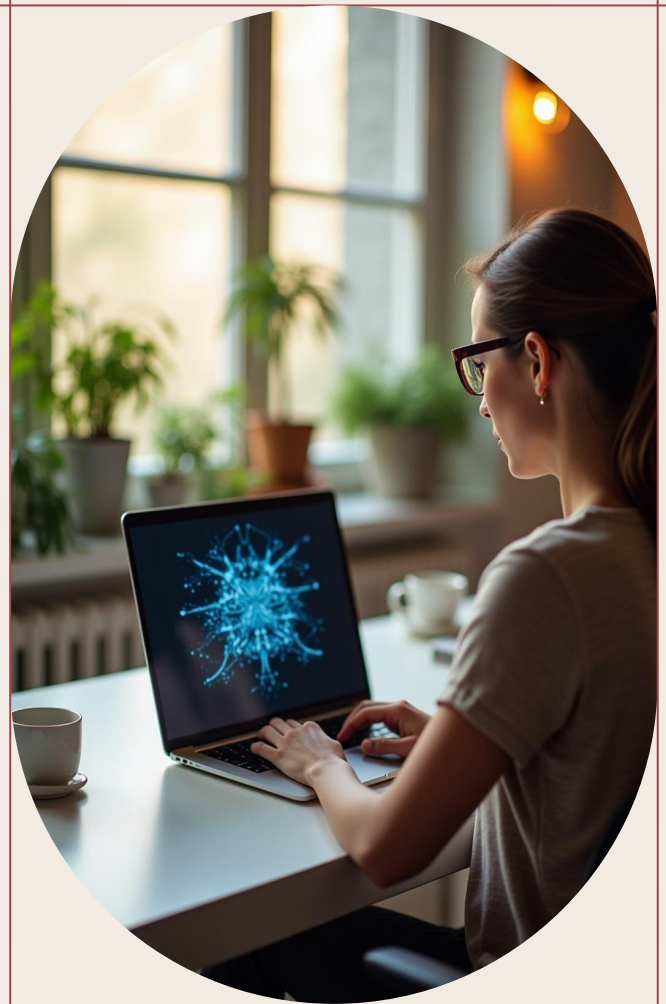
Traffic Sign Recognition with Deep Learning

Muhammad Mustafa Qaiser (2220058)
Deep Learning WS 2025

Introduction

- Traffic sign recognition is essential for autonomous driving and advanced systems.
- This project implements a compact CNN model to classify traffic signs with high accuracy.
- Our aim is to demonstrate practical deep learning applications for traffic sign understanding in real-time scenarios.

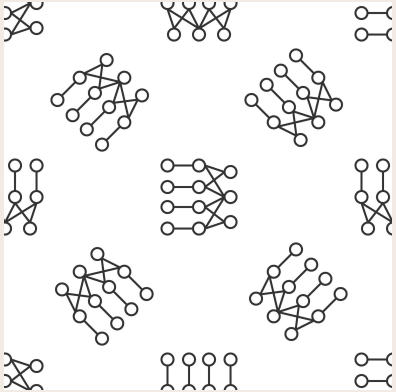


Dataset Description and Preprocessing


```
traffic_sign_small/
    labels.csv
    traffic_Data/
        DATA/
            0/
            1/
            2/
            ...
```

- The dataset consists of over 50 traffic sign classes, each represented by 32×32 RGB images.
- Data variation includes lighting, blur, and perspective challenges.
- Images are normalized and split into 80% training and 20% validation sets to ensure reliable performance evaluation.

Overview of Convolutional Neural Networks (CNNs)



- **Definition**
- **The "Flashlight" Analogy**
- **Spatial Patterns**
- **Feature Hierarchy**

