CENTRAL ASIAN JOURNAL OF MATHEMATICAL THEORY AND COMPUTER SCIENCES

https://cajmtcs.casjournal.org/index.php/CAJMTCS

Volume: 07 Issue: 02 | April 2026 ISSN: 2660-5309

.

Article

Traffic Sign Recognition Using Convolutional Neural Networks with a Tkinter-Based Interactive Interface

Zahraa Mousa Saad

1. College of administration and Economics, University of Misan, Amarah, Iraq

* Correspondence: Zahraa.m.s@uomisan.edu.iq

Abstract: Balancing between innovation in Artificial Intelligence and Cybersecurity is crucial for ensuring information security and successful modern technological applications in Iraq. As it achieving these objectives will contribute in building a secure cyber environment supporting innovation and achieves sustainable development in Iraq. And the importance of research in enhancing cybersecurity in Iraq and highlights to future challenges and the importance of continuing research in this field. So, previous studies have shown that Iraq faces major challenges in the field of cybersecurity, especially with increasing use of artificial intelligence technologies. And reflect the results the need to develop comprehensive strategies enhance from security effectiveness with a focus on: 1) Cooperation between the public sectors and private sectors: to enhance mutual understanding between government institutions and private companies around the importance of cybersecurity. 2) Training human resources: from through educational programs targeted to ensure the availability of specialists in these fields. 3) Investment in research and development: to ensure that remain proposed solutions for facing cybersecurity threats advanced and effective. 4) Developing appropriate legislation: to ensure data protection and strengthen the legal frameworks necessary for investing in artificial intelligence technologies.

Keywords: Real-Time Processing; Rectified Linear Unit; Convolutional Neural Networks; SoftMax Activation Function; Stochastic Gradient Descent; Region Convolutional Neural Networks; Support Vector Machine.

Citation: Saad, Z. M. Traffic Sign Recognition Using Convolutional Neural Networks with a Tkinter-Based Interactive Interface. Central Asian Journal of Mathematical Theory and Computer Sciences 2026, 7(2), 1-20

Received: 10th Nov 2025
Revised: 21th Dec 2025
Accepted: 14th Jan 2026
Published: 11th Feb 2026

1. Introduction

A full set of traffic rules and laws makes sure that roads are safe [31]. These laws are meant to control how drivers, pedestrians, and everyone else on the road acts. They talk about things like speed limits, wearing seatbelts, being drunk or high, and rules about who has the right of way. The most important thing about modern transportation is making sure the roads are safe. Recognising traffic signs quickly and correctly is one of the most important things that can be done to make roads safer [59]. As more and more cars are on the road, we need an intelligent system that can quickly and accurately read traffic signs. This project seeks to tackle this significant issue through the application of deep learning. This research primarily employs convolutional neural networks (CNNs) to develop a dependable and highly precise traffic sign recognition system [43]. Convolutional Neural Networks (CNNs) are a specific type of deep learning model that is very good at analysing and processing visual data, especially pictures and videos. They are based on how the human eye works, with layers that do things like find edges, shapes, and more complex features [62]. One thing that sets CNNs apart is that they can show translation invariance, which means they can recognise patterns no matter where they are in an image [27]. This is accomplished by using the same weights in different spatial locations, which lowers the number of parameters and helps the model generalise better.

CNNs have many layers that gradually pull out more abstract and complicated features. This makes them very good at a wide range of computer vision tasks. The technology helps drivers a lot by automatically recognising these signs, which is especially helpful when they are driving in new or difficult situations [50]. A lot of hard work went into putting together a complete dataset with a wide range of traffic signs. Every picture in the dataset is carefully processed ahead of time to make sure that the model can quickly evaluate and understand a wide range of indicators in a variety of settings. The project uses the gathered dataset to train a CNN model using the most advanced deep learning methods. The model can now recognise different types of traffic signs, taking into account differences in size, lighting, and other environmental factors [30]. Accuracy and Performance: The main goal of the project is to get traffic sign recognition to be very accurate and fast. We check how well the model works in real life by putting it through a lot of tests on a set dataset. A graphical interface that is easy to use has been made so that more people can take advantage of this technology. Users can upload pictures of traffic signs using this interface, and the system will automatically sort them and show what the sign means [54]. The main goal of the project is to make driving easier and safer for everyone. The goal of the project is to make drivers more aware of traffic signs and lower the chances of accidents and traffic violations by teaching them how to read and understand them correctly [58]. This project shows how useful deep learning can be in the real world, especially when it comes to controlling traffic and keeping people safe on the road. By properly implementing a system for classifying traffic indicators, we make a lot of progress toward making the road infrastructure safer and more useful [42]. Join us on this journey as we use the newest technology and deep learning to make the roads safer and more useful. Making roads safer is a job that never ends and requires cooperation between governments, law enforcement, businesses, and the public. The main goal is to cut down on car accidents and the costs that come with them, such as injuries, deaths, and lost money. By encouraging road safety, communities can make transport systems safer and more long-lasting for everyone.

