Paper for COMP30027 Report

Anonymous

1. Introduction

This report investigates the classification of traffic signs using a subset of the German Traffic Sign Recognition Benchmark (GTSRB) dataset. Three supervised learning methods are evaluated: Convolutional Neural Networks (CNNs) with residual blocks, Support Vector Machines (SVM), and Extreme Gradient Boosting (XGB). The study compares the performance of three methods, examines how architectural design and feature engineering influence classification outcomes, and explores the implications of the observed results. The strengths and limitations of each model are analysed relating to the characteristics of the dataset.

2. Methodology

This project adopts a two-stage framework using traditional machine learning models and a deep learning model as base models and then combined using a stacking ensemble.

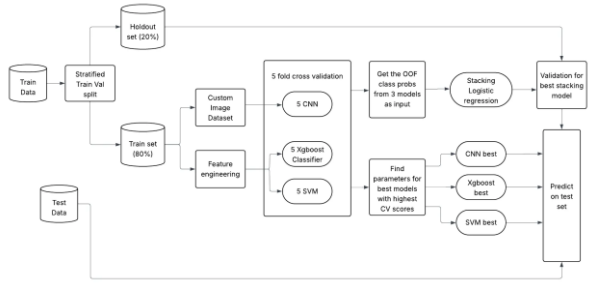

Figure 1 - The full workflow of project

2.1 Data splitting

The dataset contains 5,488 labelled images, stratified into 80% training and 20% holdout subsets. The holdout set is reserved as additional validation set for models, especially the stacking model to evaluate on.

2.2 Preprocessing Pipelines

Separate preprocessing pipelines are applied based on model type.

2.2.1 Machine Learning (SVM, XGB)

These models require structured inputs, so image data are converted into tabular features. Three groups of features are used: Histogram of Oriented Gradients reduced via Principal Component Analysis (HOG_PCA), colour histograms, mean RGB values, and additional features. These features are provided by the assignment and they capture structural and chromatic information in the images. Although additional engineered features (e.g., region-based aggregation, contour descriptors, and feature interactions) were experimented with, they offered negligible improvement and were excluded. The final feature set comprises 120 dimensions and is used consistently across SVM and XGB models.

2.2.2 Deep Learning (CNN)

The CNN directly learns from raw image pixels through convolutional layers and residual blocks. Input images are augmented using random rotations, brightness adjustments, and resizing to improve generalization and robustness to visual variability.

2.3 Model Training

All base models are trained using Stratified 5-fold cross-validation (CV). This ensures robust model evaluation and provides out-of-fold (OOF) class probability predictions, which are used as inputs for the second-stage stacking model.

2.4 Model Selection

Three base models were selected for their complementary strengths: SVM for robustness in high-dimensional spaces, XGB for capturing nonlinear feature interaction in tabular data, and CNN for its well-known capacity in image recognition tasks.

2.5 Implementation details

SVM was experimented to perform best with RBF kernel and C=10, trained on standardized fold-wise data. XGB is employed by the multi-class logarithmic loss objective, 1000 boosting rounds, and a maximum tree depth of 5, with the rest mostly default parameters. CNN utilised cross-entropy loss with the Adam optimizer, trained for 20 epochs, and the architecture is shown in Figure 2 (more in discussion).

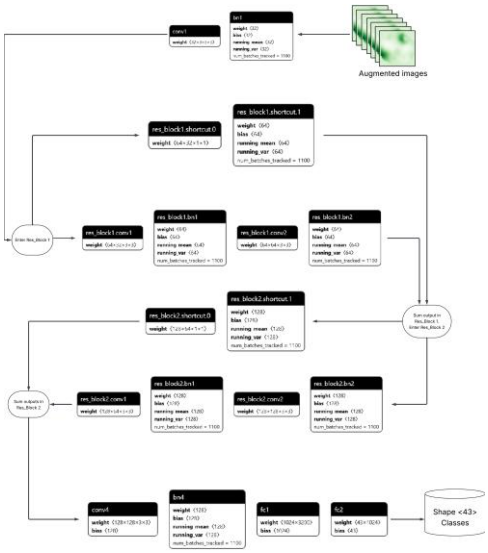

Figure 2 - The architecture of CNN

Hyperparameter tuning for SVM and XGB was omitted due to minimal gains and overfitting risks. Experiments showed that tuned parameters slightly improved CV scores but lower test accuracies. For CNN, its base configuration achieved rapid convergence and robust CV performance, rendering additional tuning unnecessary.