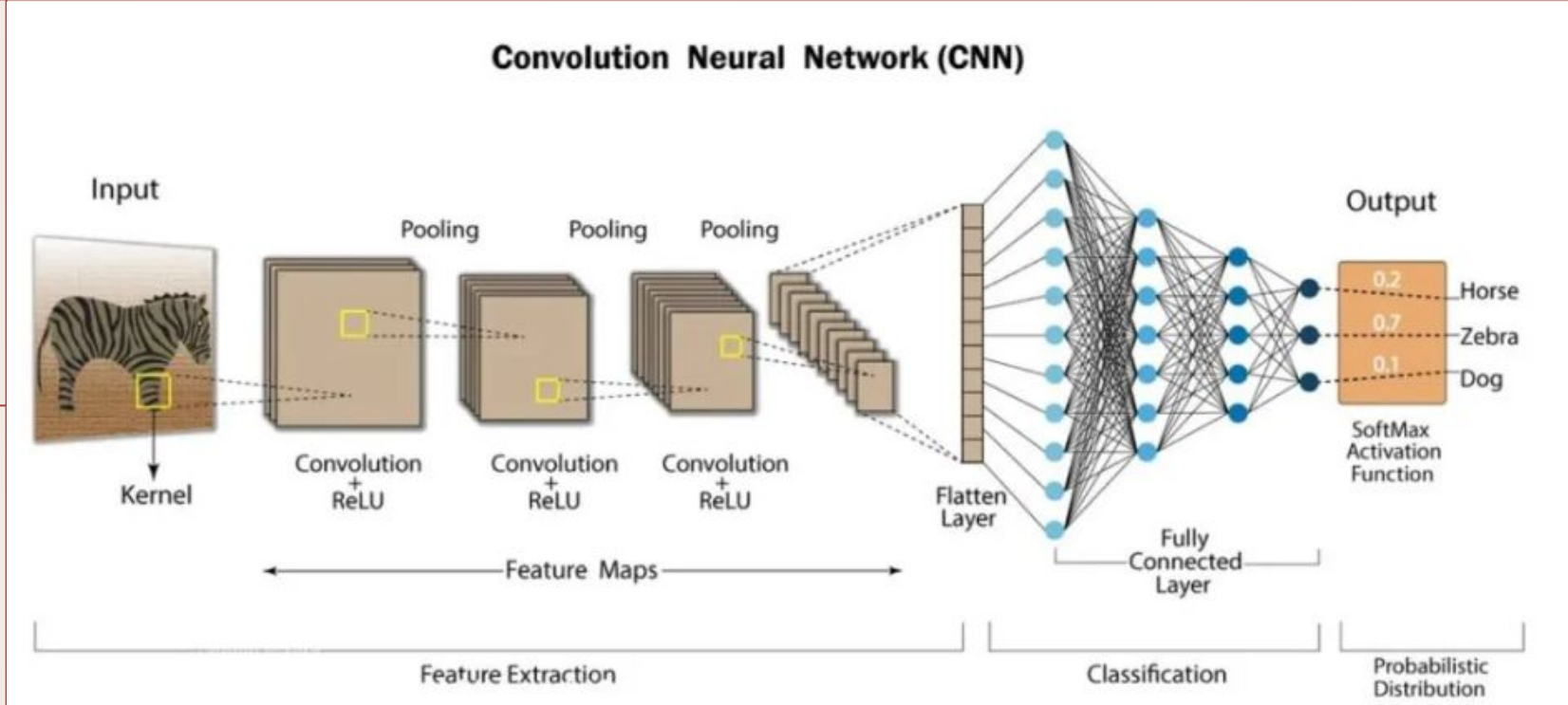


Fig. 2 “Convolution Neural Network”

