Real-Time Digit Classification System Using SVHN Dataset with Deep Learning

Report

Dhowfeek Hasan S

210171601013

B.TECH AI & DS

1. Introduction

This report presents a detailed study of a Street View House Numbers (SVHN) Classification System developed to recognize and classify in real-time using images from the SVHN dataset. The system employs deep learning techniques implemented in TensorFlow and Keras, and integrates with Streamlit to display predictions and provide real-time feedback.

2. Dataset

2.1 Data Collection

The SVHN dataset is a real-world image dataset containing cropped digit images obtained from Google Street View. Each image represents a single digit (0–9), and the dataset provides a challenging classification task with images in various lighting conditions, orientations, and backgrounds.

2.2 Data Preparation

- **Resizing**: Each image is resized to 32x32 pixels to standardize input size for the model.
- **Normalization**: Images are normalized by scaling pixel values to a [0, 1] range, which helps to improve model performance by facilitating faster and more efficient training.

2.3 Data Splitting

The dataset is split into training, validation, and testing subsets, ensuring the model has sufficient data to learn, validate, and evaluate its performance.

3. Libraries and Tools Used

3.1 Libraries

- OpenCV: Used for image processing, primarily for data preparation and display functionalities.
- TensorFlow & Keras: Employed for building and training the Convolutional Neural Network (CNN) model used for digit classification.
- NumPy: Utilized for handling numerical operations and data transformations.
- Streamlit: Integrates an interactive interface, allowing users to see predictions in real-time.

3.2 Helper Functions

- Custom helper functions, such as draw_prediction_overlay, are used to display model predictions on-screen in real-time.

4. Model Architecture

The CNN model used in this project is implemented in train_and_evaluate.py and follows the architecture below:

1. Conv2D (32 filters): Extracts low-level features from the images.
2. MaxPooling2D: Reduces spatial dimensions, retaining essential features.
3. Conv2D (64 filters): Captures more complex patterns associated with digit shapes
4. MaxPooling2D: Further down-sampling to reduce dimensionality.
5. Flatten: Converts the 2D feature map into a 1D vector.
6. Dense (128 units): Fully connected layer with ReLU activation to learn complex representations.
7. Dropout (0.5): Reduces overfitting by randomly dropping 50% of neurons during training.
8. Dense (10 units with softmax): Output layer for classification across 10 classes (digits 0–9).

5. Training and Evaluation Pipeline

5.1 Data Loading and Preprocessing: The dataset is loaded, resized, and normalized for model training.

5.2 Model Training:

- Epochs: 5
- Batch Size: 64
- The model is trained on 80% of the dataset, with 20% reserved for validation.

5.3 Evaluation Metrics:

- Accuracy: Measures the proportion of correct predictions.
- Precision: Indicates the accuracy of the positive predictions for each digit class.
- Recall: Measures the model's ability to correctly identify all true positive cases.
- F1 Score: Harmonic mean of precision and recall, used to evaluate overall performance.

The trained model (svhn_digit_classification_model.h5) demonstrated high accuracy and balanced precision, recall, and F1 score on the test set, indicating strong performance across all classes.

