# Project Title

## Extending Traffic Sign Recognition with ResNet-18 Transfer Learning and Explainable Deep Learning

---

### Team Details

- **Team Member:** Pavan Kumar Satram - Model design, experimentation, implementation, analysis, report writing

---

### 1. Overview of the Problem/Project

### 1.1 General Information / Idea about the Project and Problem

Traffic sign recognition is a core computer-vision task in modern driver-assistance systems (ADAS) and autonomous vehicles. Correctly interpreting speed limits, prohibitions, and warning signs is essential for safe driving decisions in real time. Mistakes in detecting or classifying traffic signs can lead to violations of road rules and potentially unsafe behaviour.

The German Traffic Sign Recognition Benchmark (GTSRB) is a widely used dataset for this task. It includes 43 traffic-sign classes and contains significant variation in:

- Lighting conditions (bright sunlight, shadows, overcast)
- Motion blur and sensor noise
- Partial occlusions and cluttered backgrounds
- Perspective distortions and scale changes

This makes GTSRB a realistic and challenging benchmark for machine learning models.

In the original phase of this CIS 678 project, the focus was on comparing a classical machine learning baseline (linear regression with PCA + Ridge) to two custom convolutional neural networks (CNNs):

- A plain CNN without explicit regularization, and
- A regularized CNN with batch normalization, dropout, and weight decay.

That phase emphasized how regularization affects overfitting and training dynamics on GTSRB.

This second phase extends the project in two main ways:

1. Modeling: Introduce a transfer learning approach using ResNet-18, a deeper residual network pretrained on ImageNet.
2. Explainability and analysis: Add Grad-CAM visualizations and a misclassification viewer to better understand how the model makes decisions and where it fails.

The goal is not only to push performance beyond the regularized CNN, but also to build a more realistic and interpretable model that could plausibly be used in driver-assistance applications.

---

### 1.2 Summary of Previous Baseline and CNN Models

In the earlier stage of the project, three models were implemented on GTSRB:

**• Linear Regression (PCA + Ridge)**

- Images were resized to 64×64, converted to grayscale, and flattened to 4,096-dimensional vectors.
- PCA reduced dimensionality to 150 components, and Ridge regression was used as a multi-output linear model on one-hot encoded labels.
- This model achieved test accuracy of roughly 61.8%, clearly showing that a simple linear model is insufficient for highly nonlinear image data.

**• Plain Small CNN**

- Architecture:
    - 3 convolutional layers with 32, 64, and 128 channels, each followed by ReLU and max pooling.
    - A fully connected layer with 256 units before the final classification layer.
- No batch normalization, dropout, or weight decay was used.
- The model achieved test accuracy of about 91.5%, but showed strong overfitting:
    - Training accuracy reached nearly 100%.
    - Validation accuracy plateaued much lower, with a visible gap between train and validation curves.

**• Regularized SmallCNN**

- Same base architecture as the plain CNN, but with:
    - Batch normalization after each convolutional layer.
    - Dropout in the fully connected layer.
    - Weight decay added in the optimizer.
- This regularized CNN achieved test accuracy of approximately 97.4%, and the training/validation curves were much closer together, demonstrating that regularization significantly improves generalization.

These three models established a clear performance ladder:

- Linear baseline → Plain CNN → Regularized CNN,

and highlighted how architectural choices and regularization strategies affect generalization on GTSRB.

---

### 1.3 Extension Model: ResNet-18 with Transfer Learning and Explainability

To further improve performance and realism, the extension uses ResNet-18, a widely used residual network originally trained on ImageNet. Instead of training a deep model from scratch on GTSRB, the project uses transfer learning:

- The ResNet-18 backbone (all convolutional layers, residual blocks, and batch normalization layers) is reused.
- Only the final fully connected layer is adapted to the 43-class GTSRB task.

