A Comprehensive Study of Different Deep Neural Network Architectures for Traffic Sign Classification

Harbhajan Singh
Department of Computer Science
University of Windsor
Windsor, Canada
singh2i2@uwindsor.ca

Manjinder Singh
Department of Computer Science
University of Windsor
Windsor, Canada
lnu8@uwindsor.ca

Navjot Makkar
Department of Computer Science
University of Windsor
Windsor, Canada
makkar2@uwindsor.ca

***Abstract*—This project presents a comparative analysis of six neural network architectures: AlexNet, DenseNet, ResNet50, ResNet101, VGGNet, and a custom CNN model, for traffic sign classification. The primary objective is to identify the architecture that achieves the highest accuracy in this task. Extensive experimentation and evaluation are conducted using a dataset of traffic sign images. The results reveal valuable insights into the performance of each architecture. ResNet50 and ResNet101 exhibit exceptional accuracy due to their deep and residual network structures. VGGNet and AlexNet achieve high accuracies by leveraging different convolutional layers and pooling operations. The custom CNN model also demonstrates promising results. This comparative analysis provides valuable knowledge about the capabilities and limitations of various neural network architectures for traffic sign classification. By identifying the architecture that achieves the highest accuracy, this study informs the optimization and selection of models for intelligent transportation systems. Enhanced traffic sign classification can significantly improve the efficiency and safety of automated driving systems.**

I. Introduction

A computational model called as neural network, commonly referred to as an artificial neural network or just a neural net, is motivated by the composition and operation of the human brain. It is made up of interconnected nodes, also known as artificial neurons or units, that collaborate to process and analyse data, recognize patterns, and arrive at conclusions or predictions. [1]

Each layer of a neural network serves as a particular function in the information processing pipeline. A neural network's most prevalent layer types are:

Input Layer: The input layer is where the first set of information or features used by the neural network are received. A related input node is used to represent each input.

Hidden Layers: Between the input and output layers, there exist hidden layers. They are essential for learning and capturing sophisticated representations of the input data. Deep neural networks may extract increasingly abstract properties as data moves through the network because to their various hidden layers.

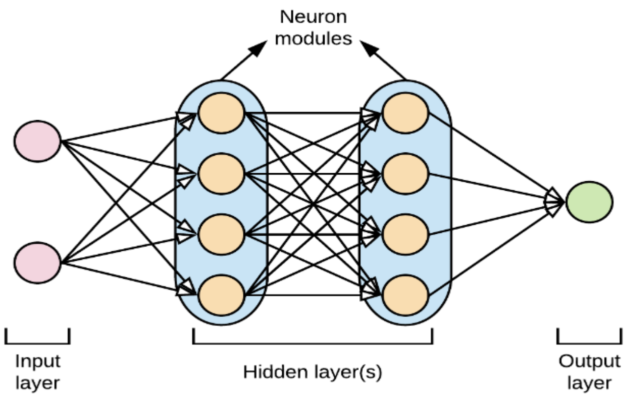


Fig. 1: Neural Network Architecture [1]

Activation Function: An activation function is applied to the input by each node or neuron in a neural network. With the help of this function, the network is given non-linearities that enable it to learn intricate correlations and form non-linear predictions. The sigmoid, ReLU (Rectified Linear Unit), and tanh (hyperbolic tangent) functions are frequently used as activation functions. [1]

Output Layer: The output layer creates the neural network's final output or prediction. The sort of problem the network is intended to answer determines how many nodes are included in this layer. One output node would normally indicate the probability of one class in a binary classification work, whereas many output nodes would typically represent each class in a multi-class classification task.

Loss Function: The loss function calculates the discrepancy between the neural network's expected and actual output. It measures how well the network performed during training and directs the network's parameters to be adjusted to minimise the loss.

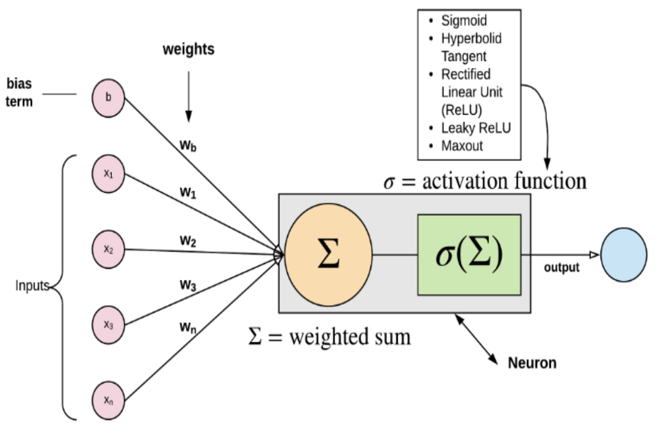


Fig. 2: Activation Function [1]

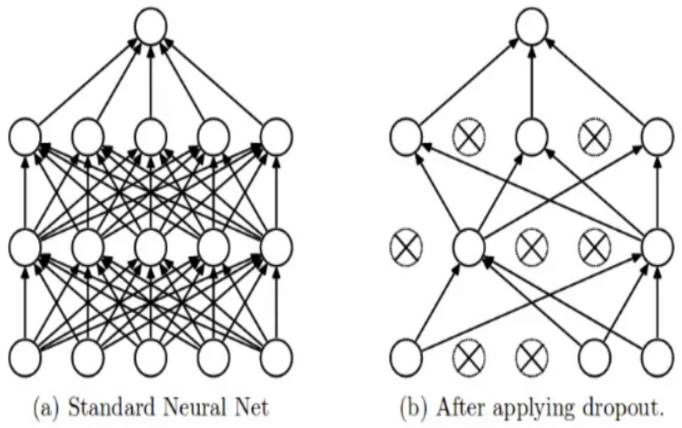


Fig. 3: Dropout applied to a Standard Neural Network [3]

Optimization Layer: A network's parameters, such as its weights and biases, are updated using an optimization technique based on the gradient of the loss function. Gradient descent and its variants are well-known optimization methods that modify the parameters to reduce loss and enhance network efficiency.