2.6 Stacking Ensemble

A logistic regression model is used as the meta-classifier. It is trained on the concatenated OOF class probability predictions from the base models. This ensemble approach aims to leverage the complementary strengths of each base learner.

2.7 Evaluation

Accuracy and **F1-score** are used as evaluation metrics. Accuracy serves as the final assessment metric on the test set. F1-score is useful for understanding class-wise balance in this imbalanced multi-class setting.

Additionally, visualisations such as class-wise performance curves, training loss plots, and Sankey plots are employed to interpret model behaviour and misclassification trends.

3. Result

The classification performance of each model is summarized in Table 1. To provide a more detailed comparison across classes, Figures 3–6 illustrate the per-class F1-score, precision, and recall curves for all 43 traffic classes of different models evaluated on the holdout set. The use of the holdout set is necessary here to ensure a fair comparison of base models with the stacking model. In addition, Figures 7–10 present True vs Predicted class mappings using Sankey plots, which offer more interpretable visualisations than traditional confusion matrix given the high multiclass count.

Models	Cross validation average		Holdout set scores		Test set
Metrics	Acc	F1	Acc	F1	Acc
CNN	0.974	0.966	0.988	0.990	0.992
XGB	0.779	0.735	0.802	0.766	0.422
SVM	0.822	0.804	0.855	0.838	0.443
Stacking			0.989	0.984	0.987

