AI Puzzle Solver

CSC 668 Artificial Intelligence

Katherine McGill, Sai Tarun, Jainab Khanam, and Amy Mailo

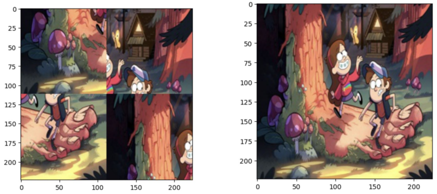
Introduction

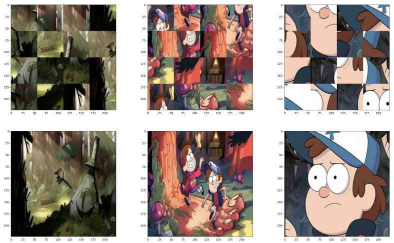
Goal

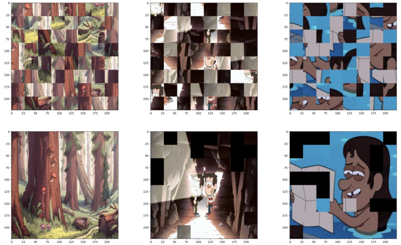
- Analyze an existing solution for solving jigsaw puzzles (2x2, 4x4, 8x8), evaluate it using new data, and attempt to improve performance

What does this code do?

- Implements a neural network-based approach to solve jigsaw puzzles, using a U-net-like architecture to predict the permutation of pieces in the puzzle
- Model trained to reconstruct correct arrangement of puzzle pieces from shuffled inputs







Methodology and Data

Data Preparation

- Data: Mammals classification (image dataset)
  - Modified original by combining all images into one folder (since we are not using them for classification) and removing unsuitable images
- Resized images to 224 x 224, 110 images randomly selected
- Images transformed into puzzles and split into train, test, and validation sets
- Process used to create different datasets and CSV files for 2x2, 4x4, and 8x8

Puzzle Solving

- Executed the existing code with the new data for all puzzle sizes
  - Reported training accuracy and validation accuracy (Note: refers to correct pieces [or tiles], not correct puzzles)
  - Printed samples of puzzles and the solutions predicted by the model
- Compared performance to baseline (performance of the existing method)
- Noted performance issues and developed/tested solutions to address them

2x2 Puzzle Results

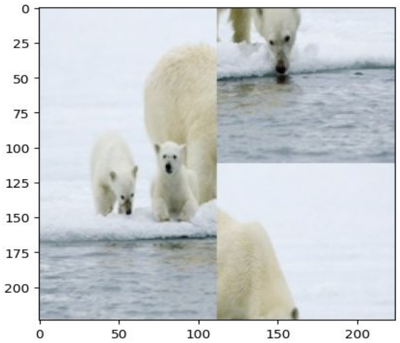
Train: 440 images, Test: 110 images, Validation: 110 images

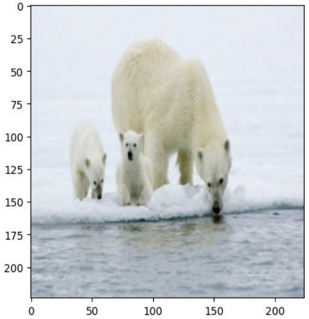
- ➢ Training accuracy (sparse_categorical_accuracy): 99.6%
- ➢ Validation accuracy (val_sparse_categorical_accuracy): 75.6%

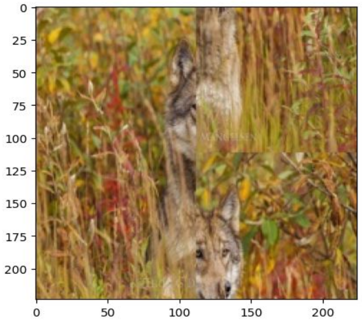
Baseline Performance

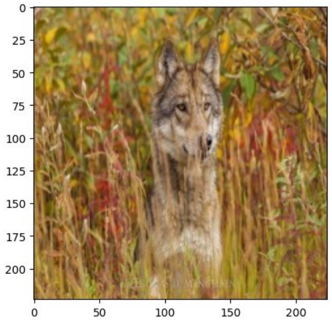
Train Acc: 100%

Val Acc: 81.14%









4x4 Puzzle Results

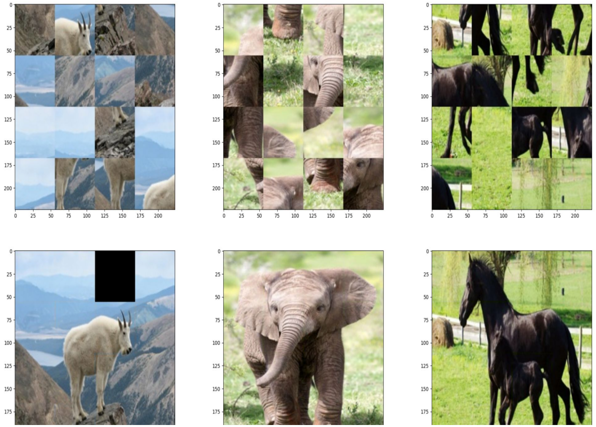
Train: 770 images, Test: 110 images, Validation: 110 images

