarity scores to solve the puzzle.

Furthermore, recent developments in Deep Learning provide the opportunity for finding better reorganizations with more robustness than traditional algorithms. Paumard *et al.* [19] proposed a CNN-based method to predict the position of fragments to detect neighboring pieces followed by shortest path optimization using graphs for the best reassembly of the images. They were able to accurately solve eroded puzzles having missing pieces as well as pieces not belonging to the original jigsaw (Outsider Fragments). Der *et al.* [5] used pre-trained ResNet-50 [9] and VGG-16 [20] architectures for feature extraction followed by a pointer network to correctly reassemble the original image. Through the combination of a feature extraction pipeline and a pointer network for combinatorial reasoning, their neural network was able to reconstruct jigsaw puzzles of any size configuration with good accuracy. Finally, Ru *et al.* [11] proposed a GAN-based auxiliary learning method to solve puzzles with no knowledge of initial images. Their proposed method solves jigsaw puzzles by using the semantic information as well as boundary information of the puzzle pieces simultaneously. Thus, their method outperformed previous approaches both quantitatively and qualitatively.

Therefore, we build upon the wide body of existing literature to develop a new approach to solving jigsaw puzzles. But the work of [2] and [11] has been the most influential in our case. They have proposed models utilizing state of art methodologies like Vision Transformers and GANs to exploit the underlying patch structure of these networks in solving the shuffled jigsaw. Using them as our inspirations, we propose two approaches, one targeting a novel architecture(we decided to call it Vignette) and the second targeting the power of representation learning by removal of relative positional embedding.

4 Approach

In our research, we have decided to explore two distinct approaches, each characterized by unique loss functions, architectures, and methodologies. Given our team's diverse academic backgrounds, we thought focusing on a singular approach or codebase presents challenges. By diversifying our methods, we can not only accommodate our varied intuitions but also ensure that obstacles encountered in one approach provide valuable insights into potential challenges that could arise in the other approaches.

4.1 Approach 1: Vignette - ViG with Feature Encoder network

In the first branch of the input feature, the prepared shuffled tiles are passed through a CLIP image encoder. The CLIP (Contrastive Language-Image Pretraining) layers mentioned learn the visual context from the encoder. This step is designed to extract both high-level and low-level features from each tile. The output of the CLIP encoder is then subjected to several convolutional layers. These layers create unique feature maps that highlight specific features within the tiles.

In the second branch, we have the shuffled image containing the background data. Each tile consists of several small sub-patches, which will be subjected to separate KNN algorithm from [8] that forms a graph connection per patch(one patch is one node), thus forming several connections from one tile to other tiles and several tiles within the tiles. Since this is unsupervised clustering, these patch connections will be almost the same for the shuffled image patches and the original image patches(we have tried this, the GitHub link is attached below). The output can be seen in the architectural figure above. This will act as our feature from this second branch. These values would act as an embedding that can be used alongside the per-tile feature extracted.