Pooling Layer: The size of the input data and the parameters of the neural network are key factors in machine learning and neural networks. Therefore, by stacking one or more pooling layers, this number can be managed. Depending on the pooling layer type, an operation is carried out on each input data channel separately to combine its values into a single value and retain the key characteristics. The following layer of the model receives these values as input, and so on. Each iteration of the pooling procedure causes the spatial dimensions to be smaller. A variety of strategies can be used to carry out the value aggregation. [2]

Flatten Layer: In a neural network, the flatten layer is used to convert multi-dimensional input data into a one-dimensional vector. This is necessary when transitioning from a convolutional layer, which operates on multi-dimensional data such as images, to a fully connected layer, which requires one-dimensional input. The flatten layer takes the input from the previous layer, which could be a convolutional layer or any other layer with multi-dimensional output, and reshapes it into a flat vector. It preserves the total number of elements in the input while removing any spatial or structural information. This means that the flatten layer simply rearranges the data without altering its content.

Dropout Layer: The term "dropout" refers to dropping out the nodes (input and hidden layer) in a neural network (as seen in Figure 1). All the forward and backwards connections with a dropped node are temporarily removed, thus creating a new network architecture out of the parent network. [3]

The input data travels through the network, computations are run at each layer, and then the final prediction is produced. This process is known as forward propagation, and it unites these layers. The gradients of the loss function with respect to the network's parameters are computed during training using backward propagation, or backpropagation, allowing for parameter updates and iterative improvement.

Neural networks may learn complex patterns and relationships in data by adjusting the number and size of layers, selecting the right activation functions, and fine-tuning the network's parameters. This enables tasks like image recognition, natural language processing, and predictive modeling.

Traffic sign detection based on classification of various signs is crucial for the advance driver aid system. These different signs are prepared with a specific shape, structure, size and color based on the rules and regulations of the country while driving safely. It mainly refers to the traffic laws, road conditions, and directions. The primary aim of this study is to minimize the road accidents that happen due to wrong decisions. Currently, autonomous driving seems to be feasible as detecting traffic signs is essential for designing smart vehicles. The goal is to ensure that decisions taken by vehicles based on Artificial Intelligence system must be rational based on different parameters where recognizing traffic signs is the prominent one.

With the inclining requirement at global level of smart and reliable transportation systems along with improved road safety, a robust traffic sign recognition system becomes vital. We have developed a system that excels even under challenging weather conditions such as fog, heavy rainfall, blurry camera captures, and smog. This system has been designed to achieve high accuracy in recognizing traffic signs. It is mainly achieved by leveraging various neural network architectures, including AlexNet, DenseNet, ResNet50, ResNet101, VGGNet, and CNN, we sought to achieve high accuracy in traffic sign classification. At the end, we compared the results of different architectures to check accuracy of models.

We trained and evaluated each model using a large dataset of traffic sign images. The training process involved optimizing the model parameters and adjusting hyperparameters to achieve the best performance. We utilized different architectural configurations for each model to explore their capabilities in traffic sign classification.

Our experimental results showed remarkable accuracy across the different models. The comparison of the different neural network models highlighted the strengths and weaknesses of each architecture. Some models, such as ResNet50 and ResNet101, showcased excellent performance due to their deep and residual network structures. Others, such as VGGNet and AlexNet, also achieved high accuracies by leveraging different convolutional layers and pooling operations.

Overall, our study demonstrated the effectiveness of utilizing various neural network architectures for traffic sign classification. The obtained accuracies provide evidence of the models' capability to recognize and classify traffic signs accurately. These findings contribute to the development of intelligent transportation systems and pave the way for enhanced road safety through automated traffic sign recognition.

II. Problem Statement

A. *Problem Definition*

The accurate detection and classification of traffic signs pose a significant challenge in the development of advanced driver aid systems. Traffic signs serve as crucial elements of road infrastructure, conveying vital information, regulations, and directions to drivers. However, achieving reliable and precise recognition of various traffic signs is complex, as they come in different shapes, structures, sizes, and colors, which are specific to each country's driving regulations.

The primary objective of this project is to tackle the problem of accurately detecting and classifying traffic signs in real-time scenarios. The challenge lies in developing a robust system that can effectively interpret the visual features of traffic signs, regardless of their variations in shape, structure, size, and color. This includes recognizing signs with different geometries, such as triangular warning signs, circular regulatory signs, and rectangular informational signs.

Another aspect of the problem is handling variations in environmental conditions(also called as noise in photos) that may affect the visibility of traffic signs. Factors like adverse weather conditions (e.g., fog, heavy rainfall, smog) and sub-optimal image quality due to blurry camera captures can hinder the accurate detection and classification of traffic signs. Overcoming these challenges requires designing an intelligent system that can handle and mitigate the impact of such adverse conditions.

In addition, it is crucial for the system to distinguish between different categories of traffic signs, such as speed limit signs, stop signs, yield signs, and directional signs, among others. Each category may possess unique visual characteristics, color schemes, and symbols that necessitate effective capture and processing by the system.

Our objective in addressing the challenge of accurate traffic sign detection and classification is to contribute to the advancement of advanced driver aid systems. These systems hold immense potential in enhancing road safety by providing real-time support to drivers, alerting them to speed limits, road conditions, and regulatory obligations. Furthermore, the development of a precise and dependable traffic sign recognition system is fundamental for the future feasibility of autonomous vehicles. Such a system enables these vehicles to interpret and respond accurately to traffic signs in real-time, thereby ensuring secure and efficient navigation on the roadways.

