Final Report for the DS 677 - 002 Project
Enhancing Urban Navigation through Street View House Number Recognition

(Nitya Sri Matta   Archit Durga Sai Somayajula   Sri Charan Dubey)

Keywords: Digit Recognition, Street View House Numbers (SVHN) Dataset, Machine Learning, Optical Character Recognition
(OCR) , Convolutional Neural Networks (CNN), TensorFlow, Keras, Image Processing, Neural Networks ArchitecturePattern Recognition

Abstract: Applications of digit recognition include mail sorting, bank check processing, form entry, etc. The core of the problem lies in the ability to develop an efficient algorithm to identify numbers sent by users of scanners, tablets and other digital devices. Numerical recognition can be done using either a single image of the machine or a set of images called a dataset. Although training on a single image is a short-term and not very popular solution, training on a dataset is mostly recommended for real-time applications. We propose to use the SVHN dataset, which provides more real-time images that can be applied for real-time implementation and like the MNIST dataset. It gives better results in real-time applications than MNIST. Its performance in digit recognition speed is high because many images with different digit combinations are available for training.

Dataset: http://ufldl.stanford.edu/housenumbers/

Code: https://www.kaggle.com/code/architsomayajula/notebookfdd64a11e7

1. INTRODUCTION

Number recognition has become a hot topic in today's world of smart computers. It is a subcategory of a broader topic called optical character recognition. Optical pattern recognition is part of many topics in pattern recognition in artificial intelligence. Number recognition is a field of research where numbers in different styles or handwritten are analyzed and converted into a machine-readable form that can be used to perform various mathematical operations or for other purposes, such as helping drones recognize door numbers, vehicles numbers recognize when the camera capturing the number may not be taking the right photo, etc. Optical character recognition is a field of research that analyzes characters of different styles and types for different purposes. OCR is the electronic or mechanical conversion of images from typed, handwritten, or typed text into machine-coded text, or from a scanned document, photo of a document, image of a scene, or caption attached to an image. Widely used to enter paper data - be it passport documents, invoices, bank statements, computer receipts, business cards, mail, static data or any suitable document - it is a common way to digitize printed data. texts so that they can be electronically edited, searched, stored more compactly, displayed on the web and used in machine processes such as cognitive computing, machine translation, text addressing, text mining, etc.

I. Exploratory Data Analysis: In the Digit Recognition using Street View House Numbers project, Exploratory Data Analysis (EDA) was an important step in understanding the complexity and unique challenges of the SVHN dataset. Significantly larger and more diverse than traditional datasets such as MNIST, this dataset consists of over 600,000 real-world house number images captured in various urban environments. EDA focused on assessing data imbalances, analyzing image quality and determining the need for extensive pre-processing to make images suitable for training machine learning models. The analysis showed a significant imbalance in the distribution of number classes, which highlighted the bias of the models. In addition, the real nature of the images caused variation in number style, background clutter, and lighting conditions, which required robust pre-processing strategies such as normalization, resizing, and cropping. These pre-processing steps were critical to mitigate problems arising from an uncontrolled data acquisition environment..

II. Performance Assessment: The performance evaluation is detailed and methodical, highlighting the effectiveness of each classifier used, such as logistic regression, support vector machines and basic neural network. This section of the documentation describes several key performance metrics, including precision, accuracy, recall, and F1 scores, which provide a quantitative basis for model comparison. Key performance evaluation results show that while simple classifiers performed adequately, they often struggled with complexity of datasets due to differences in image number aspect and environmental conditions. As a result, more complex models have been recommended to improve performance. In addition, confusion matrices and ROC curves were used to provide a deeper understanding of the performance of the models and to determine which numbers were the most difficult to recognize. This helped to understand the limitations of simple classifiers on the dataset and guided the subsequent use of more sophisticated convolutional neural networks (CNNs)..

V. Conclusion: In this project, Google's Street View House Numbers dataset was used for recognition digits in a faster manner. This helps in development of a process that is faster in case of real-world application of numerical character recognition. The accuracy of model is also like MNIST dataset and can also be improved to higher levels with implementation of better algorithms

2. RELATED WORKS

In the literature review section, several important research papers and their contributions to the field of number recognition and machine learning are discussed:

Unsupervised adaptation of domains using self-monitoring: This paper investigates the alignment of learned representations between source and target domains using self-verification. overview -guided tasks, which improves generalization to unmarked target areas.