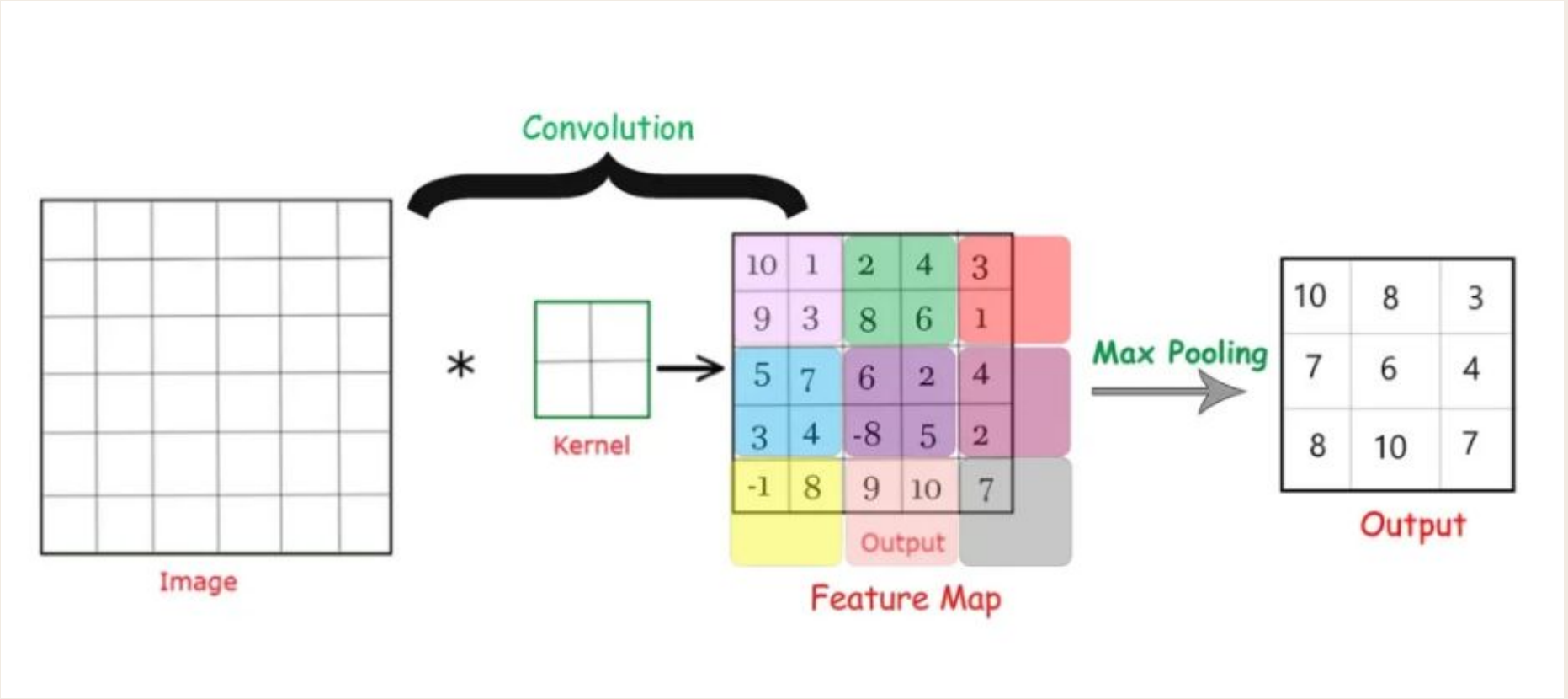


Fig.1: “CNN Visualised”

Mathematics of CNN

- The Convolution Operation
- The Activation
- Pooling Operations
- Fully Connected Layers
- The Softmax Output
- Learning: Loss and Backpropagation

The Convolution Operation

$$S(i,j) = \sum_{c=1}^{C} \sum_{m=1}^{K_H} \sum_{n=1}^{K_W} I(i+m-1, j+n-1, c) K(m,n,c) + b$$

- **Purpose:** To extract local spatial features (e.g., edges, textures) from the input data.
- **I**: The input volume with **C** channels (e.g., C=3 for RGB).
- **K**: The learned kernel (filter) of size K_H times K_W times **C**.
- **b**: A single, learned scalar bias added to every element of the output feature map **S**.
- **Parameters**: The weights in **K** and the bias b are the parameters learned during training.

The Activation

$$\sigma(x) = \max(0, x)$$

- **Purpose:** To introduce non-linearity element-wise into the feature maps.
- **x**: An input value from the feature map **S**.
- **sigma(x)**: The activated output value.
- **Mechanism:** negative input values to zero
- **Benefit:** efficient and helps mitigate the vanishing gradient problem

Pooling Operations

$$P(i,j) = \max_{m=1}^{F} \max_{n=1}^{F} A(i \cdot s + m - 1, j \cdot s + n - 1)$$

- **Purpose:** To downsample the feature map
- **A:** The input feature map to the pooling layer.
- **P(i, j):** The output value at position (i, j) in the pooled map.
- **F:** The size of the pooling filter (e.g., 2).
- **s:** The stride (step size) used to slide the pooling window.
- Mechanism: Selects the maximum value within the **F** times **F** sliding window.

