Traffic Sign Detection

Seeing the Signs for Safer Roads

Annu Kumari
2021312

Anand Vimal
2021130

Abstract

In this project, we investigated machine learning for traffic sign detection (TSD) to enhance road safety. We explore various techniques on the GTSRB dataset. Convolutional Neural Networks (CNNs), especially VGG16, in which we achieved the highest accuracy of 96.9% due to its ability to learn features directly from images. Analysis of precision, recall, and F1 scores revealed trade-offs between different models. VGG16 is a promising choice for TSD, with further exploration possible for other models.

1. Problem statement

Human error in recognizing or reacting to traffic signs is a significant contributing factor to traffic accidents which has become a major global concern. In this project, we aim to address this issue by exploring different machine learning (ML) models for traffic sign detection (TSD). By developing an accurate TSD system, we can improve road safety by:

A. *Alerting drivers:* The system can provide real-time notifications about upcoming traffic signs, particularly in situations where a driver might be distracted.

B. *Enhancing autonomous vehicles:* TSD is a crucial component for self-driving cars to navigate roads safely by recognizing and interpreting traffic signals.

2. Literature review

TSD plays an important role in Advanced Driver Assistance Systems (ADAS) and autonomous vehicles. Traditional methods using colour segmentation and shape recognition struggled with real-world variations in lighting, weather, and occlusions. In this review we explored recent advancements in deep learning, particularly Convolutional Neural Networks (CNNs), for robust traffic sign detection.

2.1. *Early Approaches:* Traditional TSD methods relied on handcrafted features like color histograms and geometric properties. Approaches like Support Vector Machines (SVMs) were used for classification. However, these methods lacked the ability to learn complex feature representations, limiting their effectiveness in diverse real-world scenarios [1].

2.2. *The Rise of Deep Learning:* The emergence of deep learning, particularly CNNs, revolutionised TSD. CNNs excelled at extracting hierarchical features from images, making them ideal for recognizing traffic signs despite variations in pose, illumination, and background clutter.

2.3. *Pioneering Works:*

• Ji et al. (2012): Introduced a CNN architecture for TSD, achieving promising results compared to traditional methods [2].

• LeCun et al. (1998): Paved the way for CNNs with their pioneering work on MNIST handwritten digit recognition [3].

2.4. *Recent Advancements:*

Redmon et al. (2016): Introduced You Only Look Once (YOLO), a real-time object detection system achieving high accuracy and frame rates, making it suitable for practical TSD applications [4].

He et al. (2017): Proposed Mask R-CNN, a deep learning architecture that performs object detection, segmentation, and pose estimation simultaneously. This capability could be valuable for TSD in future applications requiring additional sign information beyond class labels [5].

2.5. *Challenges and Future Directions:* Despite significant progress, challenges remain.

• *Occlusions:* Vehicles, foliage, or other objects can partially or completely obscure traffic signs. Research on techniques for handling occlusions effectively is ongoing.

• *Real-time Performance:* Balancing accuracy with computational efficiency is crucial for real-time TSD applications in autonomous vehicles. Exploring lightweight CNN architectures and hardware acceleration techniques are promising avenues.

3. Dataset details

We utilized the widely used GTSRB (German Traffic Sign Recognition Benchmark) dataset [6]. This dataset contains over 50,000 images of 43 traffic signs with lighting, pose, and occlusion variations. The dataset is split into training, validation, and testing sets for model development and evaluation. And we also used the Traffic sign dataset.

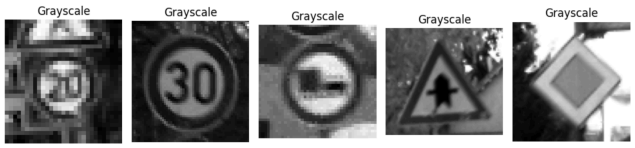
4. Data Preprocessing

Visualized the dataset to understand its content. And removed the background color from images.