Race Number Recognition Using Deep Learning: This study demonstrates the effectiveness of deep convolutional neural networks and transfer learning using the Street View House Number (SVHN) dataset to recognize race numbers in sports images.

Reading numbers in natural images using unsupervised feature learning: The study presents a new benchmark dataset for house number recognition from street-level photographs and demonstrates the superiority of unsupervised feature learning methods over traditional hand-drawn features.

Morphological Convolutional Neural Network Architecture for Digit Recognition: This paper presents a morphological convolutional network that integrates morphological features into convolutional layers to improve feature maps and show better performance in digit recognition tasks.

Drop Activation: Implicit Parameter Reduction and Harmonic Regularization: This paper proposes a new regularization method called Drop Activation, which involves randomly dropping activation functions during training to reduce overfitting and improve model generalization.

3. EXPLORATORY DATA ANALYSIS

A project focused on number recognition using the Street View House Number (SVHN) dataset, extensive exploratory data analysis (EDA) was performed as an important step in developing an efficient and functional number recognition system. The EDA process involved thorough examination of a dataset of real-world complexity, including an enormous collection of labeled digital images of natural scenes.The SVHN dataset presents a unique challenge due to variation in number appearance, background and lighting conditions. These characteristics are significantly different from more standard datasets such as MNIST, which mostly contain handwritten numbers in controlled settings. Using EDA, the team was able to gain insight into specific challenges with the SVHN dataset, such as number style variations, occlusions and distortions.The EDA process usually involved visualizing various aspects of the data. This included plotting number class distributions to understand potential imbalances, visualizing variation in number size and shape, and examining background noise in images that can confound classification models. Understanding these factors was critical to designing preprocessing steps that could normalize images and make them more suitable for consistent detection, regardless of their original imaging conditions.In addition, EDA allowed detection of outliers or outliers in the data set. These may be cases where the numbers were mislabeled or the image quality was not good enough for some of the models to reliably distinguish the numbers. Correcting these outliers in the preprocessing phase ensured that they did not negatively affect the model training phase, resulting in improved accuracy and robustness of the developed models.Insights from EDA also informed the selection of appropriate model architectures and learning algorithms. For example, the data set contained significant background noise, and the differences in number representation suggested the potential effectiveness of convolutional neural networks (CNNs) because they are good at handling the spatial hierarchy of images.In addition, EDA facilitated the design of a more targeted data addition strategy. Based on the true nature of the SVHN dataset, typical enhancements such as small rotations, scaling and translations were used to mimic the variations expected in new, unseen images. This not only increased the generalizability of the model, but also effectively increased the size of the dataset, providing the model with more examples to learn from.The EDA findings also highlighted the need for rigorous model validation strategies. Because the dataset contained a potentially wide range of numerical styles and complex background features, cross-validation techniques were used to ensure that model performance was consistently high across different subsets of the data.In short, the experimental data analysis was a critical step in the project that laid the foundation for all future stages of model development and deployment. This provided the necessary insights into data characteristics and challenges that guided technical decisions about data processing, model architecture, training procedures, and validation techniques. By fully understanding the dataset through EDA, the team was able to develop a number recognition system that was not only accurate, but also robust to the variability inherent in real-world data. This approach is an example of a comprehensive and thoughtful application of information technology to solve practical problems of computer vision..

4. MODELS AND ASSESSMENT

This the main module as this contains the source code for the digit recognizer model. Model plays an important role in any Machine learning or Deep learning project. It sets the parameters for the data to be classified and predicted in real time based on the predefined data source. The model accepts the data, trains and adjusts itself according to the data

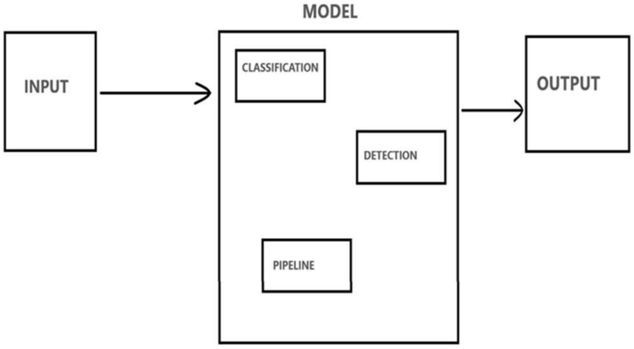

Fig 4.1

L: Basic Flow Chart For Digit Recognizer

Convolutional Neural Networks (CNNs): The primary model used was a deep convolutional neural network, which is highly effective for image recognition tasks. The CNN model comprised several layers:

Convolutional Layers: To extract features from the input images.