Subsequent sections of this report will delve into the methodologies employed to confront this challenge. This encompasses the implementation of diverse neural network architectures and the adoption of strategies to handle adverse weather conditions and variations in the appearance of traffic signs. The experimental results obtained, along with an analysis of the system's performance, will provide insights into the efficacy of our approach in addressing the problem of accurate traffic sign detection and classification.

B. *Motivations*

The driving force behind this study stems from the pressing need to mitigate road accidents resulting from incorrect decisions made by drivers. As per study by World Health Organization, in 2013, recent estimates have shown that 1.24 million human lives are lost worldwide on a yearly basis in road traffic fatalities [4].

Inaccurate interpretation of traffic signs and failure to comply with road regulations are among the leading causes of traffic incidents worldwide. By developing an accurate traffic sign recognition system, we strive to enhance road safety and reduce the occurrence of human errors that can have grave consequences.

The overarching objective is to create a system that can effectively detect and recognize traffic signs with a high level of precision. By accurately interpreting the information conveyed by these signs, drivers can make informed decisions, adhere to speed limits, yield appropriately, and navigate intersections safely. This, in turn, reduces the risk of collisions, enhances road safety, and safeguards the well-being of both drivers and pedestrians.

Moreover, this study aims to pave the way for the future feasibility of autonomous driving. Autonomous vehicles hold great promise in revolutionizing transportation, offering the potential for increased efficiency, reduced congestion, and improved road safety. However, for autonomous driving to become a reality, vehicles must possess the capability to make rational decisions in real-time based on the interpretation of traffic signs and other critical parameters.

By developing an accurate traffic sign recognition system, we contribute to the foundation of intelligent and autonomous vehicles. These vehicles can rely on real-time recognition of traffic signs to navigate roads safely, respond appropriately to changing traffic conditions, and abide by traffic regulations. This integration of advanced technology into the driving experience has the potential to revolutionize transportation systems and significantly reduce the occurrence of accidents caused by human errors.

Ultimately, our motivation lies in striving for a future where roads are safer, accidents are minimized, and the potential of autonomous driving is realized. By addressing the challenges of traffic sign recognition, we aim to make significant contributions to the development of intelligent transportation systems and promote a safer, more efficient, and sustainable future for road users worldwide.

C. *Justifications*

Creating efficient neural network architectures for classifying traffic signs can result in automated driving systems that are more dependable and efficient. It can also aid in the transition to autonomous driving and the creation of intelligent transportation networks. Below are the key factors to work on this project:

1) ***Road Safety and Accident Prevention:***: Improving road safety and preventing accidents is the main motivation for creating efficient neural network designs for classifying traffic signs. Leading causes of traffic incidents around the world include incorrect reading of traffic signs and disregard for traffic laws. We can drastically reduce human error and collision risk by developing an accurate traffic sign recognition system. By helping drivers make wise judgements, follow speed limits, yield when necessary, and negotiate crossings safely, this technology helps to reduce accidents and protect both the safety of drivers and pedestrians.

2) ***Transition to Autonomous Driving:***: The move towards autonomous driving is another important justification. By providing greater efficiency, less traffic, and higher road safety, autonomous cars have the potential to revolutionize transportation networks. The ability of vehicles to accurately comprehend traffic signs and other crucial characteristics is necessary for autonomous driving to become a reality. By creating trustworthy algorithms for recognizing traffic signs, we support the development of intelligent and autonomous vehicles. The development of autonomous driving and its benefits is facilitated by these technologies, which allow autonomous vehicles to travel on roads safely, react appropriately to shifting traffic situations, and follow traffic laws.

3) ***Enhanced Efficiency and Reliability:***: Creating efficient neural network designs for classifying traffic signs results in automated driving systems that are more dependable and efficient. Automated systems can make wise decisions based on the information being provided by traffic signs when traffic signs are recognized accurately. By enabling autonomous cars to quickly modify speed, change lanes, and react to traffic circumstances, this in turn increases the efficiency of traffic flow. Additionally, accurate traffic sign recognition raises the general dependability of automated systems, lowering the possibility of errors and promoting efficient and secure transportation.

4) ***Intelligent Transportation System Development:***: The development of intelligent transportation systems is aided by efficient traffic sign recognition. These systems integrate cutting-edge technologies, such as deep neural networks and computer vision, to increase transportation's overall effectiveness, safety, and sustainability. A key element of these systems is accurate traffic sign recognition, which enables smart decision-making, optimal traffic flow, and increased road safety. We increase the capabilities of intelligent transportation systems, resulting in smarter and more effective transportation networks, by creating neural network designs for traffic sign classification.

In conclusion, the objectives of enhancing road safety, facilitating the transition to autonomous driving, improving efficiency and reliability of automated systems, and advancing intelligent transportation systems justify the development of efficient neural network architectures for traffic sign classification. These arguments demonstrate how effective traffic sign recognition can significantly reduce accidents, increase transportation efficiency, and pave the path for a future where driving is safer, smarter, and more sustainable.