Table 1 – The results of base models and stacking model

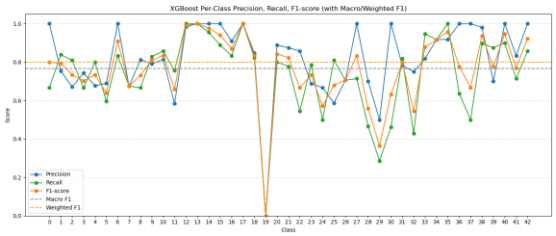

Figure 3 – Per-class evaluation metrics for XGB

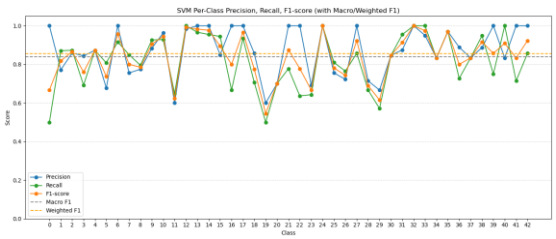

Figure 4 - Per-class evaluation metrics for SVM

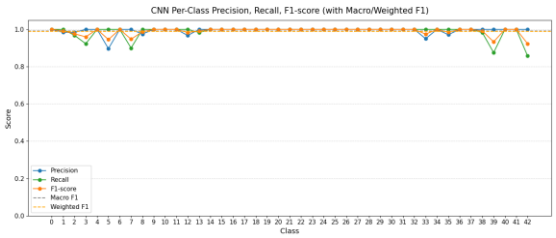

Figure 5 - Per-class evaluation metrics for CNN

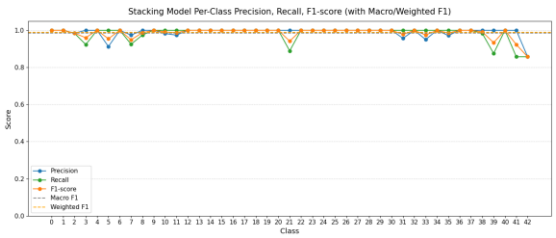

Figure 6 - Per-class evaluation metrics for Stacking

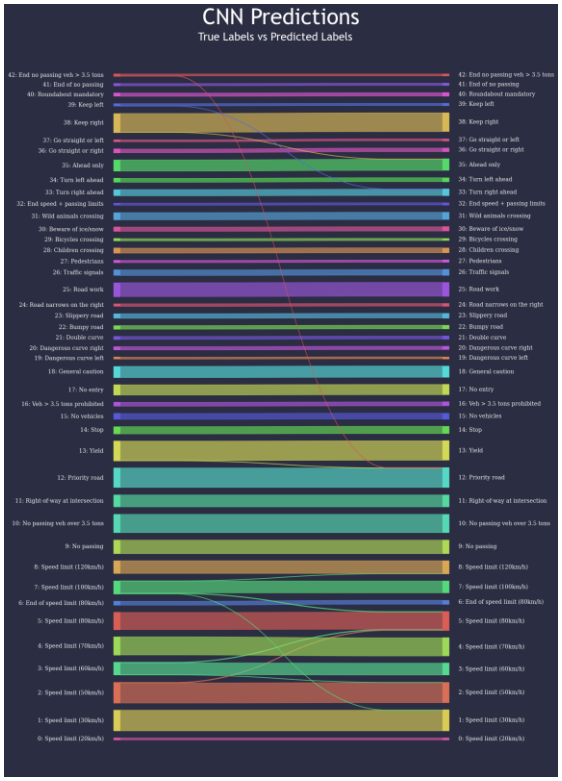

Figure 7 – Sankey plot of CNN predictions

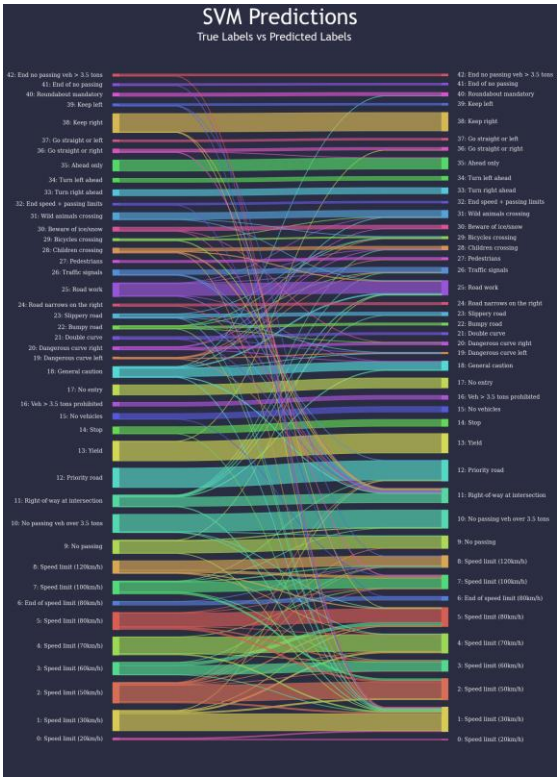

Figure 9 - Sankey plot of SVM predictions

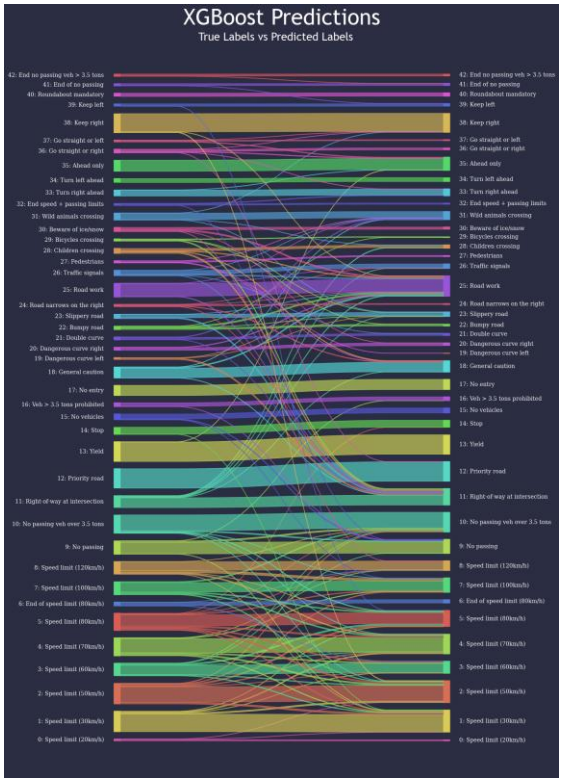

Figure 8 - Sankey plot of XGB predictions

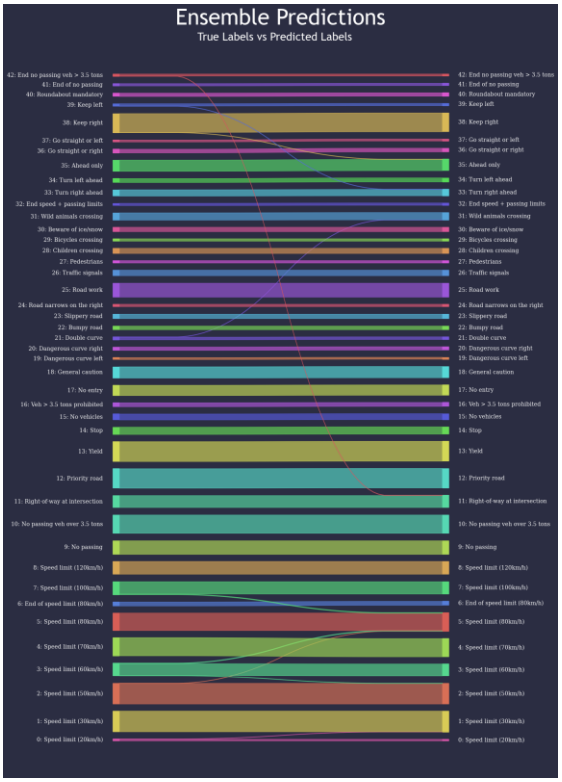

Figure 10 - Sankey plot of Stacking predictions

4. Discussion and Critical Analysis

4.1 Model Performance Across Splits

As shown in Table 1, both XGB and SVM models exhibit higher accuracy on the holdout set compared to their CV averages. This reflects higher evaluation variance in CV, where each fold encounters a different 20% of unseen data rather than just a stratified split from the full dataset. The relative consistency between CV and holdout metrics also suggests low model variance within the training domain, implying that the tabular features successfully capture some stable and repeating patterns under similar distributions.

However, the test accuracies of both models drop to around 0.40, indicating a significant distribution shift between the training and test sets. This shift is likely due to differences in image conditions (e.g., brightness, resolution) or class imbalance. While CV and holdout evaluations suggest low-variance performance under the training distribution, the test set reveals limited generalisation to new data. In contrast, the CNN consistently performs well across all splits, demonstrating stronger robustness to distributional changes. A detailed analysis of its behaviour is presented in Section 4.3.

