Traffic Sign Classification

Group 8:
Yiling Ding
Artemis Lu
Lucia Liu
Yiyi Sun

Agenda

Contents

01. **Project Overview**

02. **EDA**

03. **Feature Engineering**

04. **Model Selection**

05. **Future Work**

Problem Statement

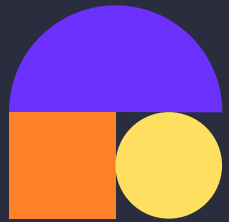
➔ The Problem We're Solving

Goal: Automatically classify traffic signs from images into 43 categories

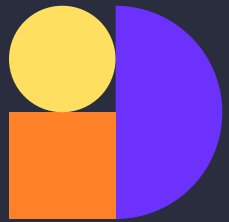
Challenge: Real-world traffic signs vary in appearance due to lighting, occlusion, and image quality

Objective: Build a robust, high-accuracy model to detect and classify signs in real time

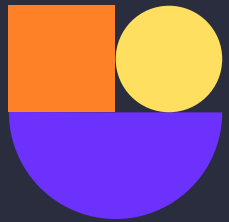
➔ Real World Impact



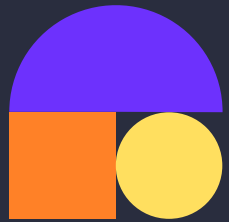
Supports autonomous vehicles and logistics automation



Enhances public safety and reduces accidents



Improves driver assistance systems in commercial and consumer vehicles



Contributes to smart infrastructure and urban planning

Hypothesis

Data Quality & Bias

Class imbalance reduces precision/recall for rare traffic signs.

Image noise (blurring, occlusion) negatively impacts prediction confidence.

Robustness

Model performance declines with real-world variations (e.g., lighting, angle).

Preprocessing (e.g., normalization, contrast enhancement) improves generalization.

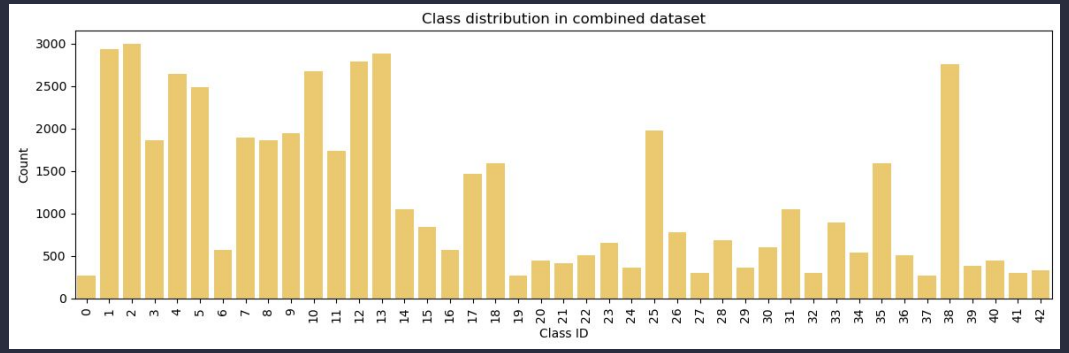
Model Performance

A baseline CNN model will perform well on frequent classes but poorly on rare or visually distinct classes.

Adding data augmentation will improve classification accuracy.

Overview of Traffic Signals Dataset

Combined Dataset Distribution of Each Class



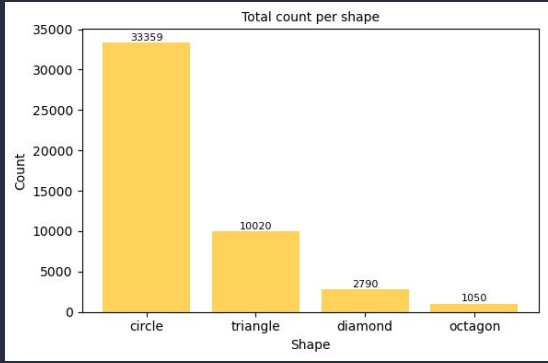
Visual Examples of Each Class

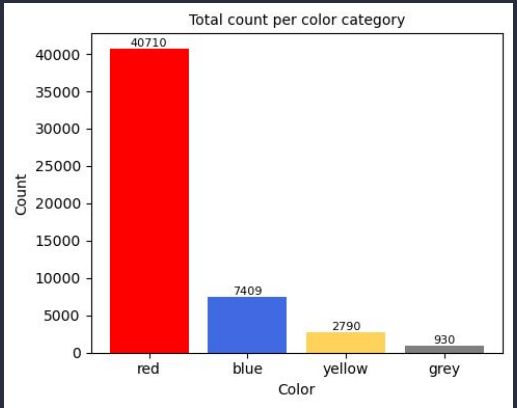
1. One class represents one signal type
2. The left panel reveals significant ***class imbalance***, with some traffic sign types appearing much more frequently
3. Some classes have 3,000+ samples, while others have fewer than 300.
4. This severe imbalance may bias the model toward majority classes.