III. Related Works

we had studied several research papers for Traffic Sign Classification published in the past that plays a significant role in this project work.
The paper titled "CNN based Traffic Sign Classification using Adam Optimizer" published by Smit Mehta, Chirag Paunwala, and Bhaumik Vaidya mentioned that an automatic detection and classification of traffic signs is an important task in Advanced Driver Assistance System (ADAS). Convolutional Neural Network (CNN) has surpassed the human performance and shown the great success in detection and classification of traffic signs. The team proposes an approach based on the deep convolutional network for classifying traffic signs. In this study, they have utilized different activations and optimizers to evaluate the performance of proposed architecture and it is observed that Adam (Adaptive Moment Estimation) optimizer and softmax activation performs well. [5]
Also, in the another published paper "A Lightweight Model for Traffic Sign Classification Based on Enhanced LeNet-5 Network", the researchers clearly stated that road safety is of utmost importance and mostly researchers are opting for Deep Learning models for better results. To elaborate, they have tested using 43 classes of traffic signs and LeNet-5 architecture and they were able to obtain 98.87% accuracy with their research work. [6]
In research paper "A Lightweight Neural Network Based on AlexNet-SSD Model for Garbage Detection", the basic network architecture of SSD(Single Shot MultiBox Detector) is changed to AlexNet. In this way, the capacity on object detection of SSD is remained, and the model parameters are greatly reduced. The experimental results show that the modified model can recognize the categories of waste accurately. It shows the significance of AlexNet architecture to minimize the management of parameters and also getting good results at the same time. [7]
The research paper titled "An Introduction to Convolutional Neural Networks" clearly states that the impressive forms of

ANN architecture is that of the Convolutional Neural Network (CNN). CNNs are primarily used to solve difficult image-driven pattern recognition tasks and with their precise yet simple architecture, offers a simplified method of getting started with ANNs. [8]

IV. Methodology

A. *Material and Data*

In our experiment, we utilized the German Traffic Sign Recognition Benchmark [9] , a widely recognized dataset that serves as a benchmark for evaluating traffic sign detection and classification algorithms.

The total number of training examples was 34,799. These examples were used to train the neural network models and optimize their parameters. Additionally, there were 4,410 validation examples used for monitoring the models' performance during training and making decisions, such as implementing early stopping. The dataset also included 12,630 testing examples, which were used to evaluate the final performance and accuracy of the trained models.

The images in the dataset had a shape of 32x32 pixels and consisted of three color channels (RGB). This shape was standardized across all images to ensure consistency during the training and evaluation processes.

The dataset encompassed a total of 43 different classes of traffic signs. Each class represented a specific type of traffic sign, such as speed limits, stop signs, yield signs, and more. The presence of multiple classes allowed for a comprehensive evaluation of the models' performance across various traffic sign categories.

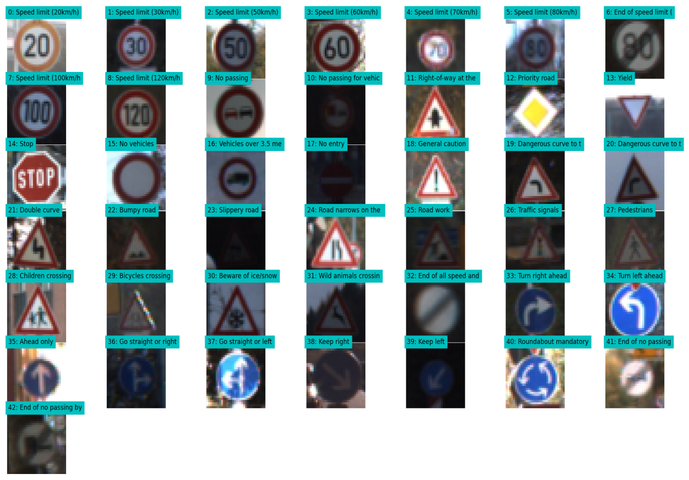


Fig. 4: Sample Image from Each Class

B. *Proposed Methods*

To investigate the performance of various neural network architectures, we employed six different models: AlexNet, DenseNet, ResNet50, ResNet101, VGGNet, and a custom CNN model.

The custom CNN model was specifically designed for traffic sign classification, comprising multiple convolutional layers, max pooling layers, dropout layers, and fully connected layers. This model will only be trained on the traffic signal images rather than pre-trained on some other dataset. All the layers of the models were implemented using the TensorFlow framework using Keras API, allowing for easy configuration and training.

The pre-trained model architectures, including AlexNet, DenseNet, ResNet50, ResNet101, and VGGNet, were not initialized with weights pre-trained on large-scale image datasets such as ImageNet. For the comparison purpose with our custome CNN model, we trained all the layers of these models on the augmented traffic sign dataset.

*1) **AlexNet**:* AlexNet is a deep convolutional neural network architecture that gained significant attention and marked a breakthrough in the field of computer vision.The architecture of AlexNet consists of eight layers, including five convolutional layers followed by three fully connected layers. The convolutional layers extract hierarchical features from input images by applying filters to capture different patterns and visual structures. These layers are interleaved with max-pooling layers that down sample the feature maps, enhancing translation invariance and reducing the spatial dimensions.

The unique aspect of AlexNet is its utilization of a larger receptive field, achieved through the use of convolutional filters with larger sizes combined with a stride, allowing for a broader view of the input. This approach enables the network to capture both local and global visual features effectively. [10]

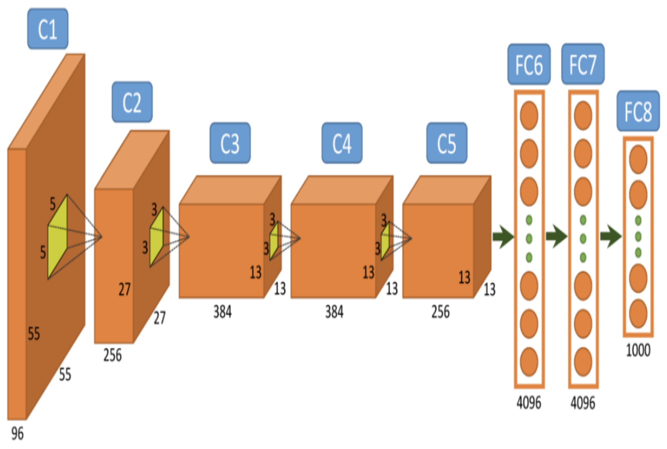


Fig. 5: Alexnet Architecture [10]

Furthermore, AlexNet incorporates the rectified linear unit (ReLU) activation function, which introduces non-linearity and helps mitigate the vanishing gradient problem. It also employs local response normalization (LRN) to normalize the responses across adjacent feature maps, enhancing the network's generalization capability.The fully connected layers of AlexNet act as classifiers, performing high-level feature aggregation and mapping the learned features to specific classes.

