PROJECT PROPOSAL: TRAFFIC SIGN RECOGNITION THROUGH DEEP LEARNING

Salwa Waseem
Student# 10090214161
salwa.waseem@mail.utoronto.ca

Maya Ramaneetharan Ramanathan
Student# 1008717596
maya.ramanathan@mail.utoronto.ca

Maryah Noorani
Student# 1008343188
maryah.noorani@mail.utoronto.ca

Seoyeon (Sally) Kim
Student# 1007713949
sysally.kim@@mail.utoronto.ca

ABSTRACT

This project proposes an advanced Traffic Sign Recognition (TSR) system which aims to increase road safety by accurately identifying traffic signs under various driving conditions. Utilizing Convolutional Neural Networks (CNN), this project aims to improve recognition accuracy. Using multiple datasets, including the German Traffic Sign Recognition Benchmark (GTSRD) and images collected from urban environments in Toronto, the project tackles challenges such as varying lighting, weather conditions, and potential biases. The network involves data preprocessing, augmentation, and a CNN architecture designed to maximize generalization across multiple scenarios. Additionally, ethical considerations surrounding bias and privacy have been addressed. This project aims to eventually contribute to the development of reliable driver-assistance systems, ultimately reducing human error and mitigating road accidents.

—-Total Pages: 7

1 INTRODUCTION

Traffic Sign Recognition (TSR) is a crucial asset in driver-assistance systems. They are utilised to improve road safety by automating detection of traffic signs. This is essential to ensure vehicles are responsive to speed warnings, road conditions and are ultimately adaptable to dynamic driving environments. This technology aims to improve safety in vehicles to reduce human error and eventually mitigate road accidents. However, there are several challenges due to assorted lighting conditions, weather, and overall accuracy of interpretation.

This project intends to generalize TSR across complex driving conditions while retaining recognition accuracy and computational efficiency. To achieve this, TSR models utilize deep learning models and algorithms to enhance classification performance. Deep learning is leveraged in this study as it is highly adaptable feature learning and is incredibly adept at scaling to large datasets. Ultimately, this deep learning model can be trained to effectively and accurately identify traffic signs, thus creating a safer driving environment.

2 BACKGROUND AND RELATED WORK

In a study conducted in 2018 by Madani & Yusof, a TSR model was created utilizing Support Vector Machines (SVM). This method highlights the colour, shape and pictogram on the traffic sign and ultimately efficiently identifies traffic signs with a recognition rate of 98.23%.

Similarly, Soni et al. (2019) combined Histograms of Oriented Gradients (HOG), Local Binary Patterns (LBP), Principal Traffic Component Analysis (PCA) and SVM for traffic sign classification in their model. This model used PCA to reduce the dimensionality to overall improve the computational efficiency of the model, hence gaining an accuracy level of 84.44%.

Namyang & Phimoltares (2020) applied a combination of HOG, SVM and Colour Layout Descriptors (CLD) to classify traffic signs. This method specifically resizes the images of traffic signs to 120x80 pixels then employs an SVM with a Radial Basis Function (RBF) kernel to improve classification accuracy, thus gaining an accuracy level of 93.98%.

Kerim & Efe (2021) employed 9 different Artificial Neural Networks (ANNs) to analyze traffic signs using specific attributes of the image. HOG and LBP were applied to extract specific features to achieve an accuracy of 95%. This model significantly outperforms models that rely exclusively on HOG.

Lastly, Li et al. (2022) utilised the German Traffic Sign Recognition Benchmark (GTSRD) to exhibit the gravity of detail granularity and dimensionality reduction. This approach utilises an improved colour-histogram with HOG and reduced the dimensionality using PCA which yielded a near perfect score of 99.99%. This model highlights the power of using colour information and dimensionality reduction to improve classification outcomes.

3 Data Processing

3.1 Data Sources

GTSRD

As used in the study conducted by Li et al. (2022), this dataset is commonly utilized for traffic sign recognition tasks as it contains a variety of traffic sign images from different categories (Stallkamp et al., 2011).

Data Collected from Toronto

According to a study conducted by SUG in 2018, increasing the diversity of data and sample size yields more accurate results for machine learning algorithms. Therefore, we will be collecting various photographs of traffic signs in downtown Toronto.