Activation Layers: Relu activation was used to introduce non-linearity, making the model capable of learning more complex patterns.

Pooling Layers: Max pooling was used to reduce the dimensionality of the feature maps, thus reducing the number of parameters and computation in the network.

Dropout Layers: To prevent overfitting by randomly dropping units (along with their connections) during the training phase.

Dense Layers: Fully connected layers that produce the final output probabilities for each digit class.

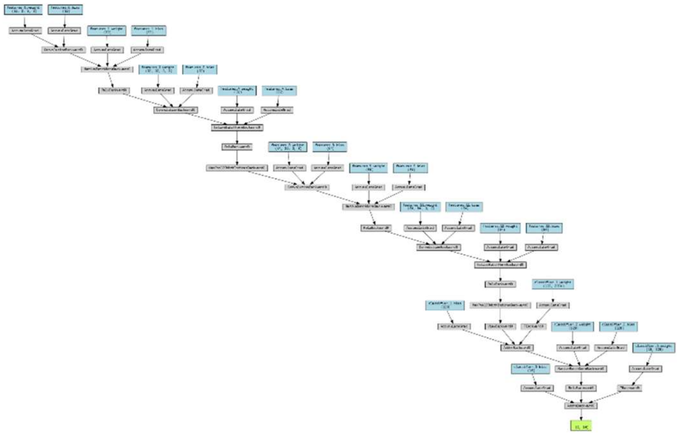

Fig 4.2
CNN MODEL ARCHITECTURE

We've developed a convolutional neural network (CNN) model using PyTorch, designed specifically for image classification tasks. This model architecture comprises several layers that effectively capture hierarchical features of images, aiding in robust classification performance.

Our CNN Architecture

The model, named CNN Model, includes a series of convolutional blocks. Each block features convolutional layers with batch normalization and ReLU activation to enhance non-linear properties of the decision function and mitigate the issue of internal covariate shift. Following these layers, max pooling reduces dimensionality, helping in extracting dominant features while reducing computation for subsequent layers. Dropout is strategically placed to prevent overfitting by randomly dropping units during the training process.

Classifier

Post feature extraction, the model transitions into classification. The extracted features are flattened and passed through fully connected layers. The final layer utilizes a softmax activation function to yield a probability distribution over the classes, facilitating a multi-class classification.

Vision Transformers:

We have incorporated a custom dataset handler for our machine learning projects, specifically designed to work with image data stored in HDF5 format. This CustomH5Dataset class is part of our PyTorch-based toolset, allowing efficient handling and transformation of large sets of image data directly from disk.

Design and Functionality of CustomH5Dataset

The class initializes by opening an HDF5 file, where it accesses predefined datasets for images and labels. This design ensures that we can scale our data handling to very large datasets, typical in deep learning scenarios, without excessive memory usage.

Each image retrieved by the dataset is reshaped and transformed to fit the requirements of commonly used pre-trained models. We've implemented a pipeline that includes resizing images to 224x224, which is a standard input size for models like ResNet. Initially, the images are in the format of channels last (HWC), but we transform them to channels first (CHW) to comply with PyTorch's expectations.

