CSE 535 Mobile Computing Project 2
Project Group 5

Yashwanth Kumar Tirupati: 1225424512, Tejaskumar Patil: 1225501316, Baibhav Phukan: 1225408392, Pranav Toggi: 1221278773, Aditya Goyal: 1225689049

Introduction:

The project's goal is to use the android application created in the previous project to upload images of handwritten digits captured by the camera to the server and classify them to place them in their respective folder. The project involves building a deep learning model to classify handwritten digits by training it using the MNIST dataset.

The project uses the TensorFlow python library to build a deep learning model and Pillow libraries for image processing. It also uses the NumPy library to perform array operations and Pickle to save the trained deep learning model in a file.

MNIST Data Augmentation:

The images available in the MNIST dataset are idealistic and do not account for the practical aspects of capturing an image of a handwritten digit. The captured images might be slightly rotated, might be of varied sizes, and the positioning of the digit might be skewed to the edges. To handle such variations in the captured images, we augment the MNIST images by applying random rotation, sheer, shift, and zoom before training the model with these images. This augmentation helps the model classify the images robustly, even when the input images are subjected to such practical variations.

Deep Learning Model for classifying handwritten images:

The model we developed uses convolutional neural networks (CNNs) because they are better at capturing the spatial dependencies in an image. In classifying images, CNNs perform much better than traditional feed-forward neural networks because CNNs can efficiently extract dominant features and suppress noise factors to a great extent.

Adam optimizer was used on the categorical cross-entropy loss function for training the model because it has a faster run time and low memory requirements and requires less tuning than any other optimization algorithm. The training was done in input batches of size 256. The model was trained for 30 epochs employed with early stopping using validation loss to prevent the network from overfitting the data.

The model's accuracy was estimated at around 98.8% using test data from the MNIST dataset. This model is saved to a file using the pickle python library to use it in the server to categorize the images received from the front-end application.

The accuracy of the model when tested on the test data from the MNIST dataset was estimated to be around 99.28%. This model is saved to a file using pickle python library so that it can be used in the server to categorize the received images.

Preprocessing for captured handwritten digit images:

The captured handwritten digit images sent to the server by the front-end application are colored images usually affected by factors like lighting conditions, the writing instrument used, and camera noise. Therefore, it is necessary to preprocess these images before using the above deep learning model to classify them. The preprocessing is done as follows:

1. Convert the RGB-colored image to a grayscale image as the model is trained using grayscale images from the MNIST dataset.
2. Binarize the image using a predefined threshold value. This generates a crisp image of the digit with sharp contrast making it easy for the model to classify.
3. Invert the image if the background of the image is white. This is done because the model is particularly trained with images with a black background.
4. Increase the thickness of the digits by modifying the values of the neighboring pixels around the digit. This is useful when the digits are written using a thin writing instrument like a ballpoint pen.
5. Smoothen the image and remove noise from the image using image blurring using averaging to enhance the image quality.
6. Resize the image to 28x28 pixels, as this is the required input size for the CNN model.

Model Analysis on smartphone captured images:

The model's accuracy was estimated to be around 96.62% when tested on images of digits written using a marker and captured using a smartphone under normal lighting conditions.

References:

1. https://www.tensorflow.org/api_docs
2. https://pillow.readthedocs.io/en/stable/