Data Collection Process

- Compile a list of common and universal traffic signs, ensuring overlap between GTSRD and signs in Toronto.
- Collect at least 200 images of diverse traffic signs from various angles, distances, and under different lighting conditions to improve model generalization.

3.2 Data Cleaning and Standardization

To ensure good data quality, the dataset will be cleaned by removing duplicates and low-quality samples that could negatively affect the models performance. The images will then be standardized to a common dimension (e.g., 32x32 pixels) and a consistent image format (e.g., JPEG) for more efficient processing in the models architecture (AI, 2022).

3.3 Data Augmentation

To enhance the model's ability to generalize, data augmentation techniques will be implemented (DataCamp, 2022). This will be done by applying various transformations, including:

- Rotations
- Changes in color and brightness
- Distance variation

Each image will be normalized by scaling the pixel values to a range of [0,1] to promote faster convergence during the training process (DataCamp, 2024).

3.4 Data Splitting

After cleaning and augmenting the dataset, we will randomize it and split it into three distinct sets (Baheti, 2021):

- **Training set (70%)**: Used to train the model.
- **Validation set (15%)**: Used for tuning parameters and evaluating performance during training.
- **Test set (15%)**: Reserved for final evaluation after training.

This approach will ensure that our traffic sign recognition model is well-prepared for successful performance in real-world scenarios.

4 Architecture

The core architecture for this project will be based on Convolutional Neural Networks (CNN) due to its superior ability to identify and classify images.

The main architecture begins with an input layer that resizes the traffic sign images given to the model to 64x64 pixels with 3 color channels (RGB). These input images are then normalized before being fed into the network where they are passed through several convolutional layers that utilize the ReLU activation function. The main purpose of these convolutional layers is to project the images onto different dimensions to find common correlations.

In between these convolutional layers, our model will also utilize multiple max pooling layers to identify the most important features of the projected images and discard the rest, saving CPU space and time. This neural network will also include dropout layers to reduce overfitting and maximize accuracy. The processed data is then sent to a flattening layer that remaps the data from a higher to a lower dimension.

The images are then sent to a fully connected layer that performs high-level reasoning and decision making. Finally, the data goes through an output layer where the softmax activation function is used and a separate set of weights and biases are applied to reach the final output data.

5 Baseline Model

The baseline model that will be employed for this is a machine learning algorithm specifically utilized for complex classifications called Support Vector Machines (SVM). SVM is applicable to both non-linear and linear datasets. This method identifies the optimal boundary called a hyperplane which separates datasets into various classes. The aim of this algorithm is to maximize the distance between two data points close to one another, referred to as support vectors, to define the optimal boundary. Furthermore, SVM can be combined with other enhancing techniques such as Histogram of Oriented Gradients (HOG) to achieve a higher accuracy.

6 Illustration

The diagram below illustrates our proposed machine learning model.

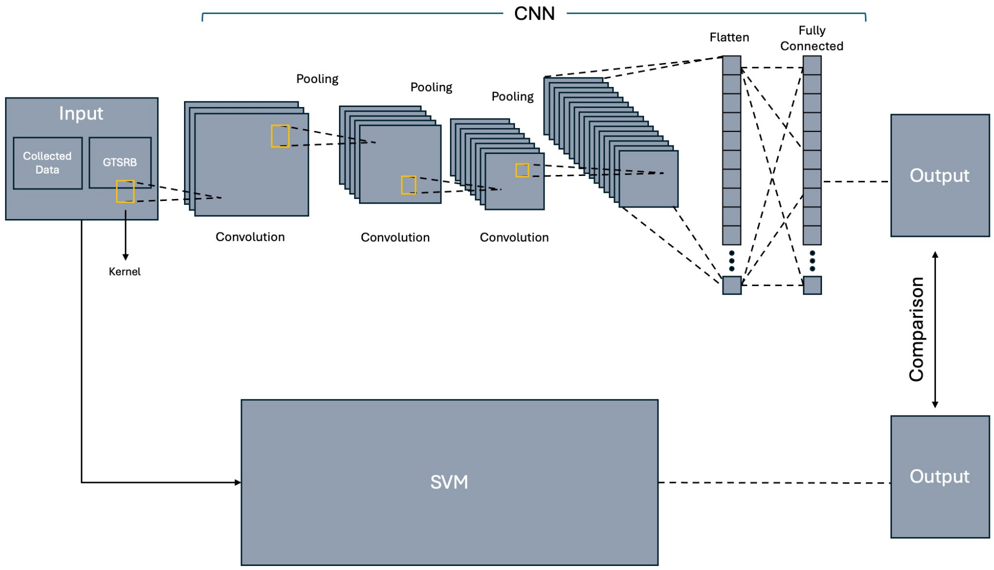

Figure 1: Diagram of proposed model.