**Key aspects of the extended model**

- **Architecture:**
    - Standard ResNet-18 backbone with residual blocks arranged in four stages (layer1–layer4).
    - The original final layer fc: Linear(512 → 1000) (for ImageNet) is replaced with Linear(512 → 43) to match the number of traffic-sign classes.
- **Input representation:**
    - Images are resized to 224×224 (the standard input size used for ImageNet-resized ResNet models).
    - Inputs are normalized using the ImageNet mean and standard deviation for each RGB channel, aligning the GTSRB images with the distribution the network was pretrained on.
- **Training strategy:**
    - Fine-tuning the network on GTSRB using the Adam optimizer with a small learning rate and weight decay to avoid destroying pretrained features too quickly.
    - A separate validation split (15%) from the original training data is used for model selection and monitoring, consistent with the original project's approach.
- **Explainability tools:**
    - Grad-CAM (Gradient-weighted Class Activation Mapping) is used to visualize spatial regions that most influence the model's prediction for each image.

  - A misclassification viewer shows examples where ResNet-18 makes incorrect predictions, alongside the true and predicted labels, supporting qualitative error analysis.

This extension therefore combines a deeper, pretrained architecture with explainability and error analysis, moving closer to realistic deployment standards.

---

## 2. Evaluation Methodology

### 2.1 Data, Preprocessing, and Train/Validation/Test Setup

All experiments use the GTSRB dataset.

- The official test set of 12,630 images is kept unchanged and used only for final evaluation.
- The official training set is split into:
  - 85% training subset
  - 15% validation subset

This mirrors the procedure used in the original project so that all models (baseline, CNNs, and ResNet-18) are evaluated under comparable conditions.

**Preprocessing and augmentation for ResNet-18**

**Training images:**

- Resized to 224×224.
- Augmented with light data augmentation to improve robustness:
  - Small random rotations (e.g., ±5 degrees).
  - Mild colour jitter on brightness and contrast.
- Converted to PyTorch tensors and normalized using ImageNet mean and standard deviation.

**Validation and test images:**

- Resized to 224×224.
- Only normalized (no augmentation), to provide a clean estimate of the model's generalization performance.

**Training configuration**

- **Optimizer:** Adam
- **Learning rate:** 1e-4

- **Weight decay:** 1e-4
- **Batch size:** 128
- **Loss function:** Cross-entropy loss
- **Hardware:** GPU (CUDA) when available, otherwise CPU

This configuration is similar to the deep-learning setup used in the original small-CNN experiments but adapted to the ResNet-18 backbone and 224×224 inputs.

---

### 2.2 Evaluation Metrics and Visualization Tools

The main evaluation metric is **overall classification accuracy** on the held-out GTSRB test set.

Additional diagnostics include:

- **Class-wise precision, recall, and F1-score**
  - Shows how well the model performs on each of the 43 classes, which is important when some signs are less frequent.
- **Confusion matrix**
  - Provides a visual summary of which classes are frequently confused with each other (e.g., similar speed-limit signs).
- **Training vs. validation curves (loss and accuracy)**
  - Helps monitor learning dynamics, detect overfitting, and compare behaviour between ResNet-18 and the previous CNNs.
- **Misclassification viewer**
  - Displays a grid of misclassified test images with both true label and predicted label, enabling intuitive analysis of failure cases.
- **Grad-CAM visualizations**
  - Produces heatmaps overlaid on input images, indicating where the network “looked” to make its prediction.
  - Crucial for interpretability in safety-critical contexts.

Combined, these tools provide a rich quantitative and qualitative assessment of the extended model and allow direct comparison to the earlier baseline and CNN models.

## 3. Results and Analysis

### 3.1 Quantitative Performance Comparison

Using the same test set and metrics as in the original work, a fourth model—ResNet-18 (Transfer Learning)—is added to the comparison:

**Final Test Accuracy Comparison**

| **Model** | **Test Accuracy** |
|---|---|
| Linear Regression (PCA + Ridge) | 0.6181 |
| Plain CNN | 0.9145 |
| Regularized CNN | 0.9739 |
| ResNet-18 (Transfer Learning) | ≈ 0.9895 |

Key observations:

- The ResNet-18 model improves test accuracy from roughly 97.4% (regularized CNN) to around 98.9%.
- In terms of error rate, this is a reduction from ~2.6% down to ~1.1%.
- Even when modest data augmentation is used, ResNet-18 consistently outperforms the earlier models on the clean GTSRB test set.