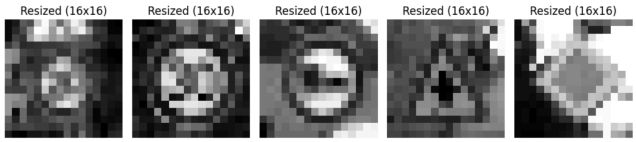
Original coloured images:

Grayscaling and resizing: Converted images to grayscale and resized them to 16x16 pixels to reduce complexity and dimensionality.

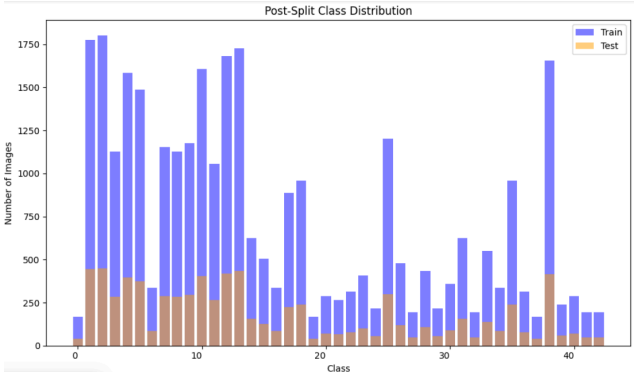
Images after Grascaling :

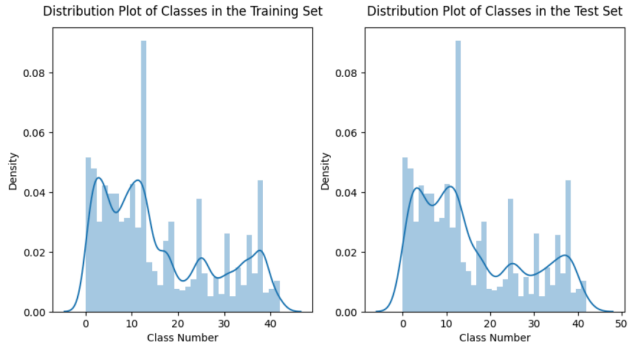


Images after resizing



Data Split: Splitted the data into training (80%) and validation sets (20%) for model evaluation.





For Dimensionality reduction:

We tried to use autoencoder for dimensionality reduction but accuracy was not that permissible because:

1. Autoencoder is overfitting or the architecture isn't ideal.
2. PCA captured the key structures (likely linear) for classification.
3. Non-linear information learned by an autoencoder might not be relevant.

PCA fitted very well for reducing the dimensions.

5. Model Classification

We explored different classification techniques for a GTSRB dataset initially. Here's a breakdown of why these techniques were applied and how the results align with their properties:

5.1 Applied *Linear discriminant analysis(LDA)* and *Quadratic discriminant analysis(QDA)*. Achieved accuracy of 71% and 64%, respectively.

These techniques were chosen for initial exploration due to their strengths in working with lower-dimensional data and interpretability. LDA, in particular, is a good starting point as

Linear Decision Boundaries: LDA assumes data classes can be separated by linear hyperplanes. This makes it efficient for computation and helps avoid overfitting, especially with limited data, as we observed with 71% accuracy.

Interpretability: LDA provides insights into the relationships between features and class separation.

While QDA offers more flexibility with quadratic boundaries, it can be prone to overfitting, particularly with limited data. This seems to be the case here, as the model achieved a lower accuracy (64%) compared to LDA.

5.2 *Random Forest, Logistic Regression, KNN*

Given the limitations of LDA and QDA with potentially

limited data, these techniques were likely chosen for xfurther exploration due to their

Higher Accuracy Potential: Random Forest (96%), KNN (90%), and Logistic Regression (88%) have the potential for higher accuracy compared to LDA and QDA, especially when dealing with complex data patterns.

Overall, the results align with the expected properties of these techniques. LDA and QDA provided a good starting point for understanding the data and exploring basic separability, while Random Forest, KNN, and Logistic Regression offered increased accuracy by potentially capturing more complex relationships within the data.

5.3 Convolutional Neural Networks (CNNs) are image recognition masters. They automatically learn features (edges, shapes) directly from images and don't rely on predefined ones. But CNNs are smarter than that - they also analyze how these features connect in space. This helps them understand an object's structure, like the difference between a cat and a dog, based on eye and ear positions. CNNs handle the complexity of real images with non-linearity, allowing them to make accurate predictions. CNNs trump Random Forests and KNN because

- CNNs learn features automatically, ditching manual effort.
- CNNs grasp how pixels connect, which is crucial for object recognition.
- CNNs handle complex image patterns better with non-linearity.