The stacking ensemble shows no significant improvement over the CNN alone. This implies that XGB and SVM do not capture complementary patterns absent in CNN's feature space. From a bias-variance trade-off standpoint, the ensemble does not reduce bias further than the CNN, nor does it reduce variance meaningfully. Thus, the CNN's performance represents the project's greatest outcome.

4.2 Analysis of XGB and SVM

Despite having lower scores, XGB and SVM exhibit instructive differences. As shown in Table 1, SVM outperforms XGB on both CV and holdout evaluations. While both models share similar F1-score trends (Figures 3–4), SVM achieves a more stable precision-recall balance. This advantage likely arises from SVM's RBF kernel, which maps features into a high-dimensional space to better separate classes with non-linear boundaries. This is critical in image-based tasks where spatial relationships matter. In contrast, XGB's tree-based splits only prioritise individual feature thresholds and lacks mechanisms to model spatial and local dependencies.

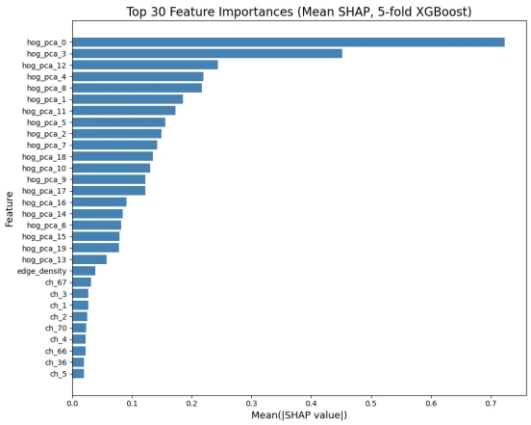

Figure 11 – Feature importance of XGB

One strength of XGB is its interpretability through feature importance analysis. Figure 11 reveals that HOG-PCA features which are compressed representations of edge and texture information dominate the decisions process, while colour histogram and other features contribute minimally. This is reasonable as traffic sign recognition relies more on shape than colour, and colour histograms alone may be ambiguous without spatial context.

To further diagnose errors, classes were manually grouped into five clusters based on shape and colour. Both models exhibit **intra-cluster** confusion: XGB tends to confuse red-white-triangular signs, while SVM struggles with red-white circular ones.

