Machine Learning Project Report

Arjun Krishnan (2022A4PS0617G)
Padi Sudeep (2022A4PS0527G)
Chaitanya Joshi (2022A7PS0730G)
Satyam Shukla (2022A3PS0527G)

1. Problem Statement

The objective of this project is to evaluate the capability of current Machine Learning (ML) algorithms in performing image transformations based on visual patterns, specifically using the IPARC dataset.

The hypothesis is: **A fully connected neural network can accurately predict which operator was used in an image transformation defined by a simple morphological operation.** The extent to which the ML model can achieve high accuracy in classifying input-output image pairs into any one of the eight different operations will determine the validity of this hypothesis.

2. Methodology

2.1 Data Preparation

- There is a dataset made separately for each of dilation and erosion.
- The dataset comprises multiple JSON files, each containing pairs of input and output images (15x15 pixels each) along with a transformation label.
- A task is defined by an operation (dilation or erosion) and an operator (SE1 - SE8). So there are eight separate task files for dilation and erosion.
- There are 10,000 image pairs in each task file. The total of 80,000 image pairs (per model) are shuffled and split into the train and test sets.
- An image transformation here is defined by an operator doing an operation on the input image, resulting in an output image.
- Given the input-output image pairs and the operation used (so separately taking the dilation and erosion datasets), the model will need to classify the operation by labeling which one of the eight operators was used.
- Preprocessing steps include:
  - Normalizing pixel values.
  - Converting the data into tensors suitable for deep learning models.

2.2 Model Architecture

- A neural network with fully connected layers:
  - **Input Layer:** Flattens the input (2x15x15) images into a single-dimensional array.
  - **Hidden Layers:** ReLU activation is used.
  - **Output Layer:** Outputs one of the eight operators used, using the softmax function.

2.3 Training Process

- The dataset was split into training (90%) and testing (10%) subsets.
- **Loss Function:** Negative Log-Likelihood Loss (NLLLoss).
- **Optimizer:** Adadelta optimizer with a learning rate scheduler to adjust the learning rate dynamically.
- **Evaluation Metrics:** Accuracy and loss per transformation class.

2.4 Adadelta Optimizer

The Adadelta optimizer was chosen for this project due to its advantages over traditional stochastic gradient descent (SGD). Adadelta is an extension of the Adagrad algorithm, which adapts the learning rate for each parameter individually, but unlike Adagrad, it does not accumulate the squared gradients over all time steps. Instead, Adadelta maintains a moving window of past squared gradients, which allows the learning rate to adjust dynamically over time.

Adadelta has the following advantages:

- No Need for Manual Learning Rate Tuning: It adjusts the learning rate dynamically based on the gradients, eliminating the need to tune the learning rate manually.
- Efficient Memory Usage: By using a moving average of squared gradients, it uses significantly less memory compared to storing all historical gradients.
- Faster Convergence: It has been shown to converge faster in many machine learning tasks compared to SGD, especially in cases where the dataset or model is large and complex.

For these reasons, Adadelta is well-suited for this task, where image transformations are learned, and the model needs to adapt quickly to various input patterns.

2.5 Model Selection

Various configurations were tested by changing the number of layers, batch size, and learning rate. Results are summarized below:

Based on average accuracy across all operations, **D1** performed best.

Model	Layers	Batch Size	Learning Rate	ReLU
D1	2	32	0.5	Yes
D2	3	32	0.5	No
D3	4	64	1.0	Yes
D4	3	32	0.6	Yes
D5	2	128	0.6	Yes

Table 1: Model configurations and key hyperparameters for dilation.

3. Experimental Results

3.1 Model Performance for Dilation

The model was trained for 20 epochs, adjusting the learning rate dynamically using a learning rate scheduler. Test accuracies for the best-performing model (**D1**) are shown below:

Structuring Element (SE)	Test Accuracy (D1)
SE1	72.57%
SE2	53.09%
SE3	49.23%
SE4	69.48%
SE5	99.80%
SE6	99.79%
SE7	100.00%
SE8	100.00%

Table 2: Test accuracy for each structuring element (SE) for dilation (model D1).

3.2 Error Analysis for Dilation

- The best performance was observed for SE5, SE6, SE7, and SE8, achieving near-perfect accuracy. These transformations involve straightforward visual patterns, which the network could learn easily.
- The worst performance was for SE3 and SE2, with accuracies of 49.23% and 53.09%, respectively. These operations involve more complex patterns.
- Reducing the number of layers from three to two (e.g., D1 vs. D2) generally improved accuracy, suggesting simpler models perform better for this task.

3.3 Model Performance for Erosion

For the erosion operation, ReLU activation was used in all trained models. Accuracy improved as the number of layers increased. The tested configurations are summarized below:

The highest average accuracy was observed for **E2**. Its accuracies across structuring elements are as follows:

Model	Layers	Batch Size	Learning Rate	ReLU
E1	2	128	0.5	Yes
E2	4	64	0.8	Yes
E3	6	64	0.8	Yes

Table 3: Model configurations and key hyperparameters for erosion.

Structuring Element (SE)	Test Accuracy (E2)
SE1	46.47%
SE2	42.12%
SE3	33.75%
SE4	50.70%
SE5	98.62%
SE6	96.65%
SE7	97.82%
SE8	98.10%

Table 4: Test accuracy for each structuring element (SE) for erosion (model E2).

3.4 Error Analysis for Erosion

- Similar to dilation, SE5, SE7, and SE8 showed the best performance due to their simpler patterns.
- The lowest accuracy was observed for SE3 and SE2, highlighting difficulty in handling more complex visual transformations.
- Increasing the number of layers consistently improved accuracy, indicating that more complex models better captured erosion transformations.

4. Conclusion and Future Work

4.1 Conclusion

The hypothesis that "a fully connected neural network can accurately predict which operator was used in an image transformation defined by a simple morphological operation" is **FALSE**. The neural network performed well on simpler transformations but struggled with complex patterns.

4.2 Future Work

- Accomodate for the current limitations in computational power, so that lower values of learning rates, higher batch sizes and more hidden layers can be used.
- Introduce convolutional neural networks (CNNs) for enhanced feature extraction.
- Use transfer learning to leverage pre-trained models for improved generalization.
- Explore hybrid approaches combining symbolic reasoning with ML techniques.

- Expand the dataset to include diverse and challenging transformations.

5. References

This project took inspiration from the MNIST classification task for handwritten digit recongition. (https://www.kaggle.com/code/imdevskp/digits-mnist-classification-using-cnn)