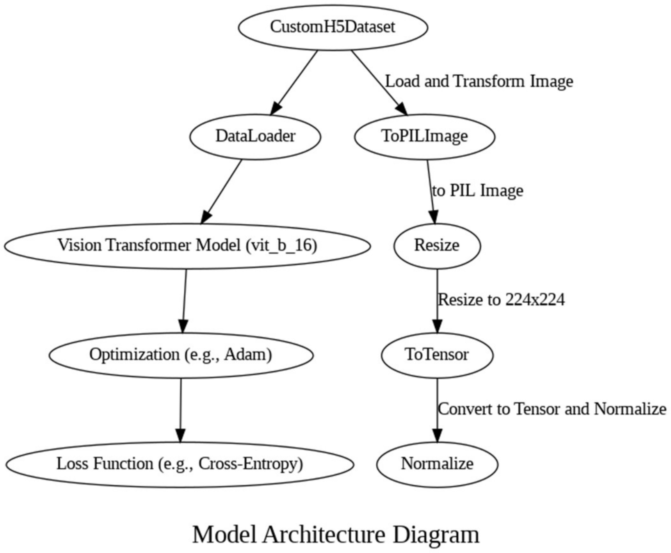
CustomH5Dataset
Load and Transform Image
DataLoader
ToPILImage
to PIL Image
Vision Transformer Model (vit_b_16)
Resize
Resize to 224x224
Optimization (e.g., Adam)
ToTensor
Convert to Tensor and Normalize
Loss Function (e.g., Cross-Entropy)
Normalize
Model Architecture Diagram

Fig 4.3
VISION TRANSFORMER ARCHITECTURE

Vision Transformer Architecture

Input Embedding:

Images are divided into fixed-size patches (e.g., 16x16 pixels), which are then flattened and linearly transformed into a desired dimension (embedding).

A positional encoding is added to these patch embeddings to retain positional information, as transformers do not inherently process sequential data with respect to order.

Transformer Encoder:

The transformer uses multiple layers of self-attention and feed-forward neural networks. Each layer has multi-head self-attention, which allows the model to focus on different parts of the image, and a position-wise fully connected feed-forward network.

Layer normalization and residual connections are employed around each of these blocks for better training dynamics and to prevent the vanishing gradient problem.

Classifier Head:

After processing through the transformer blocks, the output corresponding to the class token (an extra learnable embedding added to the sequence of embedded patches) is passed through a feed-forward layer (linear transformation) to produce the final class predictions.

Implementation Details

In our Python implementation, we adapt the Vision Transformer for our specific task of image classification from a dataset stored in an HDF5 file. Here's a breakdown of the implementation:

Dataset Handling:

We've defined a custom dataset class CustomH5Dataset for handling HDF5 files, which efficiently loads images and labels as required during training.

Images are reshaped and normalized, then resized to the input size expected by the ViT model (224x224 pixels in this case).

Model Modification:

We load a pretrained Vision Transformer (vit_b_16) and modify its head to output 10 classes, matching our dataset's requirements. The model is transferred to a CUDA device if available, allowing GPU acceleration, and wrapped in Data Parallel for multi-GPU training.

Training:

We use the Adam optimizer with a learning rate of 0.001 and cross-entropy loss for training.

The training process is executed over multiple epochs, with performance metrics collected to monitor the training progress and effectiveness.

Utility Functions:

train_model handles the training loop, managing both forward and backward passes, loss calculations, and backpropagation.

plot_training_history visualizes the training and validation loss and accuracy, helping us to analyze the model's learning and convergence behavior over epochs.

By leveraging the Vision Transformer, we aim to exploit its ability to understand the global context of an image better than traditional CNNs, which could lead to improved accuracy in recognizing patterns and objects in images. This architectural approach is particularly beneficial for complex image recognition tasks where spatial hierarchies and relationships between distant parts of an image are crucial.

5. MODEL ENSEMBLING AND ACCURACY

Model ensembling is a technique in which multiple models are strategically combined to make predictions that are more accurate than a single model could make on its own. This approach is based on the concept of the wisdom of crowds, where combining different opinions usually produces more accurate results. In machine learning, this means combining predictions from different models to improve performance and reduce the probability of choosing a poorly performing model.