Fully Connected Layers

$$H = \sigma(W \cdot F + B)$$

- F is the input vector from the previous layer.
- W and B are the learned weight matrix and bias vector.

The Softmax Output

$$\hat{y}_k = \text{Softmax}(z)_k = \frac{e^{z_k}}{\sum_{j=1}^{K} e^{z_j}}$$

- **Purpose:** To convert raw output scores into probabilities for multi-class classification.
- **z_k:** The raw output score (logit) for class k.
- **y_k:** The predicted probability for class k.
- **Property:** The resulting probabilities y_k sum to 1.
- **Usage:** Used as the activation function for the final output layer in a classification task.

Learning: Loss and Backpropagation

$$\mathcal{L}(y, \hat{y}) = -\sum_{k=1}^{K} y_k \log(\hat{y}_k)$$

$$\Delta W = \frac{\partial \mathcal{L}}{\partial W} \quad \text{and} \quad \Delta b = \frac{\partial \mathcal{L}}{\partial b}$$

- **Purpose:** To quantify the error (loss) between the model's prediction and the ground truth label.
- **y_k:** The true probability (1 or 0, for one-hot encoding) for class k.
- **Goal:** The training process seeks to find parameters that minimize this loss function
- **Backpropagation:** The gradient is calculated via the chain rule (backpropagation) to update the parameters W.

Implementation

Classification report:

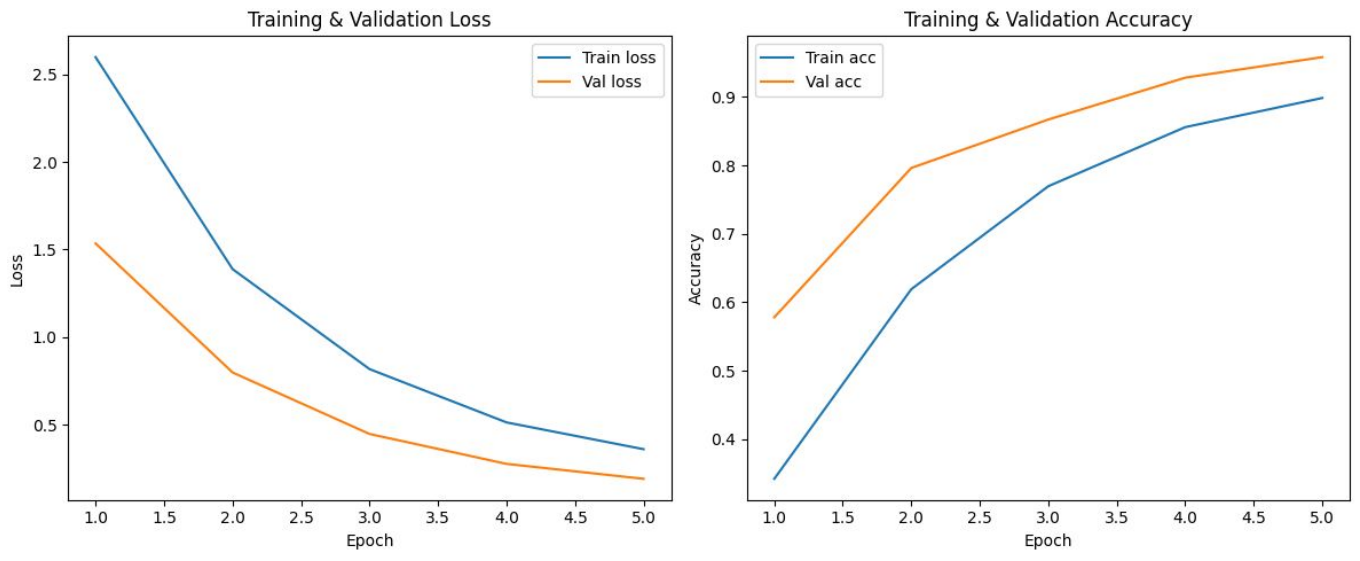
	precision	recall	f1-score	support
Speed limit (5km/h)	1.000	1.000	1.000	20
Speed limit (15km/h)	1.000	0.778	0.875	9
Dont Go straight	1.000	1.000	1.000	15
Dont Go Left	1.000	1.000	1.000	23
Dont Go Left or Right	1.000	1.000	1.000	21
Dont Go Right	1.000	1.000	1.000	6
Dont overtake from Left	1.000	1.000	1.000	22
No Uturn	1.000	0.800	0.889	5
No Car	0.976	1.000	0.988	40
No horn	1.000	1.000	1.000	30
Speed limit (40km/h)	1.000	1.000	1.000	1
Speed limit (50km/h)	0.000	0.000	0.000	0
Speed limit (30km/h)	0.824	1.000	0.903	14
Go straight or right	1.000	1.000	1.000	3
Go straight	1.000	0.667	0.800	3
Go Left	1.000	1.000	1.000	1
Go Left or right	1.000	1.000	1.000	1
Go Right	1.000	1.000	1.000	18
keep Left	0.000	0.000	0.000	1
keep Right	0.933	1.000	0.966	14
Roundabout mandatory	1.000	1.000	1.000	3
watch out for cars	1.000	1.000	1.000	109
Horn	1.000	0.750	0.857	8
Speed limit (40km/h)	1.000	1.000	1.000	43
Bicycles crossing	1.000	1.000	1.000	40
Uturn	0.889	1.000	0.941	8
Road Divider	1.000	0.600	0.750	5
Traffic signals	1.000	1.000	1.000	1
Danger Ahead	0.000	0.000	0.000	6
Zebra Crossing	0.455	1.000	0.625	30
Bicycles crossing	1.000	0.545	0.706	11