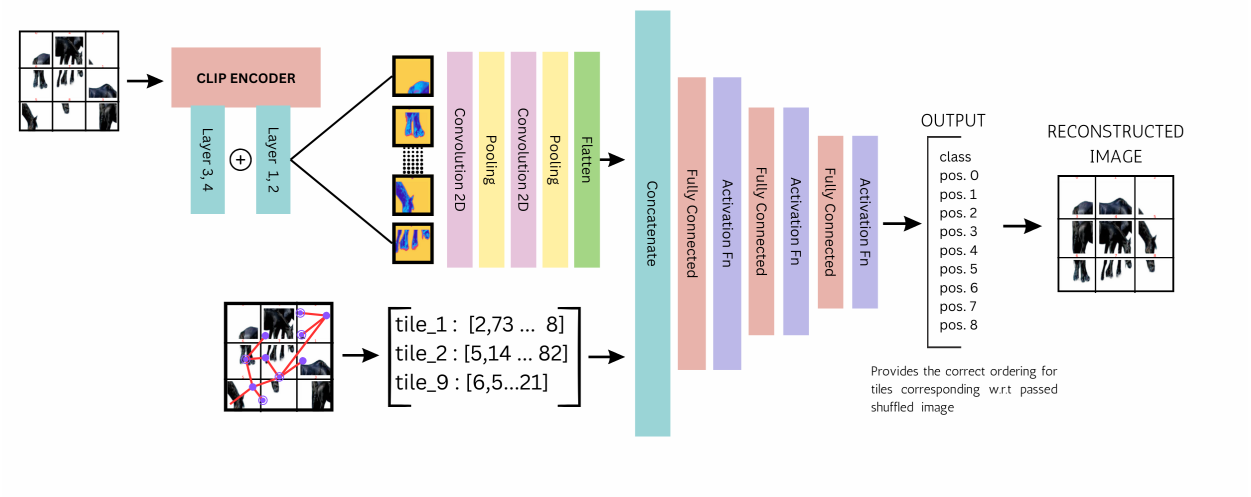

Figure 3: Approach-I Architecture

These two feature sets coupled would be then passed to a classification network. The classification module aims to distinguish different permutations, which include max pooling layers and fully connected layers. Using softmax activation would give us the classification probabilities per tile, hence giving us the right tile configuration. Similar to [11] we also plan to use the Kullback-Leibler (KL) divergence. It can measure the similarity between the predicted distribution and the target distribution. The KL divergence becomes smaller when two permutations tend to be similar. We aim to minimize the KL divergence between the reassembled result p_{predict} and the ground truth p_{real}, which can be described as follows:

$$\arg\min KL(p_{\text{predict}}, p_{\text{real}}) \tag{1}$$

We minimize the following losses to optimize the classification network:

$$L_{\text{jigsaw}}(C) = E_{x \sim P_{\text{data}}(x)}[CE(C(x), p)], \tag{2}$$

where p is the probability distribution of the real data, $C(x)$ is the probability distribution of the predicted data which indicates the probability that the result belongs to each category. p and $C(x)$ are defined as matrix with size $B \times P$, where B is the batch size. CE is cross-entropy loss which is defined as:

$$CE(C(x), p) = -\sum_i p_i \cdot \log(C(x_i)) \tag{3}$$

After the reference labels are obtained, focal (FL) loss [12] is applied as the jigsaw loss. FL loss ensures the minimization of the KL divergence whilst increasing the entropy of the predicted distribution, which can prevent the model from becoming overconfident. FL loss can down-weight easy examples and focus training on hard negatives. We replace the CE loss conventionally used with the FL loss to improve the network calibration. FL loss is defined as:

$$FL(C(x), p) = -\sum_i (1 - C(x_i))^{p_y} \cdot \log(C(x_i)) \tag{4}$$

where $(1-C(x))^{p_y}$ is a modulating factor to optimize the imbalance of the dataset and γ is a tunable focusing parameter that smoothly adjusts the rate at which easy examples are down-weighted. γ is set to 2 in the implementations in [11]. Finally, the jigsaw loss can be described as follows:

$$L_{jigsaw}(C) = E_{x \sim P_{data}(x)}[FL(C(x), \text{Pref})] \tag{5}$$

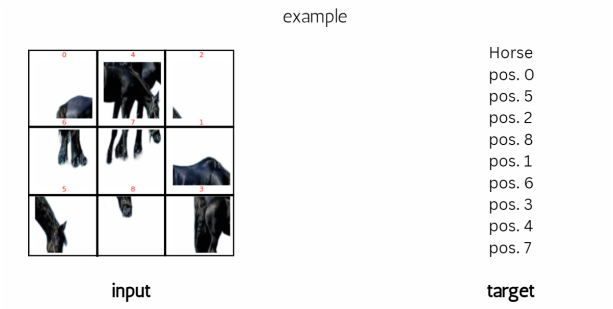

Figure 4: Approach-I Example

4.2 Approach 2: Jigsaw Vig - Relative positional embedding removal

This approach is heavily inspired by the method, authors of [2] adapted. They claim the fact that ViT works on image patches makes it potentially relevant to the problem of jigsaw puzzle solving, which is a classical self-supervised task aiming at reordering shuffled sequential image patches back to their original form. [2] also mention that, to make the ViT more robust to adversarial attacks or noisy classification problems, they have removed the positional embedding from the patch embedding, and have masked some patches of the ViT. The results they showcase are better than a vanilla ViT. Their idea of removing the positional embedding to learn the features/context of image patches makes it more suitable for the jigsaw-solving task. The second highlight of theirs is their unique loss function which also keeps into account the discarded positional embedding.

$$\mathcal{L}_{\text{total}} = \underbrace{\underbrace{\text{CE}(y_{\text{pred}}, \hat{y})}_{\mathcal{L}_{\text{jigsaw}}} + \eta \text{CE}(\overline{y}_{\text{pred}}, \hat{y})}_{\mathcal{L}_{\text{cls}}} \tag{6}$$

Here, $\tilde{y}_{pred}$ and $\tilde{y}$ denote the position prediction and the corresponding real position, respectively, and η is a hyperparameter balancing the two losses.
We feel a similar approach in terms of removal of 'relative position encoding' can be reconstructed in the Vision GNN paper [8] and the addition of the loss term which keeps into account the jigsaw relative position loss (our unique new loss function) which works similar to but not exactly same as the jigsaw loss mentioned in the [2] paper. The proposed architecture of [8] for classification detection tasks produces results better than the state-of-the-art models. Since [2] shows that the removal of positional embeddings in the jigsaw branch helps provide a consistent improvement over all tested datasets, which validates their effective design of the jigsaw branch in ViTs. We hypothesize that this similar adaptation in [8] will not only preserve the integrity of the model's classification and detection capabilities but will also enhance its proficiency in jigsaw puzzle resolution tasks.