These results demonstrate that transfer learning from a large, diverse dataset (ImageNet) provides a powerful starting point for traffic sign recognition, and that the ResNet-18 architecture can capture finer details needed to distinguish between similar sign categories.

---

### 3.2 Training Dynamics and Overfitting

Training and validation curves for ResNet-18 show the following behaviour:

- Both training and validation accuracy increase quickly in the first few epochs due to strong pretrained features.
- Training accuracy eventually approaches 100%, but
- Validation accuracy stabilizes close to 98–99% and does not collapse, indicating moderate but manageable overfitting.
- The validation loss tends to flatten rather than spike sharply, which further suggests that the model maintains good generalization.

When compared to previous models:

- The plain CNN suffered from more visible overfitting: 100% training accuracy with a large gap to validation accuracy.
- The regularized CNN corrected much of that and generalized well.
- ResNet-18 behaves similarly to the regularized CNN in terms of overfitting, but with a higher performance ceiling thanks to its deeper, residual architecture and pretrained weights.

Data augmentation slightly lowers peak validation accuracy compared to a non-augmented version, but provides improved robustness to rotated, slightly blurred, or differently lit images—closer to real-world deployment conditions.

---

### 3.3 Misclassification Patterns

Using the misclassification viewer, several recurring error patterns were observed for ResNet-18:

- **Confusion between speed-limit signs with different numeric values:**
    - For example, misclassifying “Speed Limit 60” as “Speed Limit 80” or vice versa.
    - These signs share the same circular shape and colours; only the digits differ, which can be hard to read under blur or low resolution.
- **Confusion among triangular warning signs:**
    - Warning signs often share a red triangular border with different interior icons (e.g., road work, general danger, slippery road).
    - When icons are small or noisy, the model sometimes focuses on the triangle shape more than the specific symbol.
- **Errors on degraded images:**
    - Some mistakes occur on heavily blurred, partially occluded, or poorly lit images where even a human might struggle to recognize the sign.

Overall, these misclassification patterns are similar to those seen in the regularized CNN, but less frequent. ResNet-18’s extra depth and pretrained features appear especially beneficial for handling difficult or low-quality images where simple models may fail.

### 3.4 Grad-CAM Explainability

Grad-CAM visualizations were generated to understand the spatial focus of ResNet-18 for both correct and incorrect predictions.

For correctly classified samples:

- The heatmaps typically highlight the central region of the traffic sign, including:
    - The digits in speed-limit signs.
    - Distinct interior icons in warning or prohibition signs.
    - The geometric boundary (circle, triangle) that characterizes the sign.
- Background areas (road, sky, buildings) usually show low activation, indicating that the model primarily relies on the sign itself rather than context.

For **misclassified samples**:

- In some cases, Grad-CAM shows attention focused on a subset of the sign (e.g., only the border), while the icon or digits are under-emphasized.
- In a few images, the heatmap spreads into the background, suggesting that noisy context or clutter may have influenced the prediction.
- This behaviour is particularly visible in heavily blurred or overexposed images.

These visualizations:

- Increase trust in the model when it looks at logically relevant regions and correctly classifies the sign.
- Help diagnose failure modes when it attends to the wrong regions or insufficiently focuses on key features.

In safety-critical applications such as traffic sign recognition, this kind of interpretability is valuable for both debugging and accountability.

---

## 4. Suggestions and Recommendations

The extension already implements several natural enhancements over the original project, including:

- A deeper, pretrained architecture (ResNet-18).
- Data augmentation to increase robustness.
- Misclassification visualization and Grad-CAM for interpretability.

Potential future extensions include:

1. **More advanced architectures**

    - Experiment with deeper networks such as ResNet-34 or ResNet-50 to see if additional depth brings further gains.
    - Investigate lightweight architectures like MobileNetV3 or EfficientNet for deployment on embedded hardware with limited compute.

