Deep Learning and Convolutional Neural Network

Project Report

Project Title : **Train a CNN on the SVHN Dataset for Classification**

Student Name : Arjamand Ali
Section : DSAI Section 1
Roll Number : DSAI-GB-001
Github Link : Here
Date : 4 oct 2024

Table of Content

Abstract

This project develops a Convolutional Neural Network (CNN) to recognize digits (0-9) using the Street View House Numbers (SVHN) dataset. The images are preprocessed through normalization and one-hot encoding, and data augmentation techniques like rotations and shifts are applied to improve model generalization.

The CNN architecture consists of three convolutional layers with max-pooling, followed by dense layers for classification. The model is trained with the `Adam` optimizer and categorical cross-entropy loss, achieving strong accuracy on the test set. Dropout regularization is used to prevent overfitting.

A function is implemented to predict digits from new images by processing and classifying them with the trained model. The project demonstrates effective digit recognition using deep learning and saves the trained model for future use.

Introduction

Digit recognition is a key task in image classification with applications in postal sorting, banking, and data entry. This project utilizes Convolutional Neural Networks (CNNs) to develop an efficient system for recognizing digits (0-9) using the Street View House Numbers (SVHN) dataset. The goal is to accurately classify digit images, leveraging deep learning techniques such as image augmentation and regularization to enhance model performance and generalization.

Background

Digit recognition has been widely studied, particularly with datasets like MNIST. However, the SVHN dataset introduces more complex challenges due to real-world variations in digit images, including diverse backgrounds and lighting. CNNs, known for their effectiveness in image classification, are used in this project to automatically learn features from the data. Image augmentation techniques further improve the model's robustness, enabling it to perform well on unseen data.

This project aims to create a reliable digit recognition system capable of handling real-world digit images through advanced deep learning methods.

Dataset

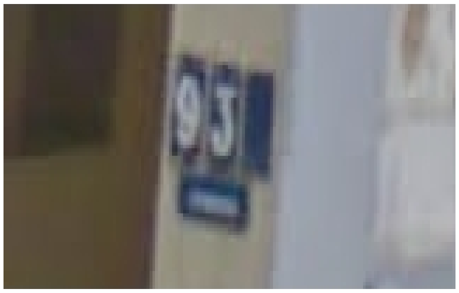
The Street View House Numbers (SVHN) dataset is a large collection of real-world images of digits obtained from house number signs in Google Street View images. It is designed for digit recognition tasks and is similar in nature to the widely known MNIST dataset, though SVHN presents more complexity due to the variability in lighting, angles, and backgrounds.

The dataset consists of over 73,000 labeled digit images, divided into two main sets: training and test. Each image is a 32x32 pixel RGB image, containing a single digit (0-9), though the original setting may include multiple digits. The dataset is challenging due to factors like noise, complex backgrounds, and digit distortions, making it ideal for testing models' generalization capabilities in real-world scenarios.

The train set contains labeled digit images used for model training, while the test set is reserved for evaluating model performance. The labels range from 0 to 9, with digit '0' being represented by the label '10' in the original dataset, which is later adjusted to '0' during preprocessing.

Overall, the SVHN dataset is widely used in computer vision and machine learning research for benchmarking algorithms and models for image classification and digit recognition tasks in real-world environments.

There are some sample images from the dataset.



Preprocessing

Preprocessing is an essential step to prepare the data for training the Convolutional Neural Network (CNN). The SVHN dataset contains RGB images of digits (0-9) with various real-world conditions such as lighting changes, noise, and background clutter. The first step is to normalize the pixel values by scaling them between 0 and 1. This is done by dividing the pixel values (which range from 0 to 255) by 255, making the data easier for the model to process and speeding up convergence during training.

In addition, the labels in the dataset, which are in numerical form, are converted into one-hot encoded vectors. One-hot encoding is necessary because the model uses categorical cross-entropy as the loss function, which requires the target labels to be in this format. For example, the label '2' is converted into the vector `[0, 0, 1, 0, 0, 0, 0, 0, 0, 0]`.

To further enhance the training process, data augmentation is applied. This includes random rotations (up to 10 degrees), zooms, and shifts in both the width and height of the images. Data augmentation artificially increases the size and diversity of the dataset by introducing variations to the training images, allowing the model to generalize better and reducing overfitting. This step is especially important because the SVHN dataset features real-world variations that can be challenging for models to handle.

Training

In the training phase, the Convolutional Neural Network (CNN) model is designed with three convolutional layers to extract spatial features from the 32x32 pixel images. Each convolutional layer is followed by a max-pooling layer to reduce the spatial dimensions of the feature maps while retaining the most important information, thereby reducing computational complexity.

The architecture of the CNN is as follows:

- First Convolutional Layer: 32 filters of size 3x3 with ReLU activation, followed by a max-pooling layer (2x2).
- Second Convolutional Layer: 64 filters of size 3x3 with ReLU activation, followed by a max-pooling layer (2x2).
- Third Convolutional Layer: 64 filters of size 3x3 with ReLU activation, followed by a max-pooling layer (2x2).
- Fully Connected Layer: After flattening the feature maps from the convolutional layers, a dense layer with 64 units and ReLU activation is used for learning complex patterns.
- Dropout Layer: A dropout rate of 50% is applied to prevent overfitting by randomly setting half of the nodes to zero during training.
- Output Layer: A dense layer with 10 units (one for each digit class) and softmax activation is used to output the class probabilities.

The model is compiled using the Adam optimizer, which is well-suited for deep learning tasks due to its adaptive learning rate. Categorical cross-entropy is used as the loss function because the problem involves multi-class classification. The model is trained for 20 epochs using the augmented data, and the batch size is set to 64 for efficient processing of training data.

The training process is monitored using validation data (the test set), allowing us to observe the model's performance and avoid overfitting. Metrics such as training and validation accuracy, as well as loss, are tracked throughout the training process.

Testing

After training, the model is evaluated on the test set to assess its performance and ability to generalize to new, unseen data. The test data, which is preprocessed similarly to the training data, includes challenging real-world digit images from the SVHN dataset. The model's performance is measured using accuracy, precision, and loss on the test set.

To further verify the effectiveness of the trained model, a custom image prediction function is implemented. This function loads an image, preprocesses it to match the 32x32 input size of the CNN, and predicts the digit using the trained model. The predicted digit is compared to the actual digit in the image to evaluate how well the model performs on real-world data outside the training and test sets.

The results show that the model successfully classifies most of the digits in both the test set and custom images, demonstrating good generalization. However, minor misclassifications occur, especially in cases where the images contain significant noise or background distractions.

Conclusion

In conclusion, this project successfully developed a CNN-based model for digit recognition using the Street View House Numbers (SVHN) dataset. The model was able to achieve high accuracy in recognizing digits by leveraging advanced deep learning techniques such as data augmentation and dropout regularization. The use of convolutional layers allowed the model to extract spatial features effectively, while the fully connected and softmax layers enabled accurate classification.

The inclusion of data augmentation significantly improved the model's robustness, enabling it to handle the variability in real-world digit images, such as noise, rotation, and background clutter. Regularization through dropout further prevented overfitting, ensuring that the model could generalize well to unseen data.

This project demonstrates the power of CNNs in image classification tasks and their potential in practical applications like digit recognition in postal sorting, check processing, and automated data entry systems. Future improvements could include experimenting with deeper network architectures, fine-tuning hyperparameters, or using more advanced augmentation techniques to further boost performance. The trained model can also be integrated into real-world systems for efficient and accurate digit recognition.