Data Source

1. The original data files
   a. ***Train.pickle***
   b. ***Valid.pickle***
   c. ***Test.pickle.***
2. ***label_names.csv***, which contains the names of the 43 traffic sign classes.

Detailed Distribution of Shape, Color, & Brightness





Shape

Circles (>60% of data) significantly outnumber other shapes

Color

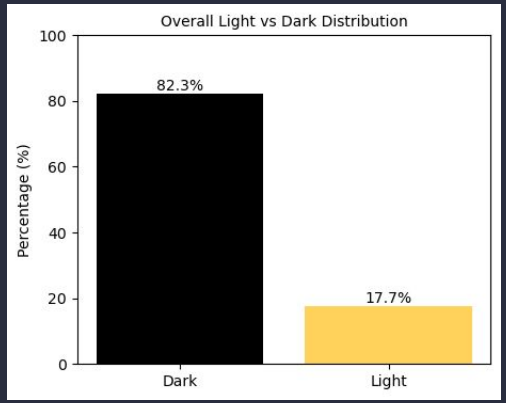
Red traffic signs dominate (80%), color features may need balanced sampling or augmentation.

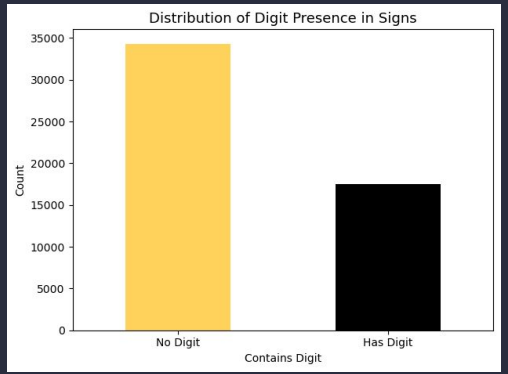
Brightness

Majority of images are captured in darker conditions, highlighting the need for low-light image handling.

Digit Presence

Digit-containing signs (~34%), suggesting need to differentiate between numeric and symbolic features when learning from sign content.





Feature Engineering & Data Transformations

Image Preprocessing

- Resizing: All images resized to 32×32 (GTSRB native size)
- Normalization: Pixel values scaled to [0, 1] for neural network stability
- One-hot/Sparse Labels: Used sparse_categorical_crossentropy — efficient for integer labels
- For model EfficientNetB0, due to its low performance, we tried to solve it by increasing the image size to 224×224 to capture more details.

Data Augmentation

Applied via a tf.keras.Sequential augmentation block:

- RandomFlip("horizontal")
- RandomRotation(±10%)
- RandomTranslation(±10%)
- RandomContrast(±25%)

Purpose: improve generalization, simulate real-world variance

Model Selection

CNN

Layer (type)	Output Shape	Param #
conv2d_25 (Conv2D)	(None, 32, 32, 32)	896
batch_normalization_21 (BatchNormalization)	(None, 32, 32, 32)	128
max_pooling2d_25 (MaxPooling2D)	(None, 16, 16, 32)	0
conv2d_26 (Conv2D)	(None, 16, 16, 64)	18,496
batch_normalization_22 (BatchNormalization)	(None, 16, 16, 64)	256
max_pooling2d_26 (MaxPooling2D)	(None, 8, 8, 64)	0
conv2d_27 (Conv2D)	(None, 8, 8, 128)	73,856
batch_normalization_23 (BatchNormalization)	(None, 8, 8, 128)	512
max_pooling2d_27 (MaxPooling2D)	(None, 4, 4, 128)	0
flatten_8 (Flatten)	(None, 2048)	0
dense_18 (Dense)	(None, 256)	524,544
dropout_9 (Dropout)	(None, 256)	0
dense_19 (Dense)	(None, 43)	11,051

Total params: 629,739 (2.40 MB)
Trainable params: 629,291 (2.40 MB)
Non-trainable params: 448 (1.75 KB)