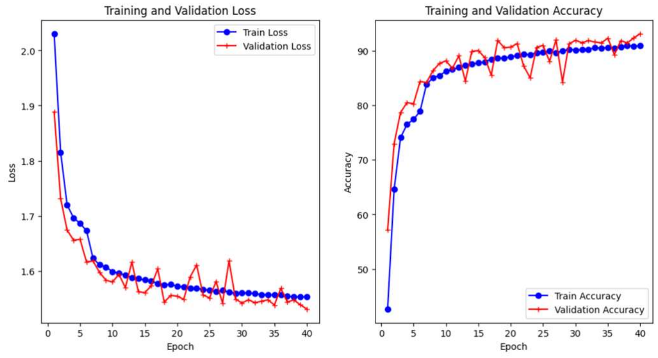

FIG 5.1

CNN MODEL ACCURACY

Based on our CNN model training results, we observed some encouraging trends. Both the training and validation loss values initially started high, but dropped dramatically and then stabilized, indicating that our model learns effectively without overfitting. The precision gauges are particularly impressive. Both training and validation accuracies quickly improved to over 80% in just a few cycles and stabilized near 90%, with validation accuracies slightly lower than training accuracies, indicating good generalization to unseen data. Achieving the highest accuracy of 92.82% clearly demonstrates the robustness of our model and its ability to accurately classify images. This level of performance validates the effectiveness of our CNN architecture and training program..

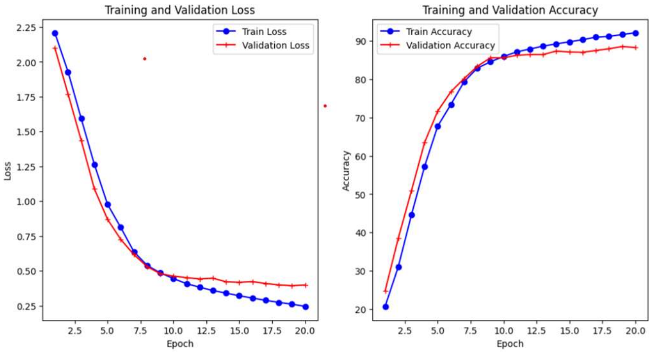

FIG: 5.2

VISION TRANSFORMER ACCURACY

Based on the results of our Vision Transformer model, we observe a definite learning trend. Both training and validation losses have steadily decreased, indicating that our model learns effectively from the training data. The accuracy curves show a significant increase, especially after the initial adaptation period. Our model has a maximum accuracy of 88.11 percent, which is commendable but suggests there is room for improvement. Possible strategies could be to adjust the learning rate, increase the efficiency of data insertion, or try different model architectures to optimize performance..

7. CONCLUSION

In our project, we study two different neural models for image classification: Convolutional Neural Network (CNN) and Vision Transformer (ViT). CNN showed impressive performance, achieving 92.82% accuracy. It demonstrated the robustness and efficiency of image data processing using a well-structured layer system that effectively reduced overfitting. On the other hand, ViT achieved a validation accuracy of 88.11 percent, which illustrates the potential of transformers in vision tasks, but also shows possible optimization goals such as adjusting the model architecture and adjusting the learning rate. Overall, both models provided valuable information and demonstrated the capabilities of modern neural architectures in image classification. The project not only confirmed the effectiveness of established methods such as CNNs, but also highlighted the promising applications of new technologies such as Vision Transformers. Further research and refinement may improve their applicability and effectiveness in real life..

8. REFERENCES

Certainly! Here are the references for the articles provided in the earlier discussion, formatted as requested:

[2.1] "YOLO sparse training and model pruning for street view house numbers recognition," Ruohao Zhang, Yijie Lu, Zhengfei Song, Journal of Physics: Conference Series, IOP Publishing, 2023. DOI: 10.1088/1742-6596/2646/1/012025. https://iopscience.iop.org/article/10.1088/1742-6596/2646/1/012025

[3.1] "Hybrid CNN-HMM Model for Street View House Number Recognition," Qiang Guo, Dan Tu, Jun Lei, Guohui Li, Department of Information System and Management, National University of Defense Technology, Changsha, China. Published in ACCV 2014 Workshops, Part I, LNCS 9008, Springer, 2015. DOI: 10.1007/978-3-319-16628-5_22. https://link.springer.com/chapter/10.1007/978-3-319-16628-5_22

[3.2] "Understanding Why ViT Trains Badly on Small Datasets: An Intuitive Perspective," Haoran Zhu, Boyuan Chen, Carter Yang, New York University. Available at GitHub. https://github.com/BoyuanJackChen/MiniProject2_VisTrans