Solving Jigsaw Puzzle Using Deep Learning

Course CSE472: Machine Learning Sessional

Fabiha Tasneem
1805072
Computer Science & Engineering
Bangladesh University of Engineering & Technology
Dhaka, Bangladesh

Sumaiya Sultana
1805079
Computer Science & Engineering
Bangladesh University of Engineering & Technology
Dhaka, Bangladesh

ABSTRACT

This project presents a robust way to solve Jigsaw Puzzles using deep learning methods. The baseline model is a simple multi-layer neural network with a Sparse Categorical Cross-Entropy loss function. To improve the accuracy, we also implemented VGGNet [5] and ResNet [3] architecture, considering Convolutional Neural Network (CNN)'s good performance on computer vision tasks. Several modifications were applied to the convolutional and the output layer of the CNN models to make them both computationally efficient and compatible with our task.

KEYWORDS

Computer Vision, Deep Learning, Convolutional Neural Networks, VGGNet, ResNet

1 INTRODUCTION

Invented centuries ago, the jigsaw puzzle is a famous intellectual game in which people assemble a picture that has been cut into pieces. Depending on the complexity of the pictures and the number of pieces (pixels), jigsaw puzzles can vary in difficulty. For human vision, the challenge is that the matching between the pixels and the template picture is time-consuming, which is why computers can help. A myriad of machine learning methods have been proposed to solve the jigsaw puzzles.

Numerous machine learning approaches have emerged to tackle the jigsaw puzzle problem. For instance, [2] pioneered an unsupervised learning method leveraging domain generalization to reconstruct images from their disordered components. Meanwhile, [4] introduced a neural network architecture capable of predicting the relative positions of puzzle pieces and determining the shortest path to reassemble the image.

Regarding the work done by the predecessors, we are motivated to base our project on solving the jigsaw puzzles by applying machine learning techniques. The inputs of our problem are 2x2 or 3x3 squares of 200 x 200 RGB images with shuffled pieces. We then use neural networks to output the predicted position for each image piece inside a picture. With those predicted positions, we can reassemble the pieces to reconstruct the correct image, which mimics the process of the jigsaw game.

2 RELATED WORK

Numerous studies have delved into image recognition and solving jigsaw puzzles through computer vision techniques. These approaches typically fall into two categories: supervised and unsupervised learning methods.

In supervised learning, researchers in [4] tackle complex datasets where image pieces are widely spaced, rendering patterns and color continuity largely unusable. Employing a two-step approach, they first use a neural network to predict piece locations and then optimize the reassembly path using a graph based on these predictions. In our project, we adopt the neural network method for piece predictions but forgo developing an optimization algorithm for reassembly. Instead, we directly utilize the predicted labels for accurate image construction.

In another study focusing on the depth of Convolutional Neural Network (CNN) models, researchers explore its impact on accuracy in large-scale image recognition. Their findings suggest significant accuracy improvements by increasing network depth to 16-19 weight layers. CNN methods prove highly

effective in image classification tasks, with representations that generalize well across datasets, achieving state-of-the-art results. Leveraging these insights, we further refine and tune our CNN models to enhance accuracy in supervised learning predictions.

In our view, given well-labeled datasets, deep learning methods exhibit superior performance in image label prediction. We anticipate achieving even higher accuracy by fine-tuning CNN architectures compared to utilizing multi-layer neural networks, which we used as our baseline model.

3 DATASET

The dataset has both 2×2 and 3×3 jigsaw puzzles. For our course project, we have opted to focus on 2×2 jigsaw puzzles to ensure simplicity and practical runtime, utilizing a dataset obtained from Kaggle [1]. As depicted in Figure 1, the 2×2 jigsaw samples consist of 200×200 RGB images with shuffled pieces. Their labels, representing solutions, are vectors with $2^2 = 4$ integer elements ranging from 0 to 3 which feature $4! = 24$ solutions. Meanwhile, the 3×3 jigsaw samples are comprised of 201×201 RGB images, with solution vectors consisting of $3^2 = 9$ integer elements valued between 0 and 8.