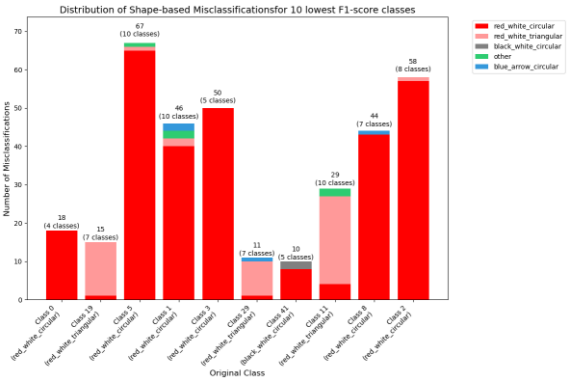

Figure 12 – The misclassified proportion by SVM

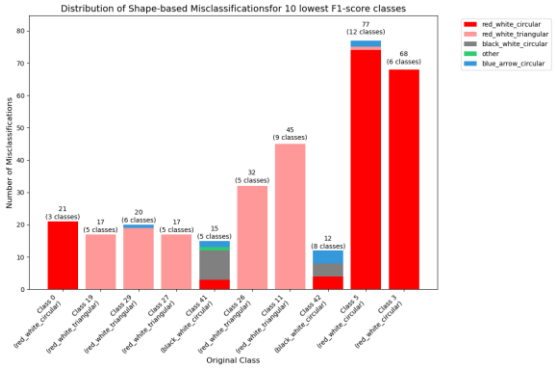

Figure 13 – The misclassified proportion by XGB

The misclassification proportions suggest that the models have learned some coarse shape-colour groupings. However, they fail to distinguish the inner symbolic differences.

For XGB, class 0 is most frequently misclassified as class 1. As indicated in Figure 11, the model prioritises HOG_PCA features which likely encode the information of shape in early splits. However, the features differentiating class 0 and 1 (Figure 14) rank lower in importance. XGB's focus on globally discriminative patterns causes it to overlook localised details. This misclassification stems from the lacks in granularity of current features to encode critical distinctions such as digit strokes or symbol geometry and amplify their significance. Thus, the model's errors reflect the expressive constraints of the features.

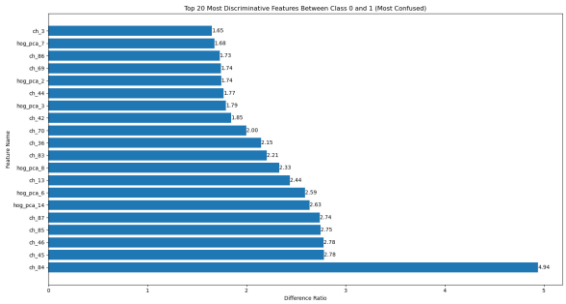

Figure 14 – Top 20 different features between 0 and 1

Figure 15 - Sample of class 0 and 1, showing difference.

Figure 16 – The class-wise RBF similarities of correct samples and wrong samples to support vectors

SVM's misclassifications follow a similar pattern. The RBF kernel similarities which is a measure of proximity to class-specific support vectors shows that misclassified samples exhibit low similarity to the support vectors in true class (Figure 16) comparing to correctly classified samples. This indicates that the feature space lacks the resolution to separate challenging samples, rather than systematic error in SVM itself. These constraints highlight the fundamental bottleneck of machine learning model with insufficiently engineered features, which is omitted in CNN.

4.2 Analysis of CNN

The superior performance of the CNN model is further boosted by architectural design and data augmentation strategies. As demonstrated in Table 2, comparative experiments under different configurations reveal the significant impact of residual connections and data augmentation on model generalisability.

Model Variation	Cross validation average		Holdout set scores		Test set
Metric	Acc	F1	Acc	F1	Acc
No residual block, no augmentation	0.972	0.967	0.985	0.982	0.979
With residual block, no augmentation	0.986	0.982	0.986	0.982	0.987
With residual block, with augmentation	0.974	0.966	0.988	0.990	0.992

Table 2 - The results of CNN under various settings

The baseline CNN, trained without residual blocks or augmentation, performs reasonably well across CV (0.972) and test sets (0.979),

indicating that even a relatively simple architecture can effectively capture local visual patterns.

To further enhance the model's performance, residual blocks were added into the network. The architecture of residual blocks is reference from the Paper by He et al. (2015). As illustrated in their study, the shortcut connections improve gradient flow in deeper networks and enable feature reuse across layers, allowing the model to capture both low-level shapes and high-level symbolic shapes. As reflected in Table 2, this modification boosts CV accuracy to 0.986 and test accuracy to 0.987.