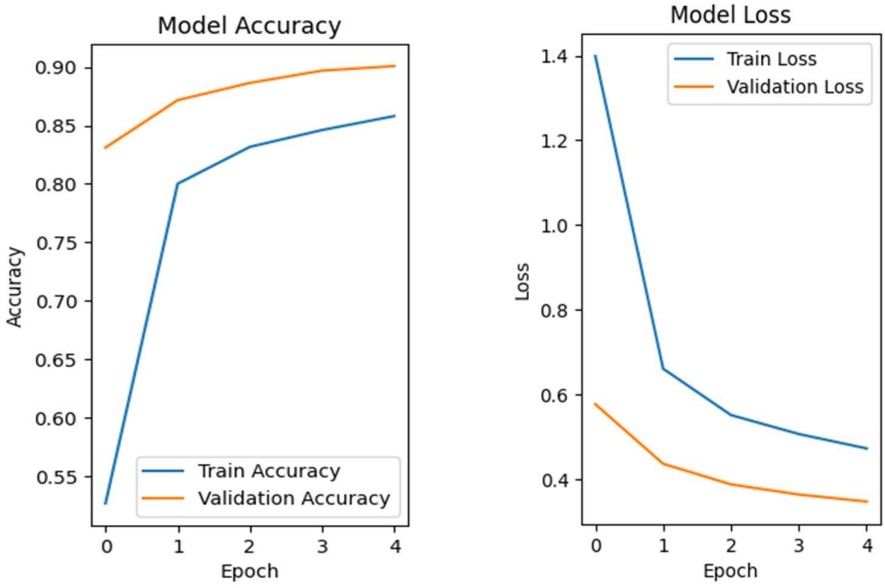
6. Results and Observations

6.1 Performance metrics:

- Accuracy: High classification accuracy on test data.
- Precision and Recall: Balanced performance across classes, minimizing false positives and negatives.
- F1 Score: High F1 score validates the model's robustness for real-time applications, showing good generalization on unseen data**.**

6.2 Visualization of Training Progress

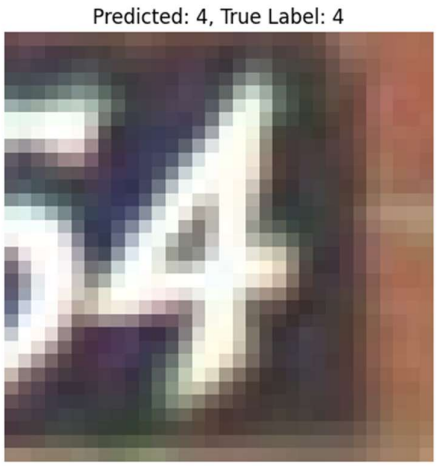
- **Accuracy Plot**: Shows the trend of training and validation accuracies over epochs, demonstrating that the model learns effectively with increasing accuracy on both datasets.
- **Loss Plot**: Illustrates the decline in loss for both training and validation datasets, confirming the model's ability to reduce error consistently.

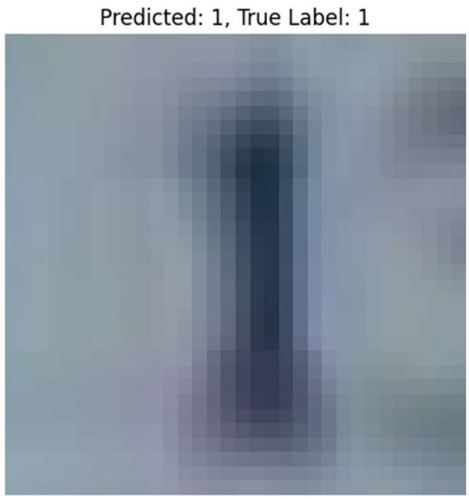


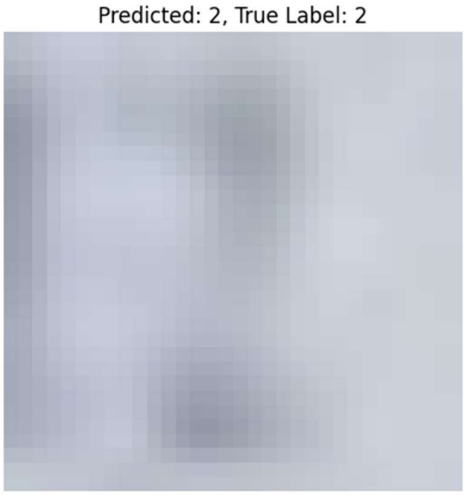
7. Model Prediction

The model was tested on a single sample from the test set to verify its classification capability. The model outputs the predicted label, which is then displayed alongside the actual image to evaluate the prediction accuracy qualitatively.

8. Output







9. Conclusion

The SVHN Classification System efficiently detects and classifies digits in real-time, providing accurate and immediate feedback. This model is suitable for applications requiring digit recognition in dynamic environments.

10. Future Work and Enhancements

Potential improvements to enhance model accuracy and generalization include:

- **Expanded Dataset**: Enhance the dataset to include images under various lighting conditions and angles for improved robustness.
- **Model Optimization**: Optimize the model architecture for faster inference, suitable for low-power devices.
- **Additional Features**: Add functionalities like error logging, confidence thresholds, and user analytics to track model performance over time.