Student Name: Immanuel Sam

USN: 22MSRDS058

Email: [22msrds058@jainuniversity.ac.in](mailto:22msrds058@jainuniversity.ac.in)

**About the Dataset**

The "SVNH" dataset is a commonly used collection of real-world images of house numbers, typically referred to as the "Street View House Numbers" dataset. This dataset contains images of house numbers captured from Google Street View. Each image in the SVHN dataset typically contains one or more clearly visible house numbers. The dataset is used in machine learning and computer vision tasks, primarily for tasks related to digit recognition, character recognition, and text detection in natural scenes.

It is an important dataset for various research and applications, including optical character recognition (OCR), deep learning, and computer vision, as it represents real-world data commonly encountered in scenarios like street signage, address recognition, and more. Researchers and practitioners often use the SVHN dataset to train and evaluate models for tasks such as multi-digit recognition and text extraction from images.

**Explaining the code**

**Step 1: Load the SVHN Dataset**

**Explanation:**  In this initial step, we load the SVHN dataset, which consists of images of house numbers collected from Google Street View. This dataset serves as the foundation for our classification task.

**Step 2: Split Data into Training and Testing Sets**

**Explanation:** It’s essential to partition the dataset into two subsets: one for training and another for testing. We typically use an 80-20 split, allocating 80% of the data for training and reserving 20% for testing.

**Step 3: Data Visualization and Preprocessing**

**Explanation:** Data visualization and preprocessing are critical. This phase involves closely inspecting the dataset for anomalies and ensuring it's in an optimal format for modeling. Tasks include normalizing pixel values and reshaping images.

**Step 4: Visualize Data**

**Explanation:** Visualization is a crucial step to gain a better understanding of the data. It allows us to view sample images along with their corresponding labels, providing insights into the data's characteristics.

**Step 5: Reshape Images**

**Explanation:** Reshaping images ensures a consistent format for processing in the model. Images are typically reshaped to a common size, facilitating uniformity in data representation.

**Step 6: Normalize Pixel Values**

**Explanation:** Normalizing pixel values involves scaling them to a common range, typically [0, 1]. This standardization aids in the model's training process.

**Step 7: Transform Labels**

**Explanation:** Transforming labels into a format suitable for a neural network often entails one-hot encoding for multi-class classification tasks. This conversion ensures labels are compatible with the model.

**Step 8: Print Total Number of Classes**

**Explanation:** Understanding the total number of classes in the dataset is essential. This information guides the design of the neural network's output layer and the choice of a suitable loss function.

**Step 9: Model Training and Evaluation**

**Explanation:** This phase involves constructing a neural network model and training it on the training data. Model evaluation is performed on the testing data to assess its performance and generalization.

**Step 10: Plot Training Metrics**

**Explanation:** Visualizing training metrics, such as loss and accuracy, across epochs provides valuable insights into the model's learning process during training.

**Step 11: Summarize Model**

**Explanation**: Summarizing the model provides an in-depth overview of its architecture, including the number of parameters in each layer. This summary is useful for understanding the model's structure.

**Step 12: Evaluate Model**

**Explanation:** Model evaluation on the testing data is crucial to assess its real-world performance and generalization. This step quantifies the model's accuracy and effectiveness in its intended task.