7 Ethical Considerations

Bias

- Geographical and cultural bias: Traffic signs vary across different regions, countries and continents. Thus, it is important to choose a particular region to base the model on, and feed data exclusive to that region to increase the accuracy of the model and avoid confusion.
- Socioeconomic bias: The training data must contain a mix of signs from both higher and lower income areas to eliminate bias and ensure that the model can identify different signs regardless of socioeconomic status.

Privacy Concerns

- Data collection: Collection of data must be done carefully to ensure that no faces or license plates are accidentally captured and used. This is done to ensure that there is no breach in privacy.

Accuracy and Safety

- Testing in diverse environments: The testing data must include various different urban and rural environments to ensure that the model has seen and is able to classify signs with satisfactory accuracy regardless of the surrounding environments.

8 PROJECT PLAN

Task	Maryah	Sally	Maya	Member 4	Deadline
Phase 1: Project Setup & Data Collection					
Data collection & preprocessing	Download dataset, perform cleaning & augmentation	Review dataset, assist with data splitting (train/val/test)	Compile list of universal traffic signs	Help with augmentation strategies	Oct 13
Toronto Data Collection	Collect 50 photos of various traffic signs	Collect 50 photos of various traffic signs	Collect 50 photos of various traffic signs	Collect 50 photos of various traffic signs	Oct 13
Phase 2: Baseline Model & Initial Training					
Build the baseline model	Assist with model setup	Implement baseline CNN	Assist with training baseline model	Help with baseline evaluation setup (metrics, confusion matrix)	Oct 20
Evaluate the baseline model	Help analyze dataset weaknesses (e.g. underfitting)	Analyze model performance, document findings	Help measure baseline performance (accuracy, confusion)	Refine data (augmentation, cleaning)	Oct 25
Phase 3: Model Optimization & Hyperparameter Tuning					
Hyperparameter tuning	Assist with tuning (learning rate, batch size)	Focus on tuning regularization & dropout rate	Focus on tuning model complexity (layer depth, optimizers)	Assist with experimentation (e.g., batch size, tuning)	Nov 10
Implement strategies to handle overfitting	Apply data augmentation (rotations, flips, zooms)	Add dropout layers to the architecture, tune dropout rate	Apply L2 regularization (weight decay) to control overfitting	Implement early stopping for the training process	Nov 13
Improve model architecture	Support advanced model experimentation	Test different architectures (residuals, deeper CNNs)	Implement advanced architectures or transfer learning	Assist with architecture analysis	Nov 15
Phase 4: Evaluation & Testing					
Model evaluation on test set	Run final accuracy metric on test set, compute overall accuracy	Evaluate model performance, interpret results (precision/recall)	Generate final confusion matrix and accuracy report	Document final accuracy, assist with report	Nov 19
Phase 5: Documentation & Presentation					
Prepare final report and presentation	Write report sections (intro, dataset)	Write methodology and model architecture sections	Write result analysis section	Write challenges, conclusion, and compile report	Nov 23

Table 1: Project Plan and Task Delegation

The tasks necessary to complete this project have been outlined and delegated, as shown in Table 1. We will collaborate by holding regular meetings every Thursday at 2PM, where we will review progress, discuss challenges, and plan upcoming tasks. To ensure consistent communication, we will utilize a dedicated group chat for daily updates, quick decisions, and clarifications. For seamless code collaboration, we will maintain a shared GitHub repository where all members can contribute. To avoid code overwriting, we will follow version control best practices, including using branches for individual tasks, submitting pull requests for review, and updating the team on active work to ensure synchronization.