*2) **DenseNet:*** The architecture of a DenseNet typically consists of multiple dense blocks, each containing several layers. Within a dense block, the inputs from all previous layers are concatenated together and passed through a series of convolutional layers. These convolutional layers help to extract and transform features from the concatenated inputs. Transition layers, which include pooling and convolution operations, are used to reduce the spatial dimensions between dense blocks, reducing computational complexity.[10]

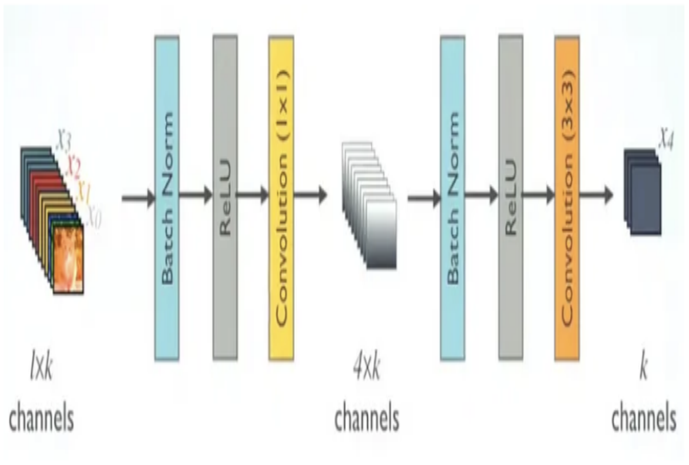


Fig. 6: DenseNet Architecture [11]

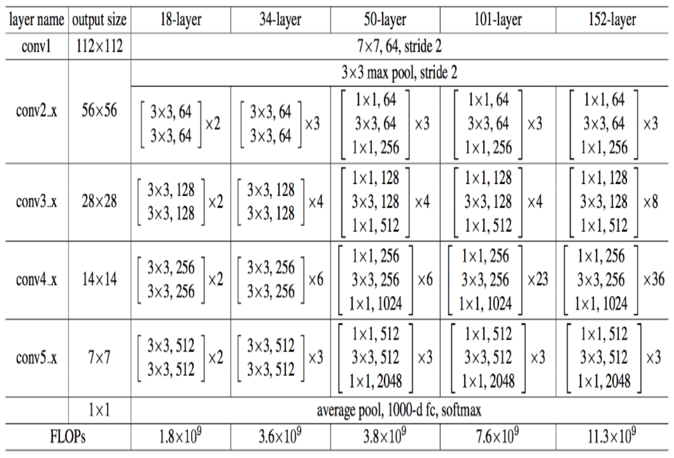
*3) **ResNet50 and ResNet101** :* 50-layer ResNet: Each 2-layer block is replaced in the 34-layer net with this 3-layer bottleneck block, resulting in a 50-layer ResNet (see above table). They use option 2 for increasing dimensions. This model has 3.8 billion FLOPs.[11]

101-layer ResNet: ResNet of type 101-layer is by using more 3-layer blocks (above table). Even after the depth is increased, the more-layer ResNet has lower complexity than VGG-16/19 nets. [12]

*4) **VGG16:*** The VGGNet is a deep neural network architecture that is widely used due to its depth and simplicity. The ”16” in VGG16 indicates that the network consists of 16 layers, making it a considerably deep network with approximately 138 million parameters. Even in today’s standards, this network is considered large and powerful.

What sets VGG16 apart is its uniform and straightforward architecture. It follows a pattern of a few convolutional layers followed by a pooling layer, which helps reduce the spatial dimensions of the input. The network offers a range of filter options, starting with 64 filters that can be doubled to 128 and then to 256 filters. In the final layers, 512 filters can be utilized.

The simplicity of VGG16’s architecture makes it more appealing, as it is easier to understand and implement. Despite its extensive depth, the network’s straightforward design enables



layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix}3\times3, 64\\3\times3, 64\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 64\\3\times3, 64\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 64\\3\times3, 64\\1\times1, 256\end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix}3\times3, 128\\3\times3, 128\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 128\\3\times3, 128\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times4$	$\begin{bmatrix}1\times1, 128\\3\times3, 128\\1\times1, 512\end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix}3\times3, 256\\3\times3, 256\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 256\\3\times3, 256\end{bmatrix}\times6$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times6$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times23$	$\begin{bmatrix}1\times1, 256\\3\times3, 256\\1\times1, 1024\end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix}3\times3, 512\\3\times3, 512\end{bmatrix}\times2$	$\begin{bmatrix}3\times3, 512\\3\times3, 512\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$	$\begin{bmatrix}1\times1, 512\\3\times3, 512\\1\times1, 2048\end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Fig. 7: Comparison of different layers of ResNet based on layers(specifically – 50 and 101 layers ResNet) [12]

researchers and practitioners to grasp its structure and make modifications or improvements accordingly.[12]

