Maddie Lebiedzinski and Ethan Kong

Image Classification with Convolutional Networks

Problem: Multiclass image classification, identifying whether an image is representative of one 3 labels: rock, paper, or scissors

Contributions: Collaborated on ideas and concept
Maddie - focused on building the CNN model and training the model.
Ethan - found the data, graphed result data and evaluated the experiment results
Collaborated on writeup, final evaluation, and presentation.

Sources:
Dataset from kaggle:

- https://www.kaggle.com/datasets/sanikamal/rock-paper-scissors-dataset
- Contains 2892 unique images of hands demonstration rock, paper, or scissors
- Our data set of images is split into training, validation, and test sets.

Libraries and imports:

- Tensorflow:
  - Image Data Generator, Image - allows for data manipulation and preprocessing, such as rescaling or rotating of images which can help improve the model's performance and generalization.
  - RMSprop - optimizes performance by adapting the learning rate for each parameter during training, allowing the model to converge faster and potentially improve performance
- Matlab: plots the performance graphs of the model
- Numpy: converting images to a numpy array and manipulating its dimensions to match the expected shape
- Other imports were used to read files

Cell 2&3 : file paths, loading and visualizing the images

Cell 4 : cv2.imread is used to load an image and .shape is getting dimensions of the image. If all the images were different this could also be used to uniformly resize all of them.

Cell 5&6 : generating the training :

- Train.from_flow_directory: provides batches of preprocessed images and their corresponding labels during the model training process.

    - Batch: divides the datasets into subsets allowing the model to more efficiently process the data

Cell 7&8 : Values associated with the images; paper is 0, rock is 1, scissors, is 2

Cell 9&10 : Constructing the model

Constructed the CNN model by defining multiple layers; convolutional layer, pooling layer, Dense layer, etc.

Conv2D layer: This layer performs convolutional operations on the input image. It applies a specified number of filters (16, 32, and 64 in this case) to the input image, each filter sliding over the image to extract features

MaxPooling2D layer: This layer performs max pooling operations to help in reducing the computational complexity and provides translational invariance, allowing the model to focus on the most important features.
Flatten layer: This layer is used to convert the 2D feature maps obtained from the previous layers into a 1D feature vector.

Dense layers: These layers are fully connected layers, where each neuron is connected to every neuron in the previous layer. The first Dense layer has 2892 neurons and uses the ReLU activation function. It acts as a hidden layer and helps to learn more complex and higher-level features. The second Dense layer has 3 neurons, representing the three classes (paper, rock, scissors) in the output. The activation function used is softmax, which converts the final layer's output into probabilities for each class

With summary of model
Then Compile the model

Cell 11 : Training the model
Model fitting is a measure of how well a machine learning model generalizes to similar data to the data on which it was trained.

The epochs and steps per epoch represent the total number of iterations of all the training data in one cycle for training the machine learning model.

Cell : Performance Graphs

Visualized the training results by graphing accuracy and loss

Loss vs Validation loss:

- The loss value represents the discrepancy between the predicted output of the model and the true labels for the training data
- The validation loss helps to assess how well the model is performing on unseen data and serves as a measure of overfitting or underfitting. The objective is to achieve low validation loss, indicating that the model performs well on unseen data and has good generalization capabilities.

Accuracy vs Val Accuracy:

- Accuracy is a metric that measures the proportion of correct predictions made by the model out of the total number of predictions
- Validation accuracy provides an estimate of the model's performance on unseen data and helps evaluate its generalization ability.

Purpose of validation:

- Analyze current model performance on unseen data
- Improve model based on analysis

Cell : Testing images

Then the model is tested on a series of test images, where here you can visualize its prediction for each image.

Analysis of your experiments, describing what you learned about your data set, and what would be the next steps if you continued with the project.

- Analysis: Model can be greatly improved during validation by using an unseen dataset using hyperparameters, adjusting model architecture, or increasing the training data can be taken to improve the model's generalization and performance.
- **Outlook and use cases**: Sign-language translator/detector that can output cohesive paragraphs from sign-language