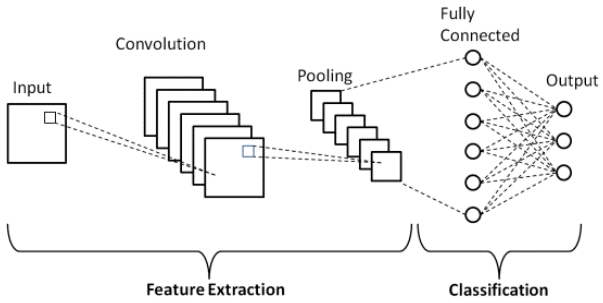


Fig: CNN Architecture

While KNN and Random Forests struggle with images for two main reasons:

- **Missing the picture:** They rely on pre-defined features, which might not capture the key aspects of images (like spatial relationships).
- **Linear limitations:** They primarily model linear relationships, while real-world images have complex, non-linear patterns.

5.4 *Visual Geometry Group (*VGG16), a powerful CNN architecture, takes a unique approach to image recognition. Instead of a few complex filters, it relies on many small 3x3 filters with stride 1, gradually extracting features from simple to complex. Consistent 2x2 max pooling layers throughout the network downsample data and capture key features. This repetitive structure with simple filters, though resulting in a large network with 138 million parameters, has proven effective for image recognition. Finally, fully connected layers combine the learned features and a softmax output layer predicts the image class.

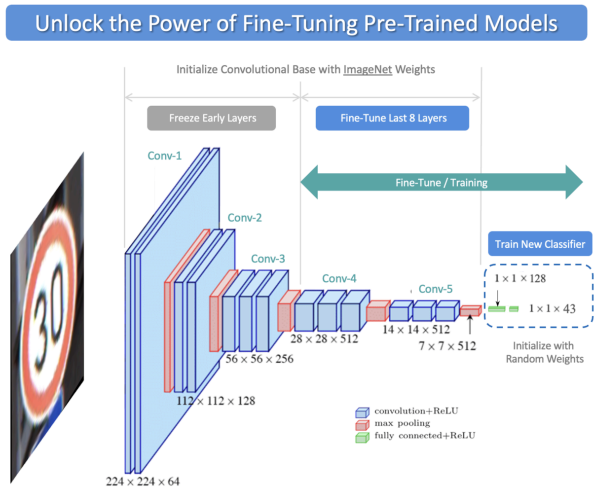


Fig: VGG16 Architecture

It isn't a completely new idea, it's CNN! But it does same things differently to achieve good results:

- *Smaller building blocks***:** It uses many small filters instead of a few big ones, making it easier to train.
- *Deeper understanding***:** With 16 layers, it can capture complex features in the images.
- *Smart downsizing:* It shrinks data size efficiently to avoid getting overwhelmed.

These choices make VGG16 a strong foundation for image recognition, although even more powerful CNNs exist today.

6. Results and Analysis:

These are the Accuracy achieved on different models-

Model	Test Accuracy
LDA	71.33%
QDA	64.06%
Random Forest Classifier	97.89%
SVM classifier	60.83%
MLP classifier	85.13%
XG Boost classifier	83.42%
Logistic Regression	96.42%
KNN	90.76%
CNN	95.46%
VGG16	96.9%

6.1 Key Observations:

VGG16 Achieves Highest Accuracy: It emerged as the top performer with a test accuracy of 96.9%. This suggests it effectively learned the patterns within the data and generalizes well to unseen examples.