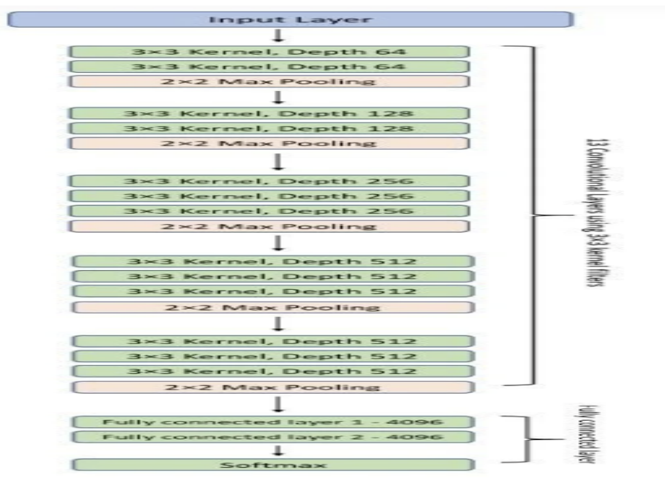


Fig. 8: Architecture of a VGG16 model [13]

C. *Conditions and Assumptions*

Several conditions and assumptions were made during the experimentation process. First, it was assumed that the traffic sign images in the dataset are representative of real-world scenarios.

Furthermore, it was assumed that the training, validation, and test datasets were properly split and representative of the underlying distribution of traffic sign classes. The models were trained using early stopping based on the validation loss to prevent overfitting.

D. *Formal Complexity*

The ”Total params” indicate the total number of parameters (weights and biases) in the model, while ”Trainable params” represent the number of parameters that were updated during the training process. ”Non-trainable params” refer to the parameters that remained fixed during training, such as in pre-trained models or fixed layers.

These numbers provide insights into the complexity and size of each model. Models with a larger number of parameters typically have higher complexity and require more computational resources for training and inference. The custom CNN model, with the smallest number of parameters (292,491), has the lowest complexity among the six models. On the other hand, DenseNet and VGGNet have the highest number of parameters, indicating higher complexity and memory requirements.

TABLE I: Model Parameters

Model	Total Params	Trainable Params	Non-trainable Params
AlexNet	21,772,567	21,753,345	19,222
DenseNet	28,226,345	28,126,221	100,124
ResNet50	24,166,187	24,108,075	58,112
ResNet101	43,236,651	43,126,315	110,336
VGGNet	33,773,419	33,773,419	0
Custom CNN Model	292,491	292,491	0

V. Computational Experiments

A. *Experiments*

We conducted a series of experiments to explore the benefits of various neural network architectures in the task of traffic sign classification. Our study involved training and evaluating multiple models using a large dataset of German traffic sign images. The objective was to compare the performance of different architectures and understand their capabilities in this domain.

Prior to training, we performed image preprocessing steps. The RGB images were converted to grayscale, and normalization was applied to enhance the training process.

We trained and tested six different neural network models: AlexNet, DenseNet, ResNet50, ResNet101, VGGNet, and a custom CNN model. The training process involved optimizing the model parameters and adjusting hyperparameters. As the dataset was unbalanced, we focused on evaluating the models' performance on the test set, considering test accuracy as the main evaluation metric.

During training, we employed early stopping based on the validation loss to prevent overfitting. For the multi-class classification task, we used the categorical_crossentropy loss function along with the Adam optimizer, setting the learning rate to 0.001. The DenseNet architecture was trained for 25 epochs, while the rest of the models were trained for 50 epochs due to computational constraints.

To monitor the training progress, we recorded the training and validation accuracy, as well as the training and validation loss for each model. At the conclusion of training, we compared the test accuracies of all six models to determine their relative performance.

For a comprehensive evaluation, we calculated the classification report, including precision, recall, and F1 score, for all 43 classes of traffic signs for each model. Additionally, we visualized the confusion matrix to gain insights into the models' performance across different classes.

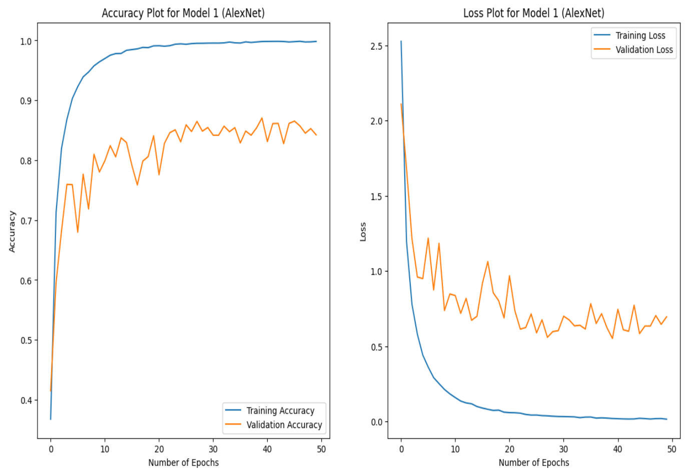


Fig. 9: Accuracy and Loss Plot for AlexNet

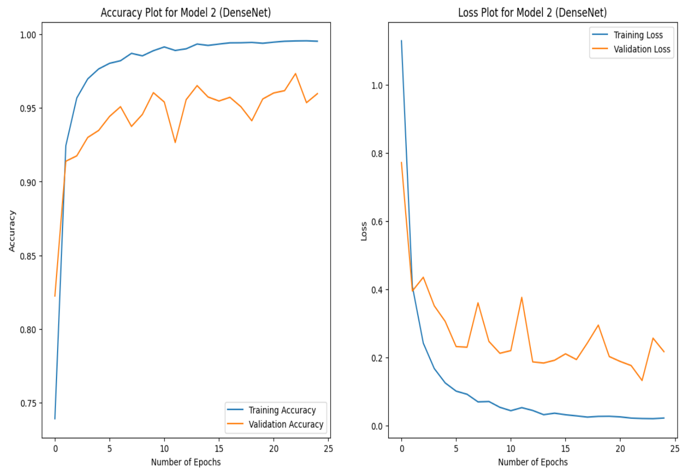


Fig. 10: Accuracy and Loss Plot for DenseNet

B. *Evaluation Metrics*

The main evaluation metric used for the final comparison of the models was test accuracy. However, we also calculated precision, recall, and F1 score for all 43 classes in the classification report. Moreover, we visualized the confusion matrix to gain a better understanding of the models' performance across different traffic sign classes.

