Peeking into the Black Box: Interpreting Digit Classification with SHAP

Mini Research Project (MRP)
MSCS2001-1 Artificial Intelligence

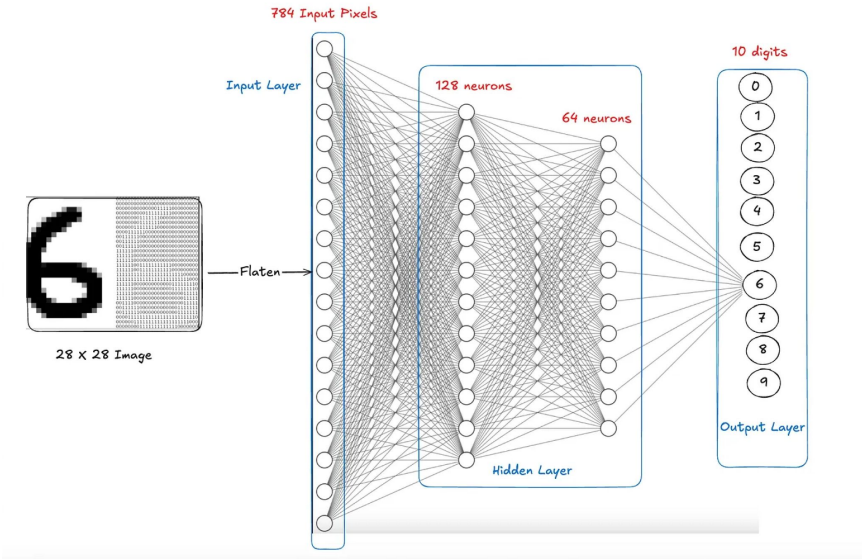
Qingwen Zheng

Motivation & Results

Motivation: Neural networks are often "black boxes." In high-stakes fields (finance, healthcare), trusting a model is as important as its accuracy. We need to know *why* a model made a decision.

Main Idea: Train a Convolutional Neural Network (CNN) to classify handwritten digits (MNIST) and apply **SHAP (SHapley Additive exPlanations)** to visualize feature importance.

Results: The model achieved **98% accuracy**. SHAP visualizations successfully highlighted specific pixel curves that confirmed the model looks at relevant shapes, not background noise.



Literature Review

The Problem: The trade-off between *Accuracy* (Deep Learning) and *Interpretability* (Decision Trees).

Method 1: LIME (Ribeiro et al., 2016): Approximates the complex model locally with a simple linear model to explain individual predictions.

Method 2: SHAP (Lundberg & Lee, 2017): Based on Game Theory. It calculates the contribution of each feature (pixel) to the prediction. It is generally more consistent than LIME.

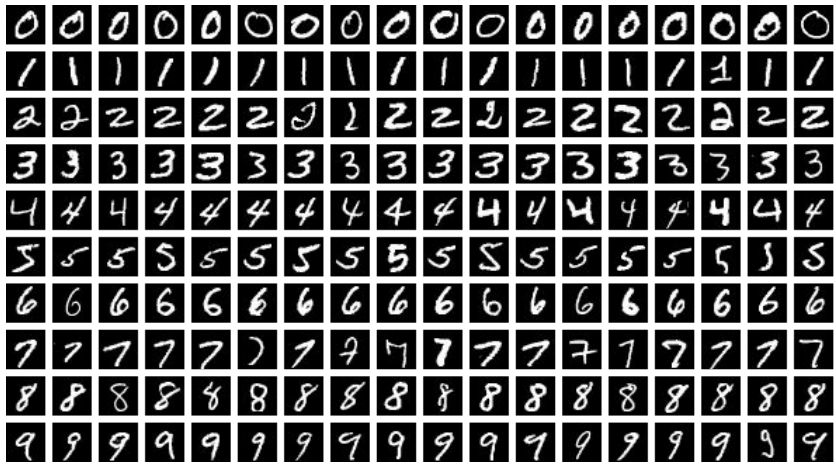
Method 3: Saliency Maps: Visualizes gradients to see which pixels change the output the most (e.g., Grad-CAM).