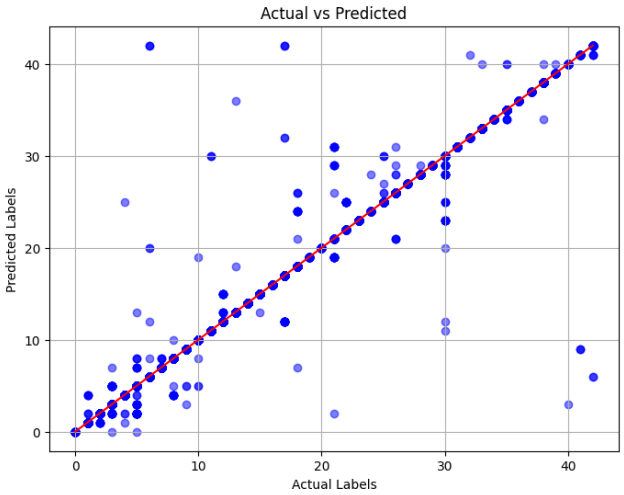


Fig: Confusion Matrix of VGG16 model

CNN Performs Well: While not quite reaching VGG16's accuracy (95.46%), CNN also demonstrates strong performance. Both CNN and VGG16 are promising choices for this task..

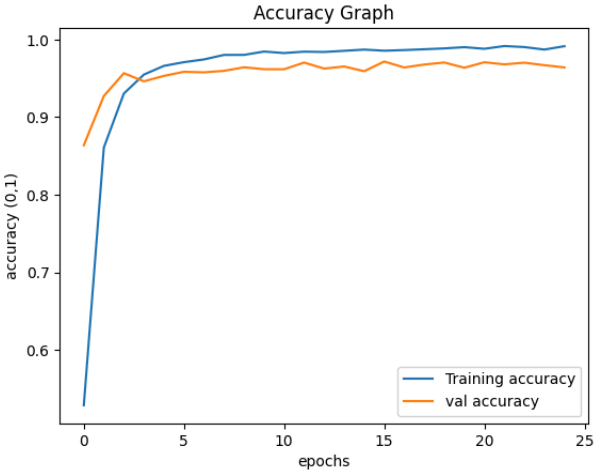


Fig: Epoch vs accuracy graph for CNN Model

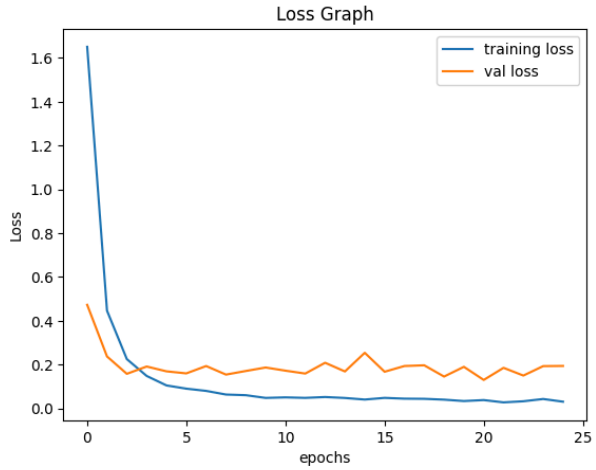


Fig: Loss vs Epoch graph for CNN Model

Random Forest Likely Overfits: Despite boasting a high test accuracy (97.89%), the significant drop in cross-validation accuracy (62%) suggests the Random Forest model is overfitting. It memorises the training data too closely, hindering its ability to generalise to new data.

Logistic Regression Trades Off Performance: Logistic Regression offers good accuracy (96.42%) but struggles with high-dimensional data, leading to overfitting and high computational cost. Additionally, it requires pre-defined features, which can be challenging to select effectively.

Other Models and Tuning Potential: The remaining models exhibited varying levels of performance. Hyperparameter tuning could potentially improve the accuracy of some, such as SVM and MLP.

6.2 Analysing the precision, recall and f1-score, The inferences we got are as follows-

VGG16, CNN, SVM shines: These models achieved perfect precision (1.0), signifying they correctly identify all positive cases they predict. However, their recall of 0.824 indicates they missed some actual positive cases. The F1-score of 0.904 suggests a reasonable balance between precision and recall.

MLP and XGBoost impress: These models display impressive F1 scores (0.975 and 0.966), indicating a strong balance between precision and recall. They make accurate positive predictions and capture the most true positives

Random Forest is decent: While achieving high precision (0.997), Random Forest suffers from lower recall (0.917), suggesting it misses some true positives.