The introduction of data augmentation further improves the test accuracy to 0.992. Note that the CV accuracy slightly decreases (from 0.986 to 0.974), this is reasonable as data augmentation introduces synthetic noise into training data. It discourages the model from fitting overly specific patterns, encouraging more robust and transferable representations. The diversity given by random rotation, scaling, and brightness changes helps the model generalize to various real-world conditions such as viewpoint shifts, illumination differences, and partial occlusion.

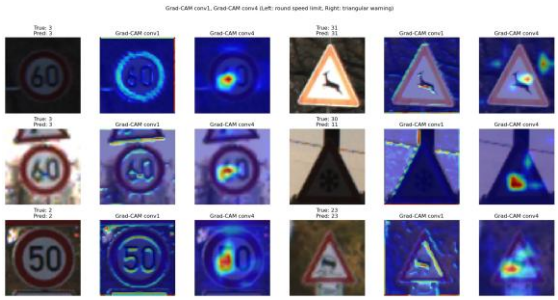

Figure 17 - Grad-CAM on samples showing model focus

To better understand the feature learning process, Grad-CAM visualisations are applied. As shown in Figure 17, the model's attention focuses on the outer contours of traffic signs (e.g., circular speed limit sign edges and triangular warning sign edges) in shallow convolutional layers, mirroring the shape represented by features like HOG_PCA. Furthermore, in deeper layers with residual blocks, the model progressively shifts focus to internal discriminative details (e.g., curvature differences between digits "50" and "60"). Residual connections support this transition by preserving spatial information from shallow layers while enabling deeper layers to learn abstract symbols. This hierarchical and adaptive feature extraction capability gives CNN a clear advantage over classical methods such as XGB and SVM in image recognition tasks.

In terms of error analysis of CNN model, it only misclassifies around 2% of the samples in CV. These wrong samples are hard cases which even hard to classify manually as shown in Figure 18. One thing they share is the low resolution and brightness. Therefore, the model can be further tuned with appropriate augmentation focusing on the brightness and resolution to further improve the score.

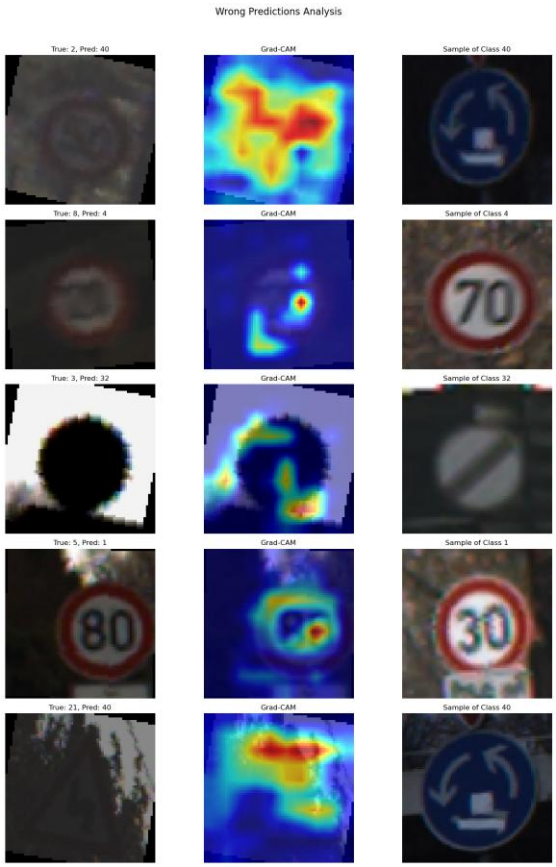

Figure 18 – Wrong samples compare predicted samples

5. Conclusion

The experimental study on the GTSRB dataset shows that CNN models added with residual connections and data augmentation significantly outperform machine learning models like XGB and SVM in image

classification tasks. While the latter offer strong interpretability, their reliance on the quality of engineered features limits their ability to capture local visual structures, reducing generalization under distribution shifts. In contrast, CNN automatically extracts multi-level visual representations, resulting in greater robustness. Future improvements may focus on targeted augmentation for low-resolution and low-brightness samples to further enhance performance.

6. Reference

He, K. *et al.* (2015) *Deep residual learning for image recognition*, *arXiv.org*. Available at: https://arxiv.org/abs/1512.03385 (Accessed: 28 May 2025).