Approach

Dataset: MNIST (60,000 training images of handwritten digits 0-9).

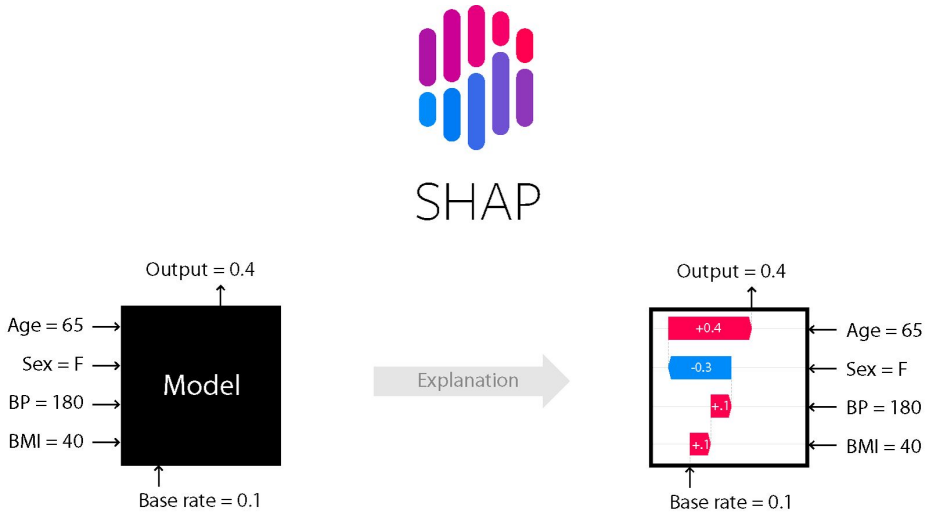
Model Architecture: Input Layer (28x28 grayscale)

- Conv2D Layer (Feature extraction)
- MaxPooling (Downsampling)
- Dense Layer (Classification)



Approach

- **Tools:** TensorFlow/Keras (Model building), SHAP Library (DeepExplainer).
- **Process:**
  1. Train CNN on normalized pixel data.
  2. Select a background distribution (random sample of training data).
  3. Calculate Shapley values for test images.
  4. Plot pixel impact (Red = increases confidence, Blue = decreases confidence).

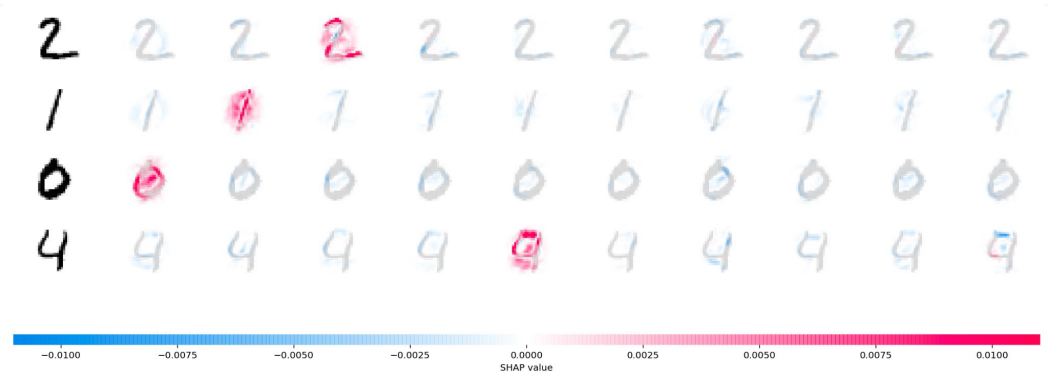


Demo & Results

Visual Analysis: (Paste the image generated by the code in Step 1 here).

- *Caption:* The red pixels show what the model "liked" (e.g., the curve of the '2'). The blue pixels show areas that argued against the prediction.

Demo Video: Link
GitHub Repo: Link



Thank you