- Google Collab
- Python
- Requirements

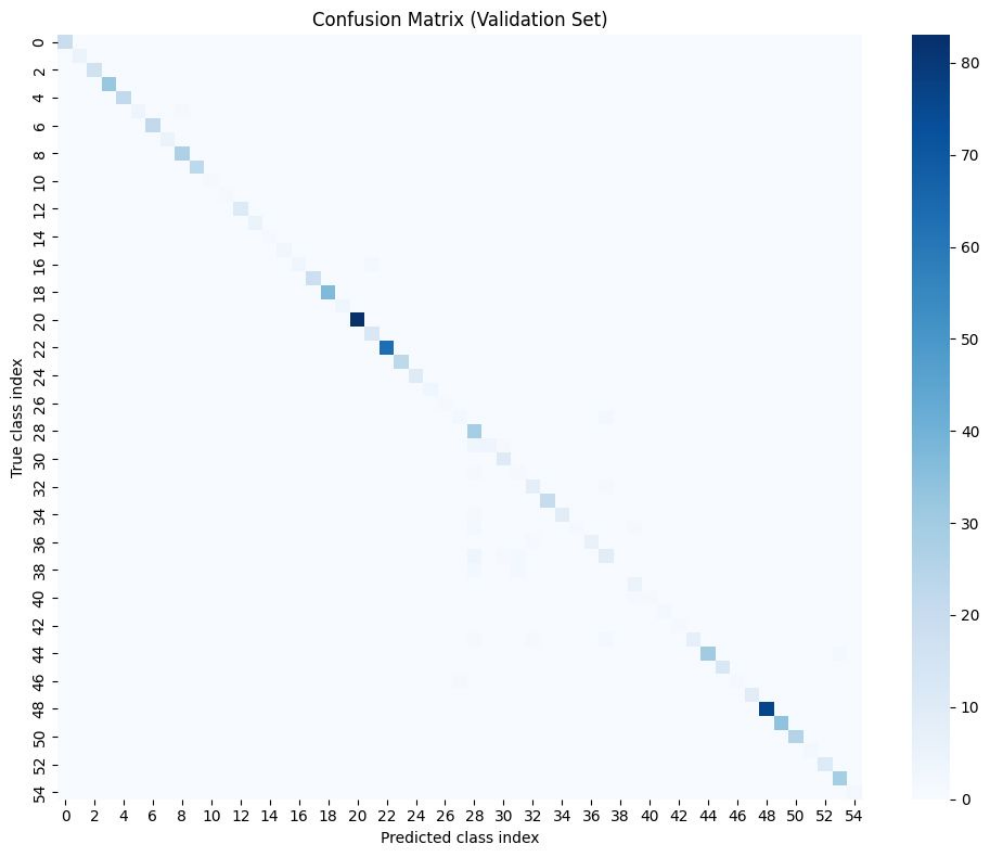
Results

- The model achieves high accuracy with tightly correlated training and validation metrics
- The confusion matrix reveals accurate classification with only a few confusions for visually similar signs.
- Inference takes approximately 4.51 milliseconds per

