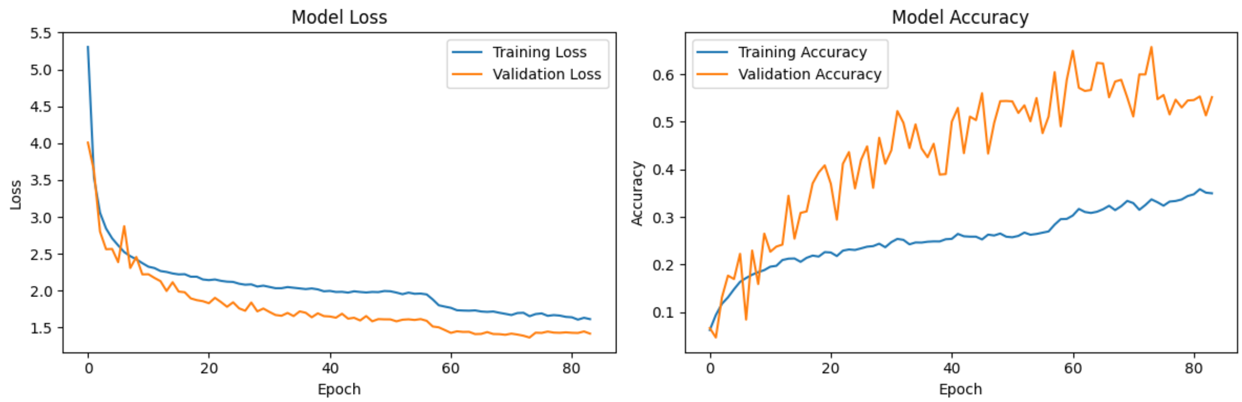
Training

1. **Data Loading and Preprocessing:**
    - Data and labels are loaded from multiple .npy files.
    - Features are normalized to the range [0, 1] to ensure stable training.
    - Data is reshaped to include a single channel dimension, as the images are grayscale.

2. **Target Binning:**
    - Continuous labels are discretized into a specified number of bins (default: 50).
    - Each bin represents a range of values, transforming the regression problem into classification.

3. **Model Architecture:**
   A CNN model is constructed using TensorFlow's Keras API. Key design elements include:
    - Three Convolutional Blocks:
        - Sequential convolutional layers with increasing filter sizes (64 → 128 → 256).
        - Batch normalization for stabilizing training.
        - Activation functions (ReLU) for non-linearity.
        - Max-pooling for spatial down-sampling and dropout for regularization.
    - Fully Connected Layers:
        - Dense layers with L2 regularization to prevent overfitting.
        - Batch normalization and dropout for further stability.
        - Final softmax activation for multi-class classification.
    - The model is compiled with the Adam optimizer, categorical cross-entropy loss, and accuracy metric.
4. **Training:**
    - The dataset is split into training and testing sets, with further division of the training set for validation.
    - Early stopping and learning rate reduction callbacks are used to prevent overfitting and adapt learning rates during training.
    - The model is trained over a maximum of 100 epochs with a batch size of 32.
5. **Evaluation:**
    - The training history is plotted, showing loss and accuracy trends for both training and validation sets.
    - Model weights and bin edges are saved for future inference.

Results:

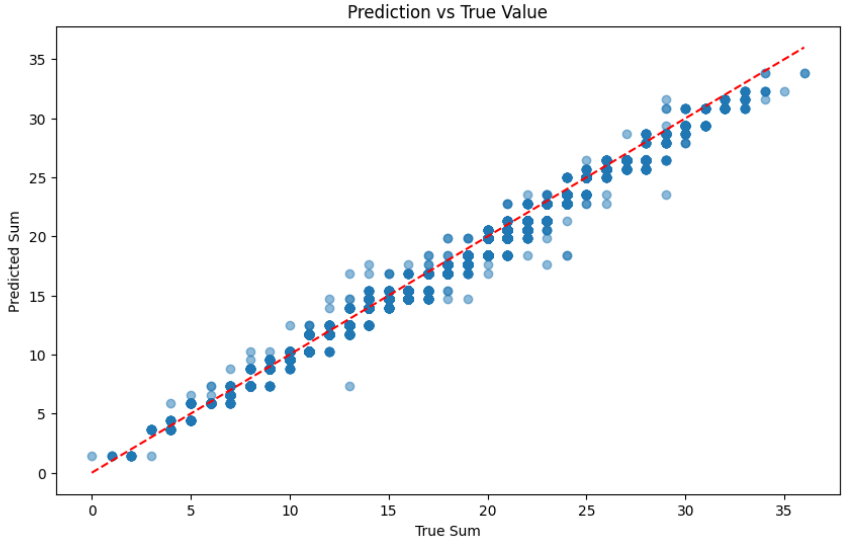


Inference

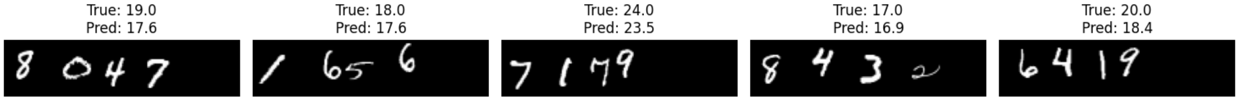
1. **Model Reconstruction:**
    - The model architecture is recreated to match the training configuration.
    - Saved weights and bin edges are loaded to prepare the model for inference.
2. **Prediction:**
    - Images are preprocessed similarly to the training pipeline (normalization and reshaping).
    - Predictions are made on the test data, and the output class indices are mapped back to the continuous scale using the loaded bin edges.
3. **Evaluation Metrics:**
    - Mean Absolute Error (MAE): Measures average prediction error.
    - Mean Squared Error (MSE): Penalizes larger errors more heavily than MAE.
    - Accuracy Within Thresholds: The percentage of predictions falling within ±0.5, ±1.0, and ±2.0 of the true values.
4. **Visualization:**
    - Random samples of predictions are visualized alongside the true values.
    - A scatter plot illustrates the relationship between predicted and true values, with a diagonal reference line for ideal predictions.

Results:

Test Results:
Mean Absolute Error: 0.6393
Mean Squared Error: 0.6901
Accuracy (within ±0.5): 49.12%
Accuracy (within ±1.0): 75.95%
Accuracy (within ±2.0): 98.55%



Demo results:


True: 19.0
Pred: 17.6

True: 18.0
Pred: 17.6

True: 24.0
Pred: 23.5

True: 17.0
Pred: 16.9

True: 20.0
Pred: 18.4