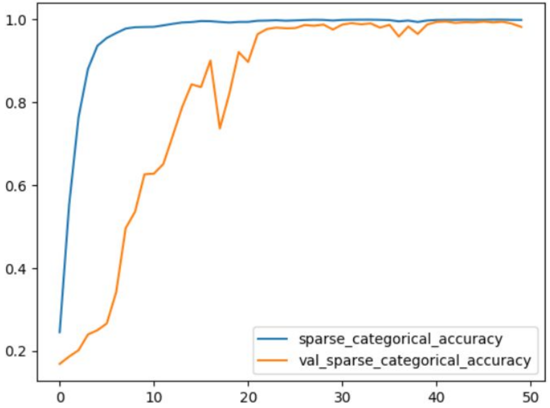
- Training accuracy (sparse_categorical_accuracy): 99.7%
- Validation accuracy (val_sparse_categorical_accuracy): 93.8%

<u>Baseline Performance</u>

Train Acc: 99.98%

Val Acc: 99.49%



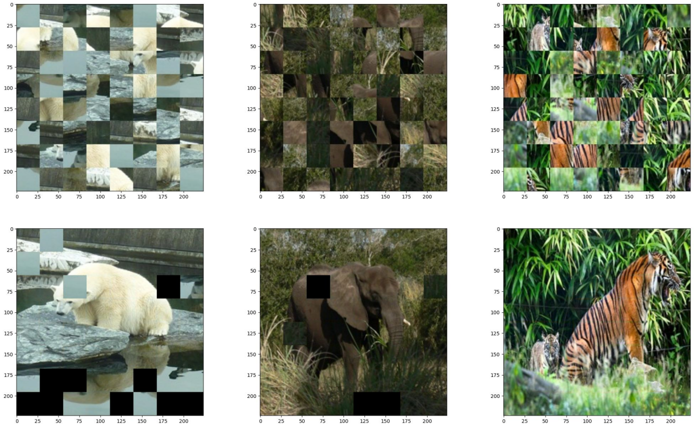


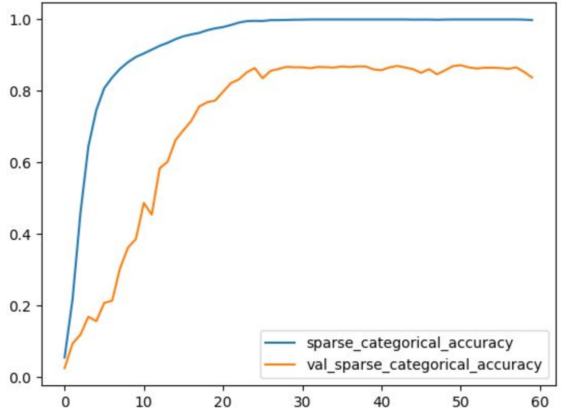
8x8 Puzzle Results

Train: 1100 images, Test: 110 images, Validation: 110 images

- ➢ Training accuracy (sparse_categorical_accuracy) 99.8%
- ➢ Validation accuracy (val_sparse_categorical_accuracy) 83.8%

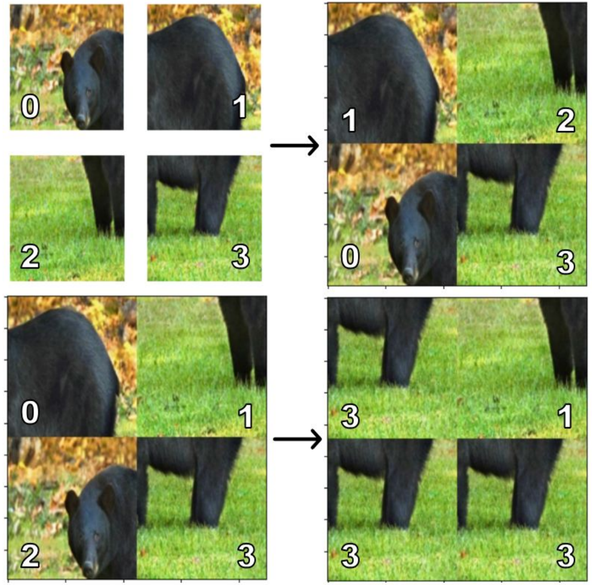
<u>Baseline Performance</u>
Train Acc: 99.99%
Val Acc: 86.11%