In this project, we will not be dealing with the 3×3 puzzles due to computational resource restraint. We are mindful of the significantly increased complexity associated with 3x3 puzzles, which feature a staggering $9! = 362,880$ possible solutions. As such, tackling 3×3 puzzles will require extensive computational resources and time investment.

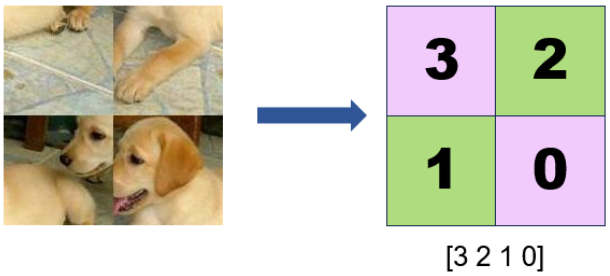

Figure 1: 2×2 jigsaw puzzle sample picture with their location matrix of corresponding labels

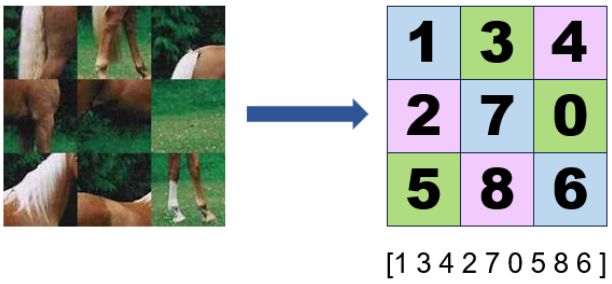

Figure 2: 3×3 jigsaw puzzle sample picture with their location matrix of corresponding labels

For the 2×2 training tasks, we have a total of 96,000 samples. Specifically, the training set comprises 93,136 samples, the validation set contains 385 samples, and the test set consists of 2,176 samples. To streamline our training process, we reduce our train dataset size by 80% so that it takes less time and uses less computational resources. While this adjustment does lead to a degradation in model performance, it allows us to significantly reduce training time and resource consumption. Despite the trade-off, we believe this approach is justified given our project constraints and objectives.

To enhance the quality and balance of a dataset intended for training machine learning models to solve 2×2 jigsaw puzzles, we first filter the dataset based on predefined label classes, each representing a unique permutation of jigsaw puzzle pieces. We ensure that an equal number of images is selected for each class. This approach is crucial for preventing bias in model training and improving its ability to generalize across different puzzle configurations.

4 MATERIALS & METHODS

To construct a resilient machine learning model capable of autonomously reconstructing shuffled jigsaw images to their original form, we employ three distinct methodologies, each offering varying degrees of comparison and complexity.

Initially, we establish a baseline model by implementing a multi-layer neural network. In pursuit of enhanced accuracy, we transition to Convolutional Neural Networks (CNN), structured by the

renowned ResNet architecture and the VGGNet architecture. Leveraging the power of convolutional layers, these methods aim to extract intricate features from the jigsaw images, facilitating more precise reconstruction.

4.1 Baseline Model Architecture and Training Procedure

Our baseline model was a multilayer neural network tailored for solving 2x2 jigsaw puzzles by predicting tile permutations. It employed early stopping with a patience of 8 epochs and model checkpointing to preserve the best weights. The architecture comprised densely connected layers with ReLU activation, followed by batch normalization and dropout regularization. Trained with RMSprop optimizer and sparse categorical cross-entropy loss, the model underwent evaluation on a separate test dataset to gauge performance. Ultimately, it served as a foundational benchmark for more complex architectures.

4.2 ResNet Model Architecture and Training Procedure

This model architecture is based on ResNetV2-50, a variant of the ResNet model known for its deep architecture and improved performance. We utilize transfer learning by leveraging pre-trained weights from the ImageNet dataset.