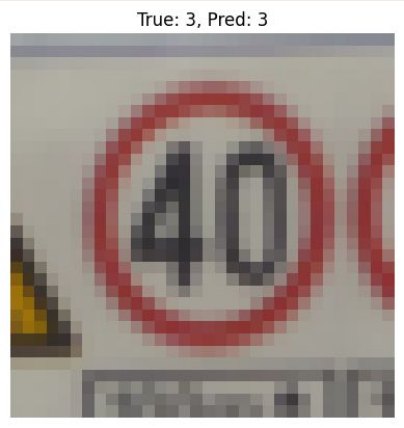
Training Accuracy

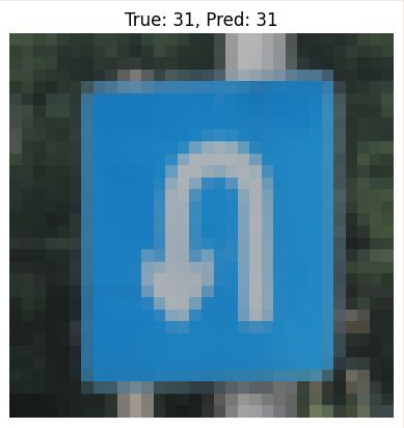


Confusion Matrix

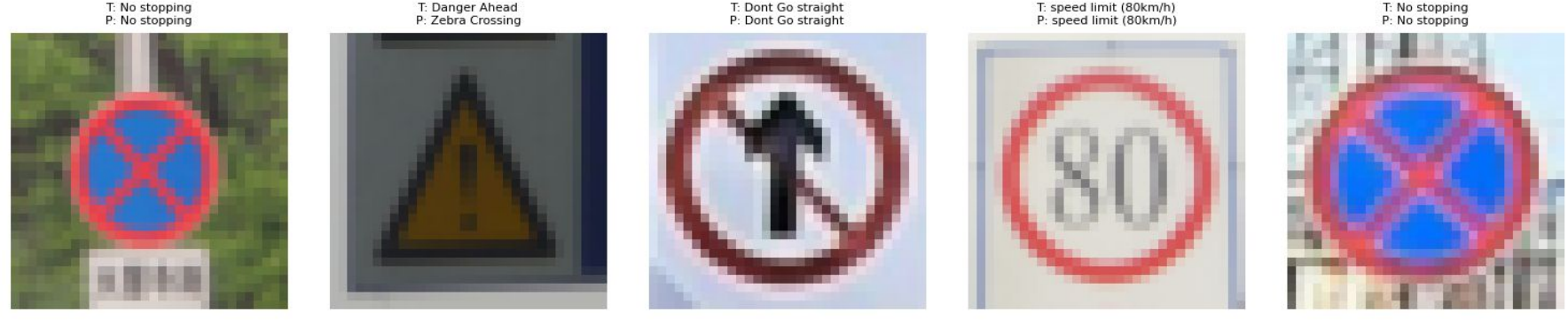


Predicted Images



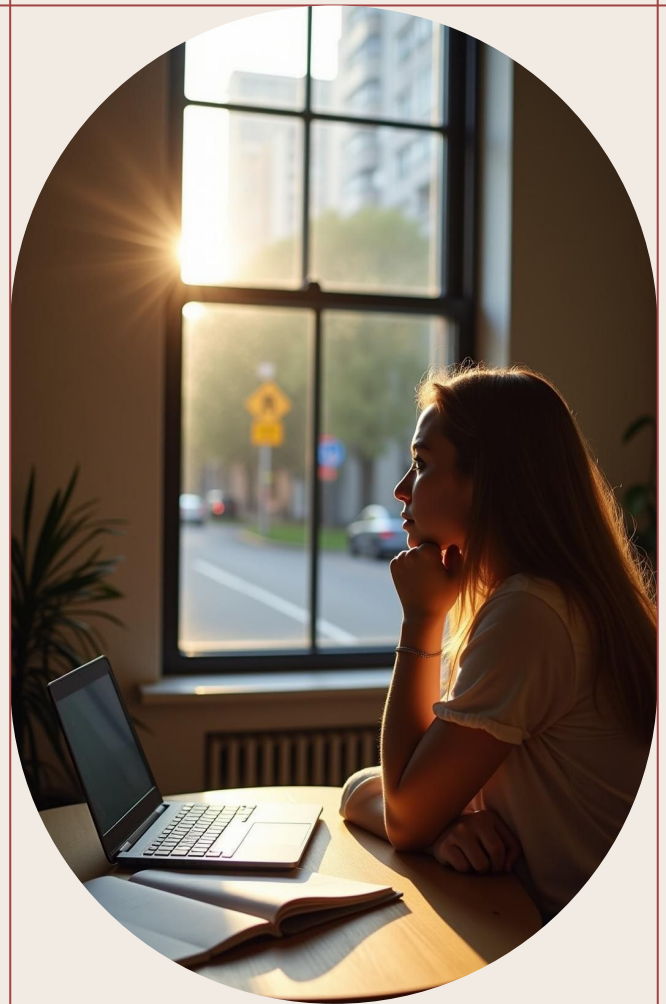


Predicted Images



Conclusions

- The compact CNN model effectively classifies traffic sign images
- Limitations include sensitivity to data diversity and lighting changes



THANK YOU

Do you have any questions?

Citations

I. Čordašić, M. Vranješ, R. Grbić and M. Herceg, "Algorithm for Extracting Signposts from the Scene and Understanding Text on Them," 2024 Zooming Innovation in Consumer Technologies Conference (ZINC), Novi Sad, Serbia, 2024, pp. 60-65, doi: 10.1109/ZINC61849.2024.10579407.

A. Biswas, A. R. Chowdhury and R. Selvanambi, "Lane detection and traffic sign detection for autonomous driving systems," 2nd International Conference on Computer Vision and Internet of Things (ICCVIoT 2024), Coimbatore, India, 2024, pp. 206-211, doi: 10.1049/icp.2024.4425.

S. Khalid, J. H. Shah, M. Sharif, F. Dahan, R. Saleem and A. Masood, "A Robust Intelligent System for Text-Based Traffic Signs Detection and Recognition in Challenging Weather Conditions," in IEEE Access, vol. 12, pp. 78261-78274, 2024, doi: 10.1109/ACCESS.2024.3401044.

Y. Lecun, L. Bottou, Y. Bengio and P. Haffner, "Gradient-based learning applied to document recognition," in Proceedings of the IEEE, vol. 86, no. 11, pp. 2278-2324, Nov. 1998, doi: 10.1109/5.726791.

G. Cui, "Research on Recognition and Classification Technology Based on Deep Convolutional Neural Network," 2021 IEEE 3rd Eurasia Conference on IOT, Communication and Engineering (ECICE), Yunlin, Taiwan, 2021, pp. 353-357, doi: 10.1109/ECICE52819.2021.9645706.

W. Wang, G. Wen and Z. Zheng, "Design of Deep Learning Mixed Language Short Text Sentiment Classification System Based on CNN Algorithm," 2022 IEEE 2nd International Conference on Mobile Networks and Wireless Communications (ICMNWC), Tumkur, Karnataka, India, 2022, pp. 1-5, doi: 10.1109/ICMNWC56175.2022.10031786.

***All Images are from google stock images or created by implementation or screenshots from my paper except:**

Fig 1. "CNN Visualised" - Link: https://svitla.com/blog/cnn-for-image-processing/

Fig 2. "Convolutional Neural Network" - Link: https://svitla.com/blog/cnn-for-image-processing/