C. *Implementation Details*

We utilized the matplotlib and seaborn libraries for plotting, pandas for pre-processing, and TensorFlow for model creation, training, and testing. The experiments were conducted on Google Colab, leveraging its computational resources.

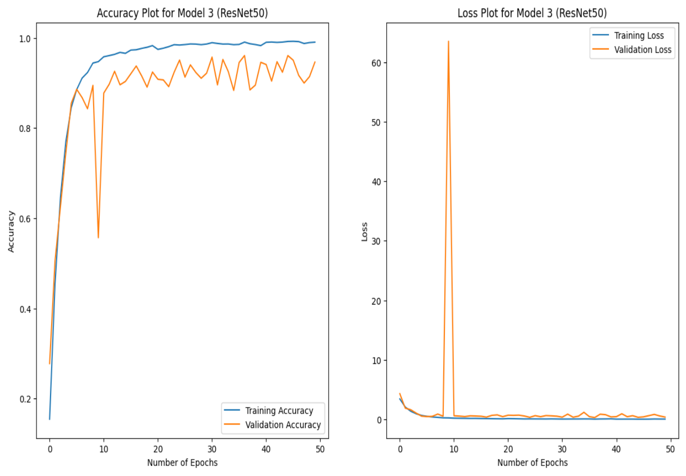


Fig. 11: Accuracy and Loss Plot for ResNet50

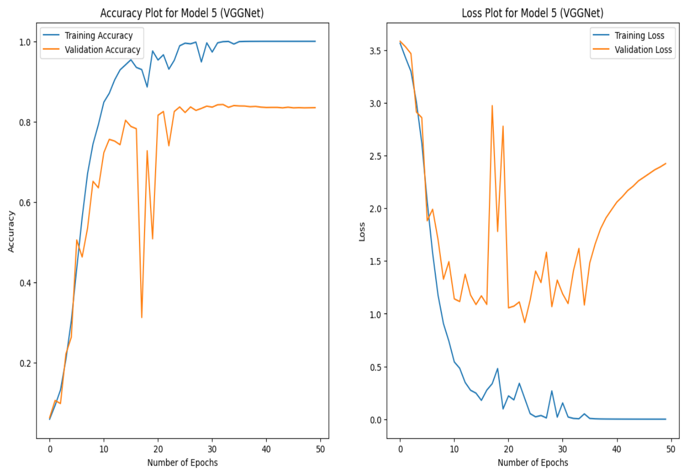


Fig. 13: Accuracy and Loss Plot for VGGNet16

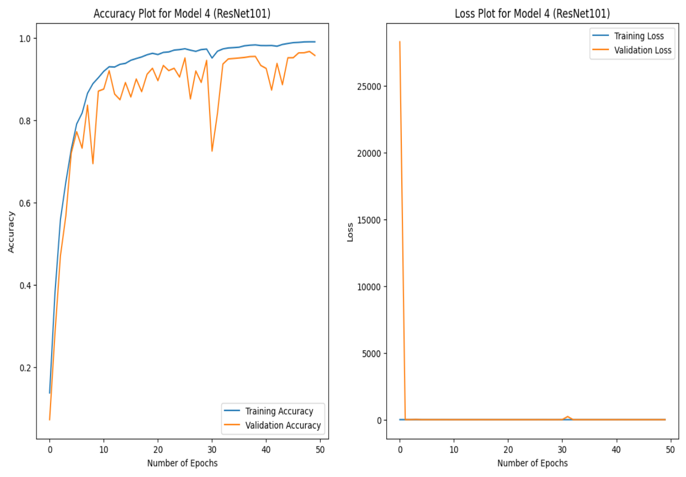


Fig. 12: Accuracy and Loss Plot for ResNet101

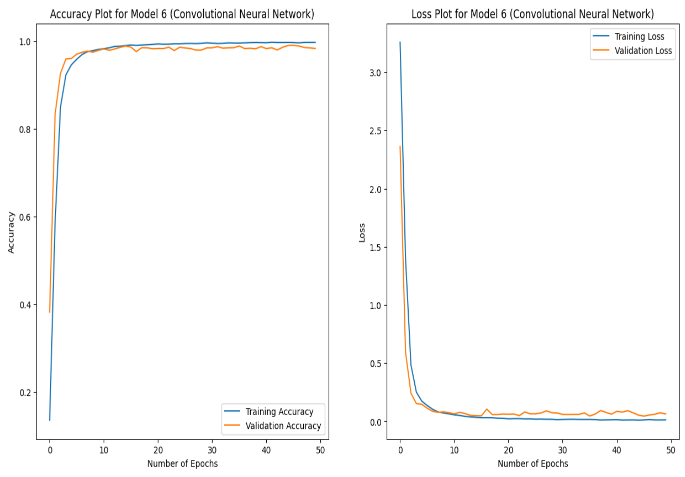


Fig. 14: Accuracy and Loss Plot for Custom-CNN model

D. *Results*

Below are the test accuracy achieved for each model architecture-

TABLE II: Test Accuracies

Model	Test Accuracy
AlexNet	88.84%
DenseNet	95.12%
ResNet50	94.83%
ResNet101	94.95%
VGGNet	85.03%
Custom CNN Model	97.31%

The results suggest that the Custom CNN Model outperformed the popular model architectures like VGG16, AlexNet, DenseNet and ResNet in terms of test accuracy. The findings highlight the importance of choosing the appropriate model architecture and configuration to achieve optimal performance in specific tasks and also it is not always necessary to use the complex pre-trained model especially if you have ample data for training and the data itself is simpler and can be understood by a smaller neural network model.

