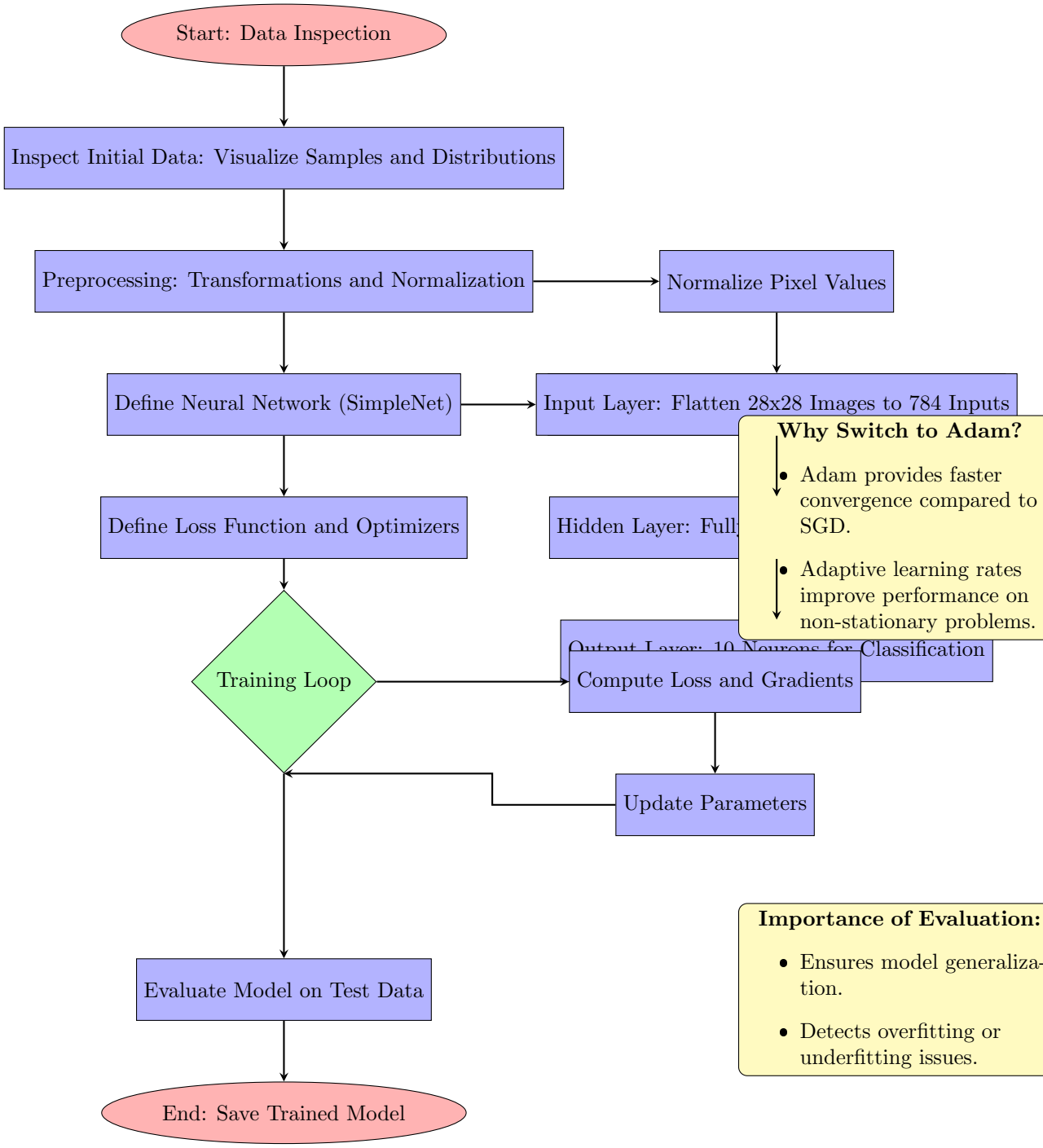
Flowchart: Neural Network Training Process and Optimizations



Explanation of Modifications

Data Inspection and Preprocessing

Before starting the training process, it is crucial to inspect the initial dataset. Visualizing sample images and distributions helps in understanding:

- Data quality: Identifying corrupted or mislabeled samples.
- Distribution: Ensuring balanced class representation.

Transformations applied during preprocessing include:

- **Normalization:** Rescales pixel values from [0, 255] to [0, 1]. This stabilizes gradient calculations during backpropagation.
- **Data Augmentation:** Random rotations and flips artificially expand the dataset, improving generalization by reducing overfitting.

Neural Network Layers

The architecture of the neural network is designed to balance simplicity and performance for the MNIST dataset:

- **Input Layer:** Flattens 28x28 pixel images into a single vector of 784 features, preparing data for dense connections.
- **Hidden Layer:** A fully connected layer with 128 neurons and activation functions to capture complex features from the input data.
- **Output Layer:** Contains 10 neurons (one for each digit class) and applies the Softmax activation function to output a probability distribution.

Switch from SGD to Adam Optimizer

SGD is a reliable optimizer but can struggle with convergence on complex datasets or non-stationary problems. By switching to Adam:

- The training process adapts learning rates per parameter, which accelerates convergence.
- It combines the benefits of momentum and RMSprop, improving stability and efficiency.

Evaluation and Generalization

Regular evaluation during training ensures that the model generalizes well to unseen data. This step is crucial to avoid overfitting, especially in smaller datasets like MNIST.