Room for improvement: AdaBoost and Logistic Regression/KNN require further attention. AdaBoost suffers from low recall (0.435), while Logistic Regression and KNN have lower F1 scores. Tuning these models or exploring different algorithms might be beneficial.

Top performers (balanced): SVM, MLP, XGBoost
Good, but could be better: Random Forest

6.3 Analysing the *mAP-score*, The inferences we got are as follows-

In a comparative analysis of classification models based on *mean Average Precision (mAP)* scores, the *Convolutional Neural Network (CNN)* and *VGG16* models exhibit superior performance, achieving mAP scores of *0.9206* and *0.9030*, respectively. Additionally, the *Random Forest, MLP, and XGBoost* classifiers demonstrate strong performance with mAP scores ranging from 0.9152 to 0.9138. Conversely, models such as *KNN and AdaBoost* perform inadequately, with mAP scores significantly lower than other models. This highlights the efficacy of deep learning models, particularly CNNs, in handling complex classification tasks, while emphasising the importance of selecting appropriate models tailored to specific data characteristics and performance requirements.

7. Prediction

Finally, on uploading the input image. It predicts signs.

Fig: Input sign given

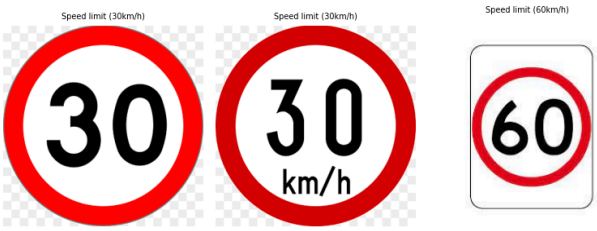


Fig: Predicted sign

8. Conclusion

VGG16 stands out as the best-performing model with exceptional accuracy and generalisation capabilities. CNN presents itself as a strong alternative. Random Forest's high accuracy is likely deceptive due to overfitting. Logistic Regression offers a good balance but requires careful feature selection and might struggle with high-dimensional data.

Further exploration through hyperparameter tuning for SVM and MLP, as well as investigating alternative approaches for AdaBoost, Logistic Regression, and KNN, could lead to additional improvements. Analysing precision, recall, and F1 scores provided valuable insights into the trade-off between correctly identifying positive cases and capturing all true positives. When selecting the most suitable model for the specific task, these factors should be considered.

Individual contributions

Annu Kumari
Dataset Load
Data Preprocessing
VGG16
Image Argumentation
LDA and QDA

Anand Vimal
Dataset Load
Image Segmentation
CNN

Different ML Models Classification
KNN and LR

References

[1] Salcedo‐Sanz, Sancho, José Luis Rojo‐Álvarez, Manel Martínez Ramón, and Gustavo Camps‐Valls. "Support vector machines in engineering: an overview." Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery 4, no. 3 (2014): 234-267.

[2] Alzubaidi, Laith, Jinglan Zhang, Amjad J. Humaidi, Ayad Al-Dujaili, Ye Duan, Omran Al-Shamma, J. Santamaría, Mohammed A. Fadhel, Muthana Al-Amidie, and Laith Farhan. 2021. "Review of Deep Learning: Concepts, CNN Architectures, Challenges, Applications, Future Directions." Journal of Big Data 8 (1). https://journalofbigdata.springeropen.com/articles/10.1186/s40537 021-00444-8.

[3] Lecun, Yann, L Eon Bottou, Yoshua Bengio, and Patrick Haaner Abstract|. 1998. "Gradient-Based Learning Applied to Document Recognition." PROC. OF the IEEE. http://vision.stanford.edu/cs598_spring07/papers/Lecun98.pdf

[4] Redmon, Joseph, Santosh Divvala, Ross Girshick, and Ali Farhadi. 2016. "You Only Look Once: Unified, Real-Time Object Detection." https://www.cvfoundation.org/openaccess/content_cvpr_2016/papers/Redmon_You_Only_Look_CVPR_2016_paper.pdf

[5] DOI: 10.1109/ICCES51350.2021.9489152.

[6] "GTSRB - German Traffic Sign Recognition Benchmark." n.d. Www.kaggle.com.