2. **Ensemble methods**
    - Combine predictions from the regularized CNN and ResNet-18 (e.g., via averaging or weighted voting) to potentially further reduce error rates.
    - Ensembles often improve robustness to difficult test cases.
3. **Systematic hyperparameter tuning**
    - Explore learning-rate schedules (StepLR, CosineAnnealingLR).
    - Use label smoothing to reduce overconfident predictions.
    - Experiment with partial freezing of early ResNet layers to trade off training time and generalization.
4. **Real-time demo / deployment**
    - Integrate the trained ResNet-18 into a live video pipeline (e.g., webcam or recorded driving footage) to simulate an in-vehicle traffic sign recognition system.
    - Evaluate latency, stability, and performance under realistic motion and lighting conditions.
5. **Richer explainability tools**
    - Complement Grad-CAM with methods like Guided Backpropagation, Integrated Gradients, or Layer-wise Relevance Propagation.
    - Compare and cross-validate explanations across methods to ensure stability and reliability.

---

## 5. Project Learning / Task Assignments

### Team Member – Pavan Kumar Satram

- Reviewed the original GTSRB project, including the linear baseline and CNN models, and their training/evaluation pipelines.
- Implemented data loading and preprocessing for ResNet-18, including resizing to 224×224 and applying ImageNet-style normalization.
- Fine-tuned a pretrained ResNet-18 on the GTSRB dataset using a consistent train/validation/test split for fair comparison.

- Developed evaluation routines for test accuracy, classification reports, and confusion matrices.
- Implemented a misclassification viewer to visually inspect and analyse model errors on the test set.
- Integrated Grad-CAM to generate visual explanations of model decisions, including heatmap overlays on original images.
- Analysed and compared ResNet-18 performance against the Linear Regression baseline, Plain CNN, and Regularized CNN, and authored this extended report describing methodology, results, and insights.

---

### 6. Conclusion

This extended project demonstrates how transfer learning with ResNet-18 and explainability tools can significantly advance a traffic sign recognition pipeline originally built on simpler models.

Key conclusions:

- **Performance improvement:**
    - The ResNet-18 model achieves around 98.9% test accuracy, outperforming both the plain and regularized CNNs and drastically improving over the classical linear baseline.
    - The error rate is reduced to just over 1%, which is strong for this dataset and task.
- **Generalization and robustness:**
    - Light data augmentation (e.g., rotations, mild colour jitter) improves the model's ability to handle realistic variations in the input data.
    - Training and validation curves show good convergence and moderate overfitting, comparable to the regularized CNN but at a higher accuracy level.
- **Interpretability and diagnostic insight:**
    - Misclassification analysis reveals that most remaining errors occur between visually similar signs or on very degraded images.
    - Grad-CAM visualizations confirm that the model usually focuses on meaningful sign regions (digits, icons, borders), increasing trust in its decisions and helping diagnose problematic cases.

Overall, the project illustrates a natural and realistic evolution of a machine learning system:

1. Start with simple baselines and understand their limitations.

2. Introduce CNNs and regularization to combat overfitting and improve performance.
3. Incorporate deep pretrained architectures like ResNet-18 to push accuracy and robustness further.
4. Add data augmentation, error analysis, and explainability to build a model that is not only accurate, but also more interpretable and deployment ready.

This progression mirrors real-world machine learning practice in industry, especially for safety-critical applications such as traffic sign recognition in driver-assistance systems.

---

### 7. References

- **GTSRB dataset website:**
  https://benchmark.ini.rub.de/gtsrb_dataset.html
- Stallkamp, J., Schlipsing, M., Salmen, J., & Igel, C. (2012).
  *Man vs. Computer: Benchmarking Machine Learning Algorithms for Traffic Sign Recognition.*
- **scikit-learn documentation:**
  https://scikit-learn.org
- **PyTorch documentation:**
  https://pytorch.org/docs/stable/
- **Torchvision GTSRB dataset implementation:**
  https://pytorch.org/vision/main/generated/torchvision.datasets.GTSRB.html