The base ResNetV2-50 model is modified by adding custom layers on top, including global average pooling, dense layers with ReLU activation, batch normalization, and dropout regularization. The model is trained using the Adam optimizer with a learning rate of 0.001 and a sparse categorical cross-entropy loss function. We employ early stopping with a patience of 8 epochs to prevent overfitting, model checkpointing to save the best-performing model, and a learning rate scheduler to adjust the learning rate by reducing it by 20% every 4 epochs.

4.3 VGGNet Model Architecture and Training Procedure

This model architecture, built upon the VGG16 convolutional neural network, was tailored for solving 2×2 jigsaw puzzles by predicting the correct permutation of image tiles. We fine-tuned this model by freezing all but the last four layers, allowing us to leverage the learned features while adapting the model to our specific puzzle-solving task.

On top of the base VGG16 layers, we appended custom dense layers to capture high-level features and facilitate classification. These layers included a Global Average Pooling layer for dimensionality reduction, followed by fully connected layers with ReLU activation, L2 regularization, batch normalization, and dropout for improved robustness and generalization.

During training, we dynamically adjusted the learning rate using a learning rate scheduler, reducing the learning rate by 20% every 4 epochs to facilitate convergence and improve model stability.

5 RESULTS

The training experiment was conducted for 2×2 Jigsaw puzzles. We use **accuracy** as our primary metric. The performance of the models is compared by evaluating their accuracy on the test set. Continuous hyperparameter tuning selected all the hyperparameters (e.g., batch size and learning rate). We tuned each hyperparameter across a wide range of values and chose a combination with the best performance on the test set.

Figure 3 shows a successful case when the VGGNet helped us to unscramble the shuffled jigsaw image (on the left) to produce the original image (on the right).

Figure 3: Successful Case for 2×2

Likewise, Figure 4 shows an unsuccessful case of the shuffled jigsaw image (on the left) that produced the wrong output (on the right).

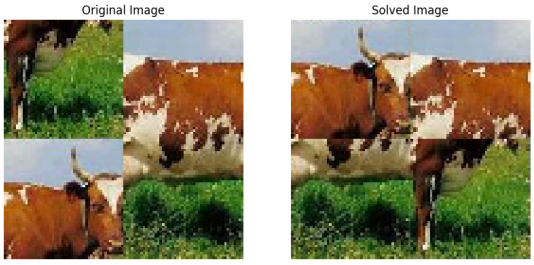

Figure 4: Unsuccessful Case for 2 × 2

5.1 Baseline Model

Our experimentations have revealed noteworthy insights into the performance of different models for jigsaw puzzle prediction. Firstly, the baseline model, employing a multilayer neural network, yielded modest results with a final test accuracy of **13.05%**. While this accuracy is low compared to more sophisticated models, it provides a baseline for comparison and underscores the complexity of the task.

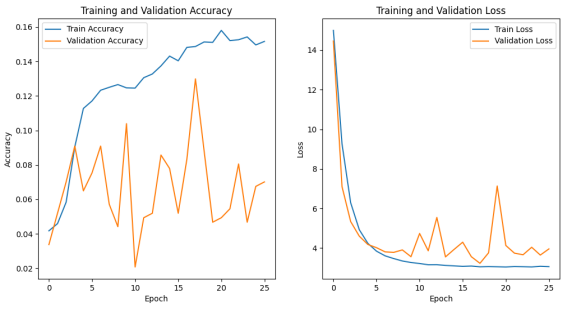

Figure 5: 2x2 Training Plots for Baseline Neural Network Model

5.2 ResNet Performance

The ResNet model achieved a final test accuracy of **48.76%** and the accuracy of the best model reached **49.31%**. These results indicate that the ResNet architecture performed reasonably well in solving the 2x2 puzzles.