9 Risk Register

9.1 Data Overfitting

Overfitting occurs in machine learning when a model accounts for the noise and outliers of its training data to an extent which negatively affects its performance on new data. This commonly occurs when there is an insufficient amount of data diversity.

Likelihood: Medium

Resolution:

- Utilize data augmentation in order to vary image color, rotation, or scale to artificially expand the data set
- Track validation loss so we can catch overfitting early on in order to adjust the training accordingly

9.2 Limited Data Diversity

While the model will be tested with a selected dataset that is comprehensive, it is possible that it will not account for all possible variations of traffic signs encountered in real-world situations. This may result in the model failing to effectively generalize the data.

Resolution:

- Use multiple datasets from different regions or create our own dataset
- Evaluate the model on real-life traffic signs collected from personal photos
- Attempt to use datasets with diverse conditions such as varying weather conditions, lighting, angles, etc.

9.3 Model Training Duration

The training for a model can be a time-consuming process when the architecture is complex or if the training data is large. This may result in project delays.

Likelihood: Medium

Resolution

- Attempt to optimize system by modifying hyperparameters, simplifying model complexity, or use more efficient algorithms in order to speed up the training process
- Set a time limit for training runs and monitor validation performance to stop training when performance plateaus to avoid unnecessary system cost
- Add checkpoints to save intermediate models in case of losing progress due to long runs or run fails

9.4 Team Member Dropout

The possibility of a team member dropping the course will result in disruptions in the project process and workload distribution.

Likelihood: Low

Resolution

- Ensure multiple team members are training in important tasks, so that if in the case of a team member leaving, there will be a smooth transition of tasks
- Maintain a comprehensive documentation of the project progress and process to help teammates who take on additional tasks

10 GITHUB LINK

https://github.com/sysallykim/aps360_fall2024_group28

11 REFERENCES

REFERENCES

Obviously AI. Data cleaning in machine learning. `https://www.obviously.ai/post/data-cleaning-in-machine-learning`, January 2022. Accessed: 2024-10-04.

Pragati Baheti. Train test validation split: How to & best practices. `https://www.v7labs.com/blog/train-validation-test-set`, September 2021. Accessed: 2024-10-04.

DataCamp. A complete guide to data augmentation. `https://www.datacamp.com/tutorial/complete-guide-data-augmentation`, November 2022. Accessed: 2024-10-04.

DataCamp. What is normalization in machine learning? a comprehensive guide to data rescaling. `https://www.datacamp.com/tutorial/normalization-in-machine-learning`, January 2024. Accessed: 2024-10-04.

Abdulrahman Kerim and Mehmet nder Efe. Recognition of traffic signs with artificial neural networks: A novel dataset and algorithm. In *2021 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, pp. 171–176, 2021. doi: 10.1109/ICAIIC51459.2021.9415238.

Wei Li, Haiyu Song, and Pengjie Wang. Finely crafted features for traffic sign recognition. *International Journal of Circuits, Systems and Signal Processing*, 16, 2022. doi: 10.46300/9106.2022.16.20.

A. Madani and R. Yusof. Traffic sign recognition based on color, shape, and pictogram classification using support vector machines. *Neural Computing and Applications*, 30:2807–2817, 2018. doi: 10.1007/s00521-017-2887-x.

Nattapol Namyang and Suphakant Phimoltares. Thai traffic sign classification and recognition system based on histogram of gradients, color layout descriptor, and normalized correlation coefficient. In *2020 - 5th International Conference on Information Technology (InCIT)*, pp. 270–275, 2020. doi: 10.1109/InCIT50588.2020.9310778.

Deepali Soni, Rahul Kumar Chaurasiya, and Sneha Agrawal. Improving the classification accuracy of accurate traffic sign detection and recognition system using hog and lbp features and pca-based dimension reduction. In *Proceedings of International Conference on Sustainable Computing in Science, Technology and Management (SUSCOM)*, 2019.

Johannes Stallkamp, Marc Schlipsing, Jan Salmen, and Christian Igel. German traffic sign recognition benchmark (gtsrb). `http://benchmark.ini.rub.de/`, 2011. Accessed: 2024-10-04.

Hyontai SUG. Performance of machine learning algorithms and diversity in data. In *MATEC Web of Conferences*, volume 210. EDP Sciences, 2018. doi: 10.1051/matecconf/201821004019.