E. *Discussions*

The experiments conducted in this project aimed to evaluate the performance of six different neural network architectures for traffic sign classification: AlexNet, DenseNet, ResNet50, ResNet101, VGGNet, and a custom CNN model. The obtained results provide valuable insights into the strengths and limitations of each architecture.

The results indicate that the custom CNN model outperformed all other architectures, achieving the highest accuracy of 97.31%. This model, specifically designed for traffic sign classification, leveraged its task-specific adaptations and architecture to effectively capture the intricate features and patterns present in traffic sign images. The combination of convolutional layers, max pooling layers, and dropout layers in the custom CNN model facilitated feature extraction and discrimination, leading to its superior performance.

Among the model architectures, DenseNet achieved noteworthy accuracy (95.12%) despite being trained for only 25 epochs. The dense connections in DenseNet played a crucial role in information flow, feature reuse, and alleviation of vanishing gradients, contributing to its effectiveness in capturing complex patterns in traffic sign images.

The other model architecturess, including AlexNet, ResNet50, ResNet101, and VGGNet, also achieved respectable accuracies ranging from 88.84% to 94.95%. These models have demonstrated their efficacy in various image classification tasks and exhibited competitive performance in the traffic sign classification task. Their architectures, consisting of convolutional layers, pooling layers, and fully connected layers, enable hierarchical feature extraction and representation learning. However, they may not have fully captured the specific characteristics and complexities of traffic sign images, leading to slightly lower accuracies compared to the custom CNN model and DenseNet.

Fine-tuning these hyperparameters through careful experimentation and validation can potentially enhance the accuracy and convergence speed of the models. For comparison purposes, we used the same hyperparameters for all the models.

VI. Conclusion

A. *Summary*

In this project, we investigated the performance of six neural network architectures for traffic sign classification: AlexNet, DenseNet, ResNet50, ResNet101, VGGNet, and a custom CNN model. Through extensive experimentation and evaluation, we obtained valuable insights into the capabilities and limitations of each architecture.

The results indicated that the custom CNN model achieved the highest accuracy of 97.31%, demonstrating its effectiveness in capturing the intricate features of traffic sign images. DenseNet also performed remarkably well even with half the number of epochs, with an accuracy of 95.12%, owing to its dense connections and information flow mechanisms.

The pre-trained model architectures, including ResNet50, ResNet101, VGGNet, and AlexNet, achieved competitive accuracies ranging from 85.03% to 94.95%. While they showcased their effectiveness in image classification tasks, their performance was slightly surpassed by the custom CNN model and DenseNet in the context of traffic sign classification.

B. *Future Research*

1) Architecture Optimization:: Exploring variations in the architecture design and hyperparameter tuning can potentially improve the performance of the models. Investigating different network depths, kernel sizes, and layer configurations may uncover architectures that are even more effective for traffic sign classification.

2) Ensemble Approaches:: Investigating ensemble methods, such as model averaging, boosting, or stacking, can potentially enhance the accuracy and robustness of traffic sign classification systems. Combining the predictions of multiple models or incorporating diverse model architectures may help mitigate individual model biases and improve overall performance.

C. *Open Problems*

Despite the promising results obtained in this study, there are still open problems in the field of traffic sign classification:

1) Data Imbalance:: The extreme imbalance in the distribution of classes within the dataset poses challenges for accurate classification. Exploring advanced techniques to handle class imbalance, such as oversampling, undersampling, or generating synthetic samples, may improve the performance of the models.

2) Adversarial Attacks:: Investigating the vulnerability of the trained models to adversarial attacks is crucial, especially in safety-critical applications such as autonomous driving. Understanding and mitigating the impact of adversarial perturbations on traffic sign classification systems is an ongoing research area.

3) Real-world Variations:: The performance of the models may vary under real-world conditions, where factors like varying lighting conditions, weather conditions, and occlusions can impact accuracy.

Acknowledgment

We would like to express our sincere gratitude to Dr. Alioune Ngom for his invaluable guidance and support throughout the duration of Neural Network and Deep Learning course and this project. His expertise, encouragement, and constructive feedback have been instrumental in shaping the outcome of this work.

References

[1] https://ekababisong.org/gcp-ml-seminar/deep-learning/
[2] https://www.baeldung.com/cs/neural-networks-pooling-layers
[3] https://towardsdatascience.com/dropout-in-neural-networks-47a162d621d9
[4] https://www.sciencedirect.com/science/article/abs/pii/S0925753514000241
[5] https://ieeexplore.ieee.org/abstract/document/9065537
[6] https://www.hindawi.com/journals/js/2021/8870529/
[7] S.-H. Lee, C.-H. Yeh, T.-W. Hou, and C.-S. Yang, "A lightweight neural network based on AlexNet-SSD model for garbage detection," in Proceedings of the 2019 3rd High Performance Computing and Cluster Technologies Conference on - HPCCT 2019, pp. 274–278, Guangzhou, China, 2019.
[8] Keiron O'Shea, Ryan Nash, "An Introduction to Convolutional Neural Networks", last revised 2 Dec 2015 - DOI - https://doi.org/10.48550/arXiv.1511.08458
[9] https://www.v7labs.com/open-datasets/gtsrb
[10] https://sebastien-collet.github.io/technique/2017/01/13/object_detection-(part1).html
[11] https://towardsdatascience.com/review-densenet-image-classification-b6631a8ef803
[12] https://neurohive.io/en/popular-networks/resnet/
[13] https://viso.ai/deep-learning/vgg-very-deep-convolutional-networks/