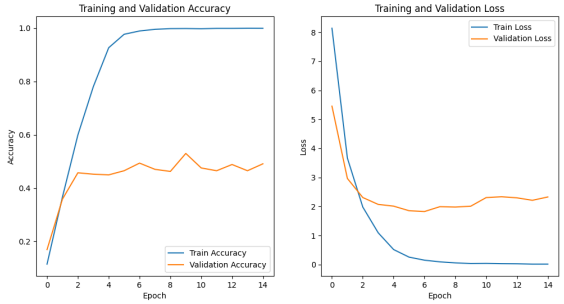

Figure 6: 2x2 Training Plots for ResNet Model

5.3 VGGNet Performance

On the other hand, the VGGNet model showed a slightly higher performance compared to ResNet. It achieved a final test accuracy of **57.31%** and the best model accuracy is **63.01%**. This indicates that the VGGNet architecture performed consistently well throughout the training process, without significant fluctuations or challenges in convergence.

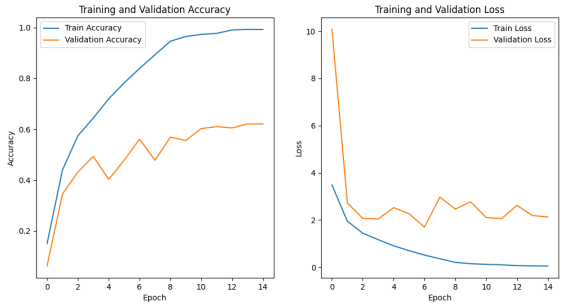

Figure 7: 2x2 Training Plots for VGGNet Model

6 DISCUSSION

These results suggest that deeper and more complex neural network architectures, such as ResNet and VGGNet, are better suited for solving jigsaw puzzles compared to simpler models. The superior performance of VGGNet indicates the effectiveness of its architecture in capturing intricate patterns and features within the jigsaw puzzle images.

6.1 Comparison and Implications

Comparing the performance of ResNet and VGGNet, we observe that VGGNet outperformed ResNet in terms of both final test accuracy and the accuracy of the best model.

Table 1: Performance of Different Models for 2x2 puzzles

Model	Best Model Test Accuracy
Baseline Model	13%
ResNet	49%
VGGNet	63%

7 CONCLUSION

In conclusion, our investigation has shown that VGGNet outperforms both the ResNet model and the baseline model when tasked with predicting 2x2 Jigsaw puzzles. This underscores the effectiveness of the VGGNet architecture in accurately reconstructing images from their shuffled components.

8 FUTURE WORK

In the future, strategies can be explored to mitigate the impact of train dataset reduction on model performance using High-Performance Computers, potentially through techniques such as data augmentation or transfer learning.

Furthermore, while our current focus has been on 2x2 puzzles with 24 possible solutions, we acknowledge the potential benefits of expanding our scope to include 3x3 puzzles but it will require extensive computational resources and time investment.

Moving forward, the application of more advanced deep learning architectures can be explored, such as Swin Transformer and EfficientNet, to further elevate our model's performance. By embracing these avenues for improvement, we aim to advance the field of jigsaw puzzle solving and contribute to the development of robust and efficient solutions.

9 ACKNOWLEDGMENTS

We extend our sincere gratitude to our project supervisor, Sheikh Azizul Hakim Sir, for his guidance and support throughout the research and preparation phases of our deep learning project.

REFERENCES

[1] Corrado Alessio. 2020. 100K Jigsaw Puzzle Images. https://www.kaggle.com/datasets/shiva-jbd/jigsawpuzzle.

[2] Fabio Maria Carlucci, Antonio D'Innocente, Silvia Bucci, Barbara Caputo, and Tatiana Tommasi. 2019. Domain Generalization by Solving Jigsaw Puzzles. arXiv:1903.06864 [cs.CV]

[3] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 770–778. https://doi.org/10.1109/CVPR.2016.90

[4] Marie-Morgane Paumard, David Picard, and Hedi Tabia. 2020. Deepzzle: Solving Visual Jigsaw Puzzles With Deep Learning and Shortest Path Optimization. *IEEE Transactions on Image Processing* 29 (2020), 3569–3581. https://doi.org/10.1109/TIP.2019.2963378

[5] Karen Simonyan and Andrew Zisserman. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv:1409.1556 [cs.CV]