Problem Statement

In today's world of advanced technology, road safety is still very important. Traffic-related injuries, deaths, and fatalities are still major threats to people and communities [53]. One of the many things that make roads safer is being able to read traffic signs correctly and quickly. These signs are very important for communication on the side of the road because they give drivers, pedestrians, and other road users important information [34]. However, it can be hard to identify and understand these signs correctly, especially when the weather is bad, the lighting is poor, or there are many things going on in modern cities [49]. This project aims to solve this difficult problem using the latest machine learning methods, especially Convolutional Neural Networks (CNNs). The biggest problem is making a system that can automatically identify and classify traffic signs from pictures and videos that is both reliable and very accurate [41]. The main goal is to make the roads safer by helping drivers understand and follow traffic signs right away. This will lower the chances of accidents, violations, and possible road hazards. There are many different shapes, colours, and sizes of traffic signs [65]. They can be broken, hidden by trash, or hard to see because of the weather.

The hard part is making a model that can accurately recognise and categorise these signs in a variety of situations. The system needs to work in real time for it to be useful and work well. This means that it should quickly process pictures or video frames, find traffic signs, and send the information to the driver without causing any delays that could put safety at risk. The main goal of the project is to make roads safer [26]. The system has to be very safe and dependable so that the information it gives drivers is correct and trustworthy. The project will collect a large dataset of traffic signs, including many different types of common road signs [48]. The dataset's images will all be carefully pre-processed so that the model can quickly analyse and understand them. The "Traffic Sign

Classification with Deep Learning" project is very important [37]. By solving the problem of accurately recognising traffic signs, it immediately makes the roads safer. The project's results could lead to fewer accidents, traffic violations, and the costs that come with them. It also promises to give drivers a useful tool for getting around complicated and changing city situations, which will make roads safer and more useful in the end. This problem statement lists the main problems, challenges, goals, and objectives of your traffic sign classification project [60]. It also stresses how important it is to make roads safer and help drivers right away.

Aim of The Project

The main goal of this project is to make a big difference in the field of vehicle safety by using advanced deep learning techniques, especially Convolutional Neural Networks (CNNs), which have the power to change the game [57]. The main reason for this project is to make the roads much safer and give drivers and other road users access to the latest technology. The main goal is to make a smart and very accurate system that can quickly and accurately sort and identify a wide range of traffic signs seen in videos and images [38]. This includes a lot of different signs, like stop signs, warning signs, speed limits, and many more. The system's goal is to be a reliable co-pilot for drivers, giving them quick, accurate, and useful information so they can respond to the signs' messages quickly and safely. By reaching this goal, the project hopes to cut down on traffic violations, make the roads safer, and cut down on the number of accidents [47]. In addition to improving technology, this project is based on a strong commitment to putting the health, safety, and quality of life of people and communities who travel on today's complicated roads at the top of the list of priorities [28]. The end goal is to change how people think about road safety by using deep learning to create a world where driving is not only useful but also, most importantly, safe.

Project Domain

This project is in the middle of image processing and computer vision, and it focuses on transport and road safety. This fast-growing field has many uses for improving traffic management and road safety that are based on the expert analysis and interpretation of visual data, especially photos and videos [52]. Your project is well-positioned to be a leader in this field because it addresses the important issue of correctly recognising traffic signs, which is necessary for safe and effective driving. The technology behind this project is computer vision, which is the ability to teach robots how to understand and interpret visual data [33]. This is important for keeping people safe on the road. It uses Convolutional Neural Networks (CNNs), a cutting-edge deep learning method, to identify and sort many different types of traffic signs. The goal is to give drivers reliable, real-time information about the complicated messages these signs are sending, which can include warning signs, speed limits, stop signs, and more [46]. The project's domain is unique because it is at the crossroads of cutting-edge technology, transportation, and safety. It has the power to change the basic safety assumptions of modern transportation and is very much in line with the ideas of traffic control and road safety. It's not just a technology project; it's a promise to keep people and communities safe and healthy as they navigate the ever-changing web of modern roads [61]. Your project hopes to create conditions that make road travel more efficient while also maintaining the highest standards of safety, dependability, and security by adding more lanes to this field.

Scope of The Project

This project has a wide range of important factors that it will cover. The research delves deeply into advanced deep learning techniques, especially Convolutional Neural Networks (CNNs). To do this, we need to make, use, and improve cutting-edge models that can find and sort a wide range of traffic signals [40]. Real-time processing capabilities add to the project's technical skills by making sure that it can quickly analyse picture and video data and give accurate and immediate interpretations. Also, the project needs careful management of a large and diverse dataset of traffic signs, which includes collecting, curating, and pre-processing the data so that the model can be trained quickly. The main use of the project is to recognise traffic signs. It wants to find and sort a wide range of traffic signs, such as stop signs, warning signs, speed limits, and more [35]. The app's scope expands to give drivers real-time help and teach them how to make decisions while driving when they can quickly and accurately understand these signs. Generalisation is another important part of the project. It lets the system adapt to new traffic signs and different weather conditions, making it more useful and effective. The project's significance extends beyond its technical attributes to encompass its potential and impact. Its main goal is to make the roads safer by reducing accidents, violations, and other dangers [56]. This makes the roads safer for everyone who uses them. The project's potential goes beyond the local level and could have an impact on the development of smart transport systems and their use in communities and civilisations all over the world. It also opens the door for future progress in related areas like smart cities, driverless cars, and more general computer vision applications [63]. This means that the field of transportation and safety technology will continue to change and grow in exciting and important ways.