5 GitHub

`https://github.com/gadhvirushiraj/dolphin.git`

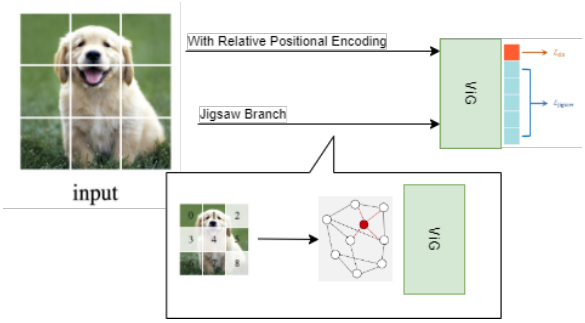

Figure 5: Approach-II Architecture

References

[1] B. J. Brown, C. Toler-Franklin, D. Nehab, M. Burns, D. Dobkin, A. Vlachopoulos, C. Doumas, S. Rusinkiewicz, and T. Weyrich. A system for high-volume acquisition and matching of fresco fragments: Reassembling theran wall paintings. *ACM Transactions on Graphics*, 27(3), 2008.

[2] Yingyi Chen, Xi Shen, Yahui Liu, Qinghua Tao, and Johan A. K. Suykens. Jigsaw-vit: Learning jigsaw puzzles in vision transformer, 2023.

[3] T. S. Cho, S. Avidan, and W. T. Freeman. A probabilistic image jigsaw puzzle solver. In *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 183–190, Jun 2010.

[4] E. Demaine and M. J. Demaine. Jigsaw puzzles, edge matching,and polyomino packing: Connections and complexity. *Graphs and Combinatorics*, 23:195–208, 2007.

[5] L. Dery, R. Mengistu, and O. Awe. Neural combinatorial optimization for solving jigsaw puzzles: A step towards unsupervised pre-training. Technical Report 110, Stanford University, 2017.

[6] H. Freeman and L. Garder. Apictorial jigsaw puzzles: The computer solution of a problem in pattern recognition. *IEEE Transactions on Electron. Comput.*, 13:118–127, 1964.

[7] A. C. Gallagher. Jigsaw puzzles with pieces of unknown orientation. In *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 382–389, 2012.

[8] Kai Han, Yunhe Wang, Jianyuan Guo, Yehui Tang, and Enhua Wu. Vision gnn: An image is worth graph of nodes, 2022.

[9] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 5, page 6, 2015.

[10] D. Koller and M. Levoy. Computer-aided reconstruction and new matches in the forma urbis romae. *Bullettino Della Commissione Archeologica Comunale di Roma*, 15:103–125, 2006.

[11] Ru Li, Shuaicheng Liu, Guangfu Wang, Guanghui Liu, and Bing Zeng. Jigsawgan: Auxiliary learning for solving jigsaw puzzles with generative adversarial networks. *IEEE Transactions on Image Processing*, 31:513–524, 2022.

[12] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection, 2018.

[13] H. Liu, S. Cao, and S. Yan. Automated assembly of shredded pieces from multiple photos. *IEEE Transactions on Multimedia*, 13(5):1154–1162, Oct 2011.

[14] W. Marande and G. Burger. Mitochondrial dna as a genomic jigsaw puzzle. *Science*, pages 318–415, 2007.

[15] M. A. O. Marques and C. O. A. Freitas. Reconstructing strip-shredded documents using color as feature matching. In *Proc. ACM Symp. Appl. Comput. (SAC)*, pages 893–894, 2009.

[16] Mehdi Noroozi and Paolo Favaro. Unsupervised learning of visual representations by solving jigsaw puzzles, 2017.

[17] G. Paikin and A. Tal. Solving multiple square jigsaw puzzles with missing pieces. In *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, pages 4832–4839, Jun 2015.

[18] C. Papaodysseus, T. Panagopoulos, M. Exarhos, C. Triantafillou, D. Fragoulis, and C. Doumas. Contour-shape based reconstruction of fragmented, 1600 bc wall paintings. *IEEE Transactions on Signal Processing*, 50(6):1277–1288, Jun 2002.

[19] M.-M. Paumard, D. Picard, and H. Tabia. Deepzzle: Solving visual jigsaw puzzles with deep learning and shortest path optimization. *IEEE Transactions on Image Processing*, 29:3569–3581, 2020.

[20] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. 2014.

[21] R. Yu, C. Russell, and L. Agapito. Solving jigsaw puzzles with linear programming, 2015.