ResNet50

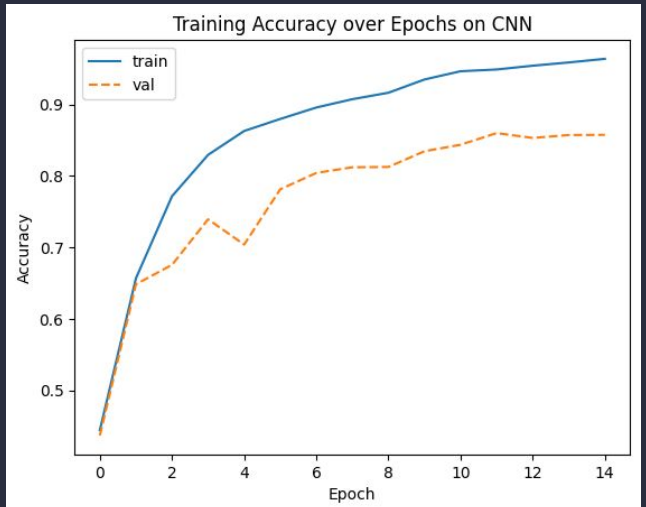
Layer (type)	Output Shape	Param #
conv2d_28 (Conv2D)	(None, 32, 32, 128)	6,272
max_pooling2d_28 (MaxPooling2D)	(None, 16, 16, 128)	0
conv2d_29 (Conv2D)	(None, 16, 16, 64)	131,136
max_pooling2d_29 (MaxPooling2D)	(None, 8, 8, 64)	0
conv2d_30 (Conv2D)	(None, 8, 8, 32)	32,800
max_pooling2d_30 (MaxPooling2D)	(None, 4, 4, 32)	0
conv2d_31 (Conv2D)	(None, 4, 4, 16)	8,208
max_pooling2d_31 (MaxPooling2D)	(None, 2, 2, 16)	0
flatten_9 (Flatten)	(None, 64)	0
dense_20 (Dense)	(None, 512)	33,280
dropout_10 (Dropout)	(None, 512)	0
dense_21 (Dense)	(None, 43)	22,059

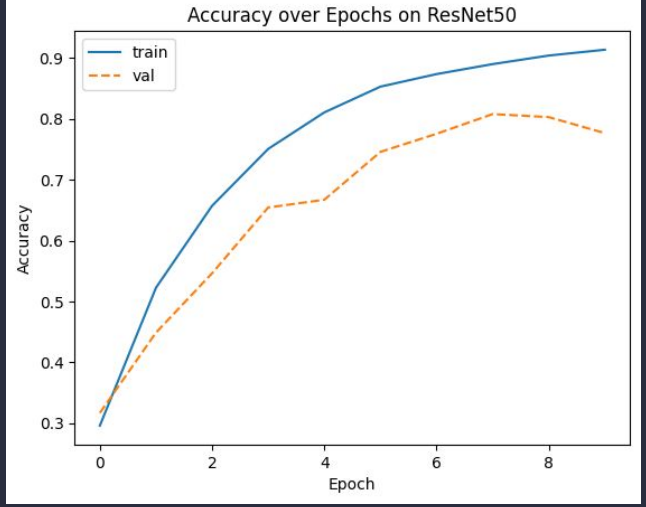
Total params: 233,755 (913.11 KB)
Trainable params: 233,755 (913.11 KB)
Non-trainable params: 0 (0.00 B)

Model Selection

x	EfficientNetB0	ResNet50	CNN
Train Accuracy	2.25%	91.82%	95.12%
Validate Accuracy	5.44%	77.67%	86.46%
Test Accuracy	5.71%	80.95%	84.99%
Recall Score	\	81.03%	85.62%
F1 Score	\	80.07%	85.24%

What we did to reduce overfitting: L2 regularization, Early stopping, Reducing epochs from 50 to 15, Adjusting learning rate





Future Work

Problem: Current pipeline assumes one sign per image — not realistic for road scenes.
Solution: Shift from classification to **object detection** to handle:

- Multiple signs
- Varying sizes, rotations, lighting
- Real-time road conditions

Next Steps:

- Train YOLOv5 or SSD on traffic sign detection
- Combine detection + classification
- Deploy on video stream or dashboard camera input

Thank you

Contribution

1. Problem Statement (Lucia)
2. Assumptions/Hypotheses about data and model (Lucia)
3. Exploratory Data Analysis (Ashley)
4. Feature Engineering & Data Transformations (Yiling)
5. Proposed Approaches (Model) with checks for overfitting/underfitting (Yiling & Artemis)
6. Proposed Solution (Model Selection) (Yiling & Artemis)
7. Results (Accuracy) and Learnings from the methodology(Artemis)
8. Future Work(Yiling)