2. Materials and Methods

Convolutional Neural Networks (CNNs) are deep learning models that are made to work with structured grid data like pictures and videos. CNNs are often used in computer vision tasks like image segmentation, object recognition, and image classification. Convolution is an important step in CNNs. It means putting a small filter (also called a kernel) over the picture that was input [45]. The filter gathers characteristics by adding up the results of multiplying each element by itself. This is done all over the picture to make feature maps. During training, learnable filters in convolutional layers change to recognise different patterns and features in the input data [64]. To make the model non-linear, an activation function (usually REL - Rectified Linear Unit) is applied to each element after each convolution operation [39]. The network needs this non-linearity to learn complicated patterns. Pooling layers are used to keep the most important information while making the feature maps smaller in space. Max-pooling is a common way to pool data. It picks the highest value in a small area of the feature map [55]. Pooling helps lower the amount of work that needs to be done and stops overfitting. After a few convolutional and pooling layers, fully connected (Dense) layers are often added. These layers do the same things that regular neural networks do, where each neurone is linked to every neurone in the layer before and after it [32]. You can use fully connected layers for classification or regression tasks. A Flatten layer is put before the fully linked layers to change the 2D feature maps into a 1D vector. This is necessary because fully connected layers need flat input. To keep from overfitting, dropout layers are used. During training, dropout randomly turns off some neurones so they can't help the network make predictions [51]. This makes the network want to learn stronger features.

The last layer of the CNN makes the network's output. The number of neurones in this layer depends on the task at hand. In image classification, the output layer usually has the same number of neurones as there are classifications. The SoftMax activation function is often used to make class probabilities [36]. The job determines which loss function to use. Categorical cross-entropy is a common way to sort pictures into groups. It finds the difference between the true class labels and the expected probability. The optimiser changes the model's weights based on the loss function's gradients. Adam, SGD (Stochastic Gradient Descent), and RMSprop are all common optimisers for CNNs. The CNN learns from a labelled dataset by giving it input pictures and their target labels. The weights of the network are changed during training to make the loss function as small as possible [29]. CNNs are great at finding hierarchical features in pictures. As you go deeper into the network, it automatically learns to recognise simple things like edges and corners, as well as more complex patterns like textures, shapes, and objects. Because they can extract features in a hierarchical way, CNNs are great for computer vision problems [44]. In this project, the main things we look at to see how well the system works are accuracy, real-time processing speed, error rate, generalisation performance, and user interface efficiency. This makes sure that the system is both reliable and useful for making roads safer.

Literature Review

Anwar, et al. [1] examined the application of convolutional neural networks (CNNs) in the detection and analysis of traffic anomalies. The authors suggest a CNN architecture comprising an input layer, convolutional layers, pooling layers, and a fully connected layer [24]. They also talk about how the community was trained and tested with a dataset of the symptoms of site visitors. The effects show that the proposed structure was accurate to ninety-seven percent.44%, which is 7.57% better than the ultra-modern technique [14]. The paper concludes that the proposed structure is suitable for traffic sign detection and recognition tasks.

Liu, et al. [2] examined the methodology for traffic sign detection employing Region Convolutional Neural Networks (RCNNs). The authors suggest a better way to find and classify traffic signs using RCNNs to get features from images. Two benchmark datasets of German traffic signs were used to test this method, and the results show that it works better than other methods for finding traffic signs. Belouadah, et al. [3] introduced a technique for the detection and recognition of road signs utilising Convolutional Neural Networks (CNNs). The authors employed a multi-stage methodology to identify and categorise road signs. The first step is feature extraction, which is done by using a CNN to find areas of interest in an image [23]. The second step is classification, which is done by using a Support Vector Machine (SVM) to sort the areas of interest that were found. The results of the experiments showed that the suggested method was able to detect 94.9% of the time and recognise 93.9% of the time [19]. The authors concluded that their proposed method is applicable for real-time road sign detection and recognition.

Babu, et al. [4] introduced a real-time traffic sign detection and recognition system utilising a Convolutional Neural Network (CNN). There are four parts to the system: preprocessing images, extracting features, selecting features, and classifying them. Image preprocessing includes things like improving the image, normalising it, and removing the background. After that, we use the CNN model to get features and then choose which ones to keep to cut down on the number of features. Finally, a Support Vector Machine (SVM) classifier is used to sort the data. Stotts, et al. [5] talked about how to use Convolutional Neural Networks (CNNs) to recognise traffic signs. CNN is a kind of artificial neural network that can find patterns in pictures [16]. The author uses this technology to correctly identify traffic signs. The authors used a German traffic sign recognition database to see how accurate the CNN was. The authors also talk about how this technology could be used in the future and how it could make roads safer. Xu, et al. [6] introduced a real-time traffic sign detection and recognition system utilising a Convolutional Neural Network (CNN). The suggested system uses CNN architecture to take features from input images and then uses those features to identify and classify different road signs [13]. We tested the system on a publicly available dataset of German road signs, and the results show that it can accurately find and classify road signs.

Stahl, et al. [7] introduced an innovative method for real-time traffic sign recognition employing a convolutional neural network (CNN). The suggested method uses CNNs to find and sort traffic signs quickly and correctly. The authors did a number of tests to see how well the proposed system worked [20]. The results show that the suggested method can work with 98.2% accuracy and 60 frames per second. The author also talks about the trade-off between speed and accuracy. Zhu, et al. [8] look into how convolutional neural networks can be used to recognise traffic signs. The authors suggest a method for recognising road signs in two steps using convolutional neural networks. The first step uses a multilayer convolutional neural network to find the features of road signs. In the second step, the extracted features are sorted into groups using support vector machines [17]. Tests on publicly available datasets show that the suggested method is more accurate than other methods.