Concerns

- Results were not reproducible for 2x2
  - Failure Cases:
    - Good accuracy, but unsolved puzzles
  - Potential Issues:
    - Tile numbers are reassigned
    - Model is given the puzzle order as y instead of the solution order (more difficult to solve)
- Training, testing, and validation sets were all created from the same images. The puzzles may be unique, but possible overlap.

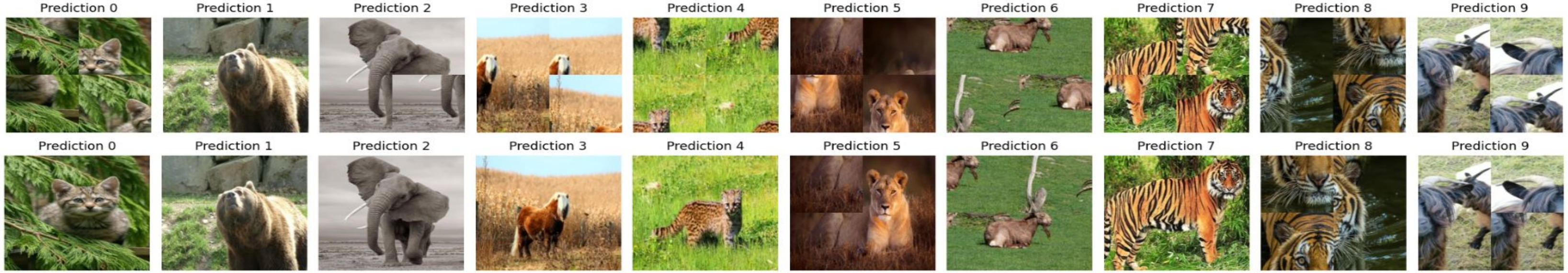


Our Solutions

- Split original 709 images into training, testing, and validation sets prior to creating the puzzles (to ensure there is no chance of test puzzles being in the training data)
    - 70/15/15; Train: 509 images, Test: 110 images, Validation: 110 images
    - 80/10/10; Train: 583 images, Test: 73 images, Validation: 73 images
- Create new CSV files that reflect the solution order (given the reassignment of the tile numbers)
- Evaluate solutions with the 2x2 puzzle solving code
    - Compare performance of the model using the original CSV files to the performance using the new CSV files
    - Compare performance of the different train/test/validation splits
- Prediction: the baseline data splitting method with the new CSV will have the best performance (because the images used for the splits are not entirely unique)

Comparison of Results for 2x2 puzzles

Performance	Baseline Original CSV	Baseline New CSV	70/15/15 Original CSV	70/15/15 New CSV	80/10/10 Original CSV	80/10/10 New CSV
Val Accuracy	80.23%	75.68%	65.68%	67.05%	62.33%	64.04%
# Solved	1/10	6/10	0/10	3/10	2/10	3/10



The table shows the highest reported accuracy in 100 epochs. The training accuracy was either 100% or nearly that for all attempts. This indicates overfitting when compared to the validation accuracies. The sample images are the performance comparison for the baseline data splitting method with original CSV (top) and new CSV (bottom).

Conclusions and Next Steps

- Conclusions
  - Original code worked well with 4x4 and 8x8 puzzles, but worse with 2x2
  - Adding code to convert the CSV tile/piece order did not always improve the reported accuracy, but did increase the number of solved puzzles
  - Even with our modifications, performance was not great when unique images were used in test set (model is still overfitting)
- Next Steps
  - Try additional methods to improve performance/ reduce overfitting:
    - Increasing size of the training data
    - Adjusting the number of layers in the model (possibly simplifying)
    - Tuning the model hyperparameters
  - Evaluate and compare performance on different (more varied) images

Important Links

- Existing Method: https://www.kaggle.com/datasets/serhiibiruk/jigsaw-puzzle
- Data Source: https://www.kaggle.com/datasets/anirudhg15/mammals-classification
- Our GitHub Repository: https://github.com/amymailo/solvingpuzzles
- Our Data: https://kaggle.com/datasets/bbec2171f34965a432783efb7b709e6a7ff5a3e8cde6b2fe5060fe29bb6d111f