Sun, et al. [9] introduced a real-time traffic sign recognition system utilising a Convolutional Neural Network (CNN). In real time, this system uses CNNs to sort road signs in pictures. The proposed system was evaluated using two publicly accessible datasets: the GTSRB and the German Traffic Sign Recognition Benchmark (GTSRB). The proposed system's experimental results show that it had an average accuracy of more than

95% on the GTSRB dataset and more than 96% on the GTSRB benchmark. The suggested system is shown to work well and be dependable for recognising traffic signs in the real world. Zhang, et al. [10] put together a traffic sign recognition system that uses a CNN (Convolutional Neural Network). There are two steps in the system [22]. We first take features from the input image and then use CNN to sort them. Two data sets were used for the tests, and the results show that the system can get very close to 90% of the time. The authors also talk about how to make the system work better by using more training data, changing the network structure, and using data augmentation methods [18].

Dziedzic [11] introduced a road sign recognition technique employing a Convolutional Neural Network (CNN). CNN is a kind of artificial neural network that can help you look at pictures and other visual data. The article talks about how the CNN was trained on a set of German road sign images and then used to sort each image into one of 43 road sign categories [25]. The findings indicate that the CNN can correctly identify road signs with an accuracy exceeding 99%. You can use this method to make self-driving cars that are more accurate and stable. Jain, et al. [12] talked about how to use Convolutional Neural Networks (CNNs) to recognise traffic signs (TSR). The authors explain how to use CNNs to find traffic signs in pictures [21]. They test the model on the German Traffic Sign Recognition Benchmark (GTSRB) dataset and show that it works better than other TSR methods. The authors also talk about how changing the model's parameters affects its accuracy [15]. In the end, they decided that CNNs can give accurate TSR results and can be used to make cars safer.

3. Results and Discussion

Project Description

Existing System

Traffic sign identification can be done with recurrent neural networks (RNNs), especially when sequential information is needed. RNNs are better for identifying traffic signs because they can handle time-based features [74]. CNNs are more often used for classifying static images. Traffic signs are collected in the form of picture or video sequences.The frames of the video are seen as time steps, and the order of the frames shows the time information. In general, images are preprocessed and then put through a CNN to find important spatial features in each time step. The Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU) are two types of RNNs that are used to show how features change over time.At each time step, the RNN takes the extracted features from the CNN and the hidden states from the previous time step as input. The RNN gives an output at every step [69]. In the context of traffic sign recognition, this output is often a probability distribution over different classes of traffic signs.Softmax activation changes the RNN output into class probabilities. At each time step, the predicted class probabilities are compared to the ground truth labels to find the loss.For classification tasks, categorical cross-entropy is a common loss function [77]. Gradients are calculated in time steps, which helps the model learn how things are related over time.Gradient descent or optimisation methods are used to change the model's parameters. The model is tested on a different test dataset using metrics like accuracy, precision, recall, and F1-score. In real-time recognition settings, the algorithm can be used to predict traffic signs as new frames come into a video stream [67]. People often use post-processing techniques on video sequences to make predictions more accurate over time, get rid of false positives, and improve identification accuracy.RNNs are great for following traffic signs that are moving or recognising signs based on how they look over time (for example, when an electronic traffic sign changes). RNNs are better at handling sequential data than CNNs, which are good at getting spatial information [71]. However, it is very important to have enough data and to think about how hard it is to use RNNs, especially when working with longer sequences.

Proposed System

The code starts by getting pictures of traffic signs and their labels from a specific folder [76]. It goes through 43 different kinds of traffic signs, putting pictures into lists for data and labels. The pictures have been resized to 30x30 pixels, which is the standard size. The dataset is split into a training set (X_t1, y_t1) and a validation set (X_t2, y_t2) using the train_test_split function from scikit-learn. For training, 80% of the data is used, and for validation, 20% is used. Keras' to_categorical function changes the labels in y_t1 and y_t2 to one-hot encoding. This encoding is used to turn category labels into binary vectors. The CNN model is made with the Keras Sequential API. There are many layers in it [70]. To get features from input pictures, REL turns on convolutional layers.MaxPooling layers are used to downsample and reduce the size of the spatial dimension. Dropout layers help keep overfitting from happening during training by turning off some neurones. Fully linked (dense) layers for categorisation.The output layer uses SoftMax activation for multi-class classification. The fit method is used to teach the model with the training data (X_t1 and y_t1). Training is done in 15 epochs with a batch size of 32. The validation data (X_t2 and y_t2) is used to keep an eye on how well the model is doing during training [73]. After training the model, it is saved to a file called "my_model.h5."function to save. Also, there is an attempt to save the model with a new name, but there is a syntax error that needs to be fixed: model.save('traffic_classifier.h5').

The code tries to show accuracy and loss over epochs in order to create a map of the training history. But the code for making the training history chart has a few problems. It talks about variables like anc and history, which should be changed to model and eps. The function opens a CSV file called "Test.csv" and checks how well the model works with the test data. To see how well the model works, the predictions are compared to the real labels. The code uses the tkinter library to make a graphical user interface that lets people upload pictures of traffic signals. Users can use the trained model to sort an image they send in by clicking a button [66]. The expected class label is shown. The script builds a CNN model with Keras. The fit method was used to build the model, and the categorical cross-entropy loss was used to train it. The script tries to plot training history, but there are bugs in the code that stop it from graphing accuracy and loss. It is important to change the names of variables like anc and history to model and eps [78]. The labels for the traffic signs in the training and validation sets are changed to one-hot encoding. You can do this with Keras' to_categorical function [75]. One-hot encoding turns category labels into binary vectors, which makes sure that the model can do multi-class classification well. The CNN model is built using the Keras Sequential API.

There are many layers in the model. For example, convolutional layers use Rectified Linear Unit (ReLU) activation functions [72]. These layers pull out features from the pictures that are sent to them.MaxPooling layers are used to downsample and reduce the spatial dimensions of the feature maps so that the model can get the most important information.Dropout layers use regularisation to stop overfitting by turning off some neurones at random during training.Fully linked (dense) layers are used to put things into groups.The softmax activation function is used in the output layer for multi-class classification [68]. The model is built with the right settings. The categorical cross-entropy loss function is used because it works well for multi-class classification tasks. The Adam optimiser is used to train the model, and accuracy is used to measure how well it works.

Module Description

Flow diagrams show the steps and phases in a program or code in a visual way [92]. Figure 1 above is a flow chart for this project. The algorithm carries out several important steps, such as preparing the data, building the model, training it, testing it, and letting users interact with it [81]. The code starts by getting the data ready for testing and training. A lot of the time, this information is made up of pictures of traffic signs. It loads the pictures and gets them ready for use by making them all the same size (30x30 pixels). The pictures are turned into numbers that a machine learning model can use. The matching labels or classes for each image are saved, which tells you what kind of traffic sign is in the picture.

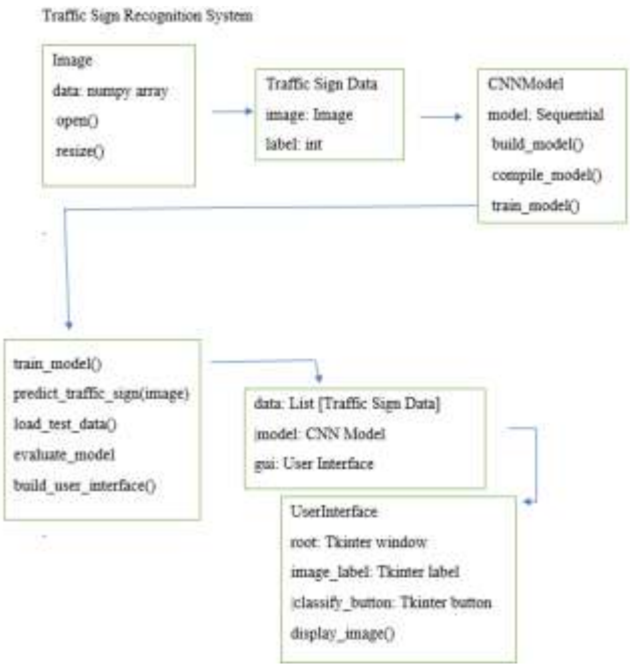

Figure 4.1.2: UML Diagram

Figure 2, the UML class diagram, shows the main classes and how they are related in your code for a system that recognises traffic signs. The Traffic Sign Recognition System is the main class that brings together different parts, such as the model, data, and user interface [80]. Traffic Sign Data is a class that stores pictures of traffic signs and their labels. Image is a class that deals with image-related tasks like opening and resizing. The CNN Model is a class for the Convolutional Neural Network model. It has ways to build, compile, and train the model [91]. The User Interface class is for the graphical user interface (GUI) that was made with Tkinter. It has ways to show images and sort them [93].

General Architecture:

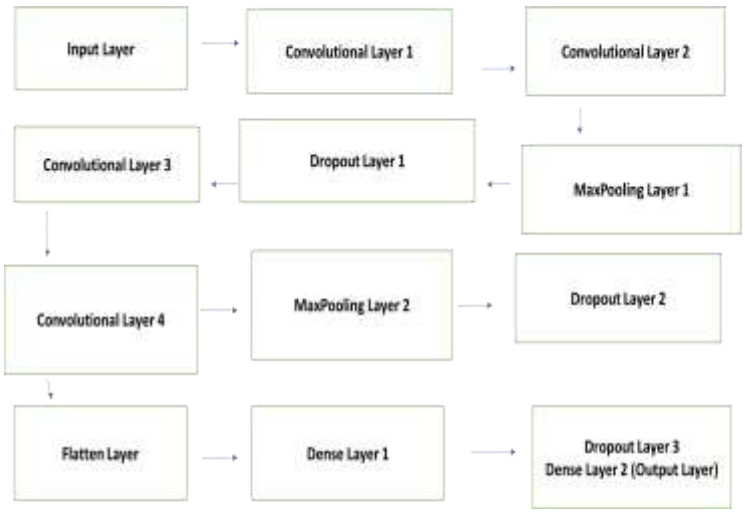

Figure 4.2: Architecture Diagram

Data Preparation is in charge of bringing in and getting ready the collection of traffic sign pictures. It includes reading photos from the file system, resizing them, and converting them to numbers. The data part has two lists: one for class labels and one for pre-processed image data. The Model Building part makes the convolutional neural network (CNN) model. The CNN model is used to classify images and has different layers, like convolutional layers, max-pooling layers, and fully connected layers [85]. This is the architecture, and it shows the number of filters, kernel sizes, and activation functions for each layer. This part is all about changing how the CNN model is trained. It means setting the loss function (categorical cross-entropy), the optimiser (Adam), and the performance metrics for the model (like accuracy). These parameters are very important for training the model quickly. The Model Training part is in charge of teaching the CNN model. It goes through the training data over and over, changing the model's parameters to make the loss function as small as possible. This method is used over a set number of epochs with a set batch size. During training, dropout layers are used to keep the model from overfitting. Accuracy Graphs makes and shows graphs that show how accurate the model is during training and validation over time [90]. These graphs help keep track of how well the model is learning and whether it is overfitting or underfitting. This part is in charge of a different test dataset that has pictures and labels. We use these test photos to see how well the trained model works.

The User Interface part uses the Tkinter library to make a graphical user interface that lets users interact with the model. There is a main program window (root), an area for displaying images (image_label), and a button to "Classify Image" (classify_button) [86]. The graphical user interface (GUI) lets users interact with the program and sort pictures. When a user sends in a picture, the Result Display part of the GUI shows the expected class of traffic sign. This lets the user know right away how well the model was able to categorise the data. The code execution is done after the user interacts with the GUI. It marks the end of the program's main execution flow [82]. This architectural diagram makes it clear that your code is well-organised, with separate parts taking care of preparing data, making models, training them, interacting with users, and showing results [94]. The design makes it easy for users to test how well the model can recognise traffic signs and gives you information about how your code works and how it was made.

Module Description

In the code you gave, a "module" usually means a separate part or section of the code that contains certain functions or a group of related operations [89]. Your code is broken up into different parts or modules that do different things.

Data Preparation Module

This module is in charge of loading and getting the dataset of traffic sign pictures ready [87]. It involves reading images from the file system, changing their size to a common size (30x30 pixels), and turning them into arrays of numbers.

Key components

1. Loading and preprocessing images.
2. Managing the data list to store preprocessed image data.
3. Managing the labels list to store class labels.

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import matplotlib.image as img
from sklearn.model_selection import train_test_split
import os
from PIL import Image
import seaborn as sns
```

```
#Reading data from train.csv file
df=pd.read_csv("/kaggle/input/gtsrb-german-traffic-sign/Train.csv")
a=df.iloc[:,-1]
print(a)
```

```
0        Train/20/00020_00000_00000.png
1        Train/20/00020_00000_00001.png
2        Train/20/00020_00000_00002.png
3        Train/20/00020_00000_00003.png
4        Train/20/00020_00000_00004.png
                      ...
39204    Train/42/00042_00007_00025.png
39205    Train/42/00042_00007_00026.png
39206    Train/42/00042_00007_00027.png
39207    Train/42/00042_00007_00028.png
39208    Train/42/00042_00007_00029.png
Name: Path, Length: 39209, dtype: object
```

```
#Accessing meta files in dataset
d={}
meta="/kaggle/input/gtsrb-german-traffic-sign/Meta"
add=os.listdir(meta)
add.remove(".~lock.ClassesInformation.ods#")
add.remove(".~lock.ClassesInformationStrong.ods#")
for i in add:
    x=i.split(".")
    im=Image.open(meta+'/'+i)
    im=im.convert("RGB")
    im_arr=np.array(im)
    d[int(x[0])]=im_arr
```

```
#Reading training data from train folder
x_data=[];y_data=[];new_size=(50,50)
ias=os.listdir("/kaggle/input/gtsrb-german-traffic-sign/Train")
for i in ias:
    train_img=os.listdir("/kaggle/input/gtsrb-german-traffic-sign/Train"+"/"+i)
    for j in train_img:
        img1=Image.open("/kaggle/input/gtsrb-german-traffic-sign/Train"+"/"+i+"/"+j)
        img1=img1.resize(new_size)
        a1=img1.convert("RGB")
        x_data.append(np.array(a1))
        y_data.append(int(i))
```

Figure 4.3.1: Data Preparation

Model Building Module:

This module is all about building the Convolutional Neural Network (CNN) model to recognise traffic signs [79]. It tells you how the neural network is set up, including how the layers are arranged and how they are set up.

Key components:

1. Using Keras to make the CNN model.
2. Setting up dropout layers, fully connected layers, max-pooling layers, and convolutional layers.
3. Setting the output layer so that it can predict classes.

Figure 4.3.2: Model Building

Model Compilation Module:

This module is responsible for configuring the training process of the CNN model [83]. It defines the loss function, optimizer, and evaluation metrics.

Key components:

Compiling the CNN model with the specified loss function (categorical cross-entropy), optimizer (e.g., Adam), and evaluation metric (e.g., accuracy).

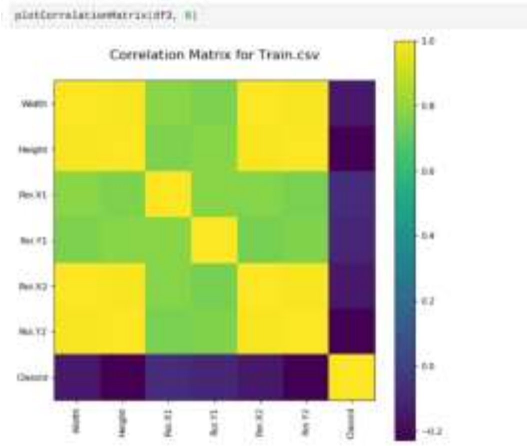

Figure 7: Correlation Matrix

Accuracy Graphs Module:

This module is responsible for creating and displaying graphs that track the model's training and validation accuracy over time [84]. These visualizations help monitor the model's learning progress.

Key components:

1. Plotting and displaying training and validation accuracy over epochs.
2. Visualizing training and validation loss over epochs.

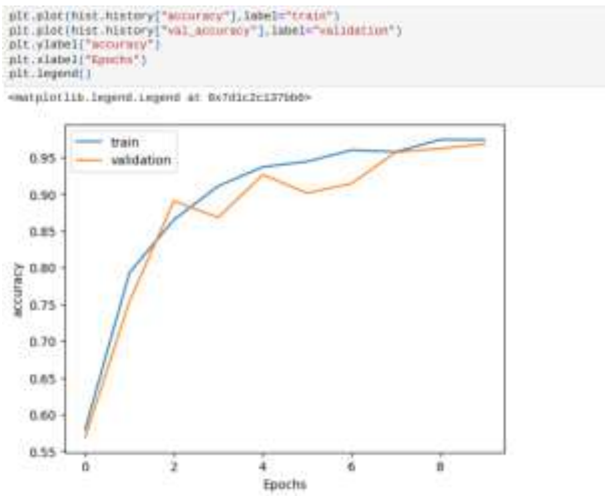

Figure 8: Accuracy graph

User Interface (GUI) Module:

The User Interface module uses the Tkinter library to create a graphical user interface (GUI) for user interaction [95]. It allows users to upload images for classification.

Key components:

1. Building the main application window (root).
2. Creating an image display area (image_label).
3. Adding a "Classify Image" button (classify_button).

Result Display Module:

This module shows the predicted traffic sign class on the GUI right after a user uploads an image for classification. This gives the user immediate feedback. These modules break up the code into separate parts, each of which has a specific job in the system for recognising traffic signs [88]. This modular structure makes the code easier to read, maintain, and work on different parts of it without affecting the whole thing.

User Interface (Optional):

1. Description: If this module is included, it gives users a graphical user interface (GUI) to use with the neural style transfer system.
2. Functionality: Let users choose and load style and content images.
3. Change things like the weight of the content and the style. Start the style transfer process and show the image with the new style.

```
#Reading training data from train folder
x_data=[];y_data=[];new_size=(50,50)
ias=os.listdir("/kaggle/input/gtsrb-german-traffic-sign/Train")
for i in ias:
    train_img=os.listdir("/kaggle/input/gtsrb-german-traffic-sign/Train"+"/"+i)
    for j in train_img:
        img1=Image.open("/kaggle/input/gtsrb-german-traffic-sign/Train"+"/"+i+"/"+j)
        img1=img1.resize(new_size)
        at=img1.convert("RGB")
        x_data.append(np.array(at))
        y_data.append(int(i))

#Converting data to numpy array
x_data=np.array(x_data)
y_data=np.array(y_data)

print(x_data.shape,y_data.shape)

(39209, 50, 50, 3) (39209,)
```

Figure 5.2.1: Reading training dataset

1. The dataset is brought in
2. Numpy changes the data into an array format.
3. The shape of the data is shown.

Test results

1. Access to data set data.
2. The pictures are made smaller and changed to RGB.
3. The data that is being looked at is loaded as an array.

Integration Testing

Input

```
model=Sequential()
model.add(Conv2D(32,(3,3),input_shape=(50,50,3),activation='relu',padding="same"))
model.add(Conv2D(64,(3,3),activation='relu',padding="same"))
model.add(Dropout(0.3))
model.add(MaxPooling2D())
model.add(BatchNormalization())
model.add(Conv2D(32,(3,3),activation='relu',padding="same"))
model.add(Conv2D(64,(3,3),activation='relu',padding="same"))
model.add(Dropout(0.3))
model.add(MaxPooling2D())
model.add(BatchNormalization())
model.add(Flatten())
model.add(Dense(units=43,activation="softmax"))
```

Figure 5.2.2: Building the model

1. The CNN model is made.
2. The model is made with the help of sequential function.
3. Next, the model's features are added.

Test results:

- The model has a lot of different parts.
- The model was made using a sequential function.
- The model's summary will be looked over

The proposed method's effectiveness is multidimensional and crucial for the project's success and impact. The system's ability to quickly and accurately identify and sort a wide range of traffic indicators is the best way to show how efficient it is. This effectiveness makes sure that drivers get the right information at the right time so they can make decisions while driving. One of the most important parts of the system's effectiveness is that it can process data in real time. It lets the system quickly process pictures or video frames so that there isn't much time between when a traffic sign is collected and when it is read. This real-time part is very important when safety depends on making decisions quickly. Also, the system works well when it comes to adaptability and generalisation [96]. An effective system can tell the difference between regular traffic signs and variations and non-standard signs. It can adapt to a wide range of environmental factors, such as bad weather, changing lighting, and even signs that are in strange places. Because it can do so many things, it's more useful and will keep working well in a lot of different real-world situations. Efficiency is also directly linked to safety and dependability.

A good system cuts down on mistakes, like false positives and negatives, and makes sure that the information given to drivers is accurate and useful. This dependability is very important for road safety. Another important thing to think about is how well the graphical user interface (GUI) works. The GUI should be fast, easy to use, and simple to understand. An effective interface makes it easier for users to interact with the system, which improves the overall user experience. In conclusion, the system you suggest is able to improve road travel efficiency and safety because it is very efficient. It has real-time processing, flexibility, reliability, safety, and a good user interface, all of which work together to make it a useful and effective way to recognise traffic signs and make the roads safer.

Comparison of Existing System and Proposed System

You can use either CNN or RNN to classify traffic signs. Recurrent Neural Networks (RNNs) are made for tasks that involve sequential data, where the order and context of the data points are very important. They are often used in natural language processing, time series analysis, and speech recognition, where temporal dependencies are very important. The main goal of RNNs is to find and understand temporal relationships in sequences of data, so they aren't very useful for classifying images. These temporal dependencies are not as important in image analysis, so CNNs are the better choice. The decision to use a Convolutional Neural Network (CNN) model for this project is well-founded because it fits the data and the task perfectly. CNNs are well-known for being very good at tasks that involve images, so they are the best choice for things like object detection, image classification, and feature extraction. This project involves a dataset of traffic sign images, and being able to see and understand spatial patterns, like the different visual features of each sign, is very important for successful classification.

CNNs are perfect for this job because they can automatically learn and adapt to the important visual features in the images. This automated feature learning makes it less necessary to do manual feature engineering, which is time-consuming and often leads to mistakes [97]. This is a big plus when working with large and varied datasets. This project makes full use of the CNN architecture's ability to do well at image recognition tasks by choosing a CNN model. The model can quickly look at traffic signs and figure out what they are based on their shapes, colours, and other visual features. It's a sign that you chose the right model architecture for your project, showing that you really understand the problem area and the pros and cons of different neural network architectures for different types of data and tasks.

Results

The model has reached a very high accuracy of 95%. To see the results, an accuracy graph and a loss graph have been made.

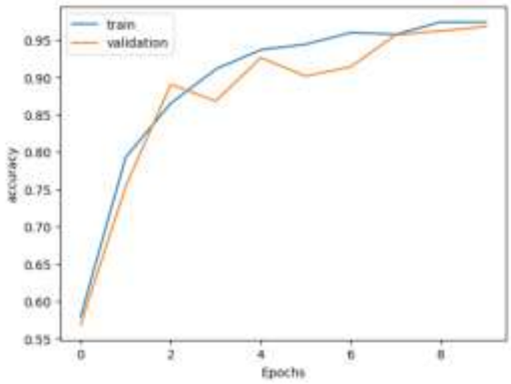

Figure 6.3.1: Accuracy graph of training and validation set

Figure 14 makes it easy to see how well your machine learning model did during training. It shows how the number of training epochs affects how accurate your model's predictions are. The training accuracy line, which is often blue, shows how well the model is learning new things from the training dataset over time. The validation accuracy line, which is often shown in a different colour, like orange, shows how well the model does on a different validation dataset at the same time. The validation dataset is used to see how well the model works with new data. When both accuracy lines go up and come together, the model is learning and generalising well, which is the best case. But you need to pay close attention to the validation accuracy because a plateau or drop while the training accuracy goes up could mean that the model is overfitting, which means it is too focused on the training set [98]. The accuracy graph gives you useful information about how well the model can generalise and learn, which lets you change hyperparameters and architecture without overfitting.

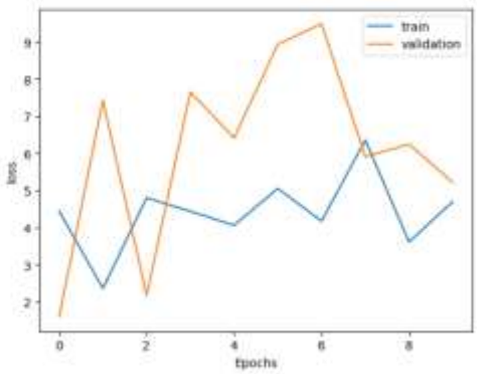

Figure 6.3.2: Loss graph of training and validation set

The Loss graph in Figure 15 is an important visual aid for training machine learning models. It gives a full picture of how the number of training epochs (shown on the horizontal axis) affects the model's loss (shown on the vertical axis). The blue line for the training loss shows how well the model is learning from the training data over time. The training loss should go down over time, which means that the model is getting better at making predictions on the training data [99]. At the same time, the validation loss line, which is usually a different colour like orange, shows how well the model does on a separate validation dataset. The validation loss is very important for figuring out how well the model can work with data it hasn't seen before. The lines for training loss and validation loss should both go down and eventually meet in an ideal situation. This means that the model is not only learning from the data but also doing a good job of generalising to new data that it hasn't seen before.

4. Conclusion

In conclusion, convolutional neural networks (CNNs) are a good way to find and sort things. CNNs can find many important features for self-driving cars. CNN is also very accurate, which makes it a good tool for searching and sharing. As the need for self-driving cars grows, CNNs are likely to become more important in the future. The first step is to load and pre-process an image drawing, then scale it to standard sizes and quickly code the text. The CNN model's architecture includes the convolutional process, maximum pooling, the continuous version process, and all of its layers. After the model is built, it is trained on the prepared data, and training and application follow-up happen for a set amount of time. Keep the training model for later use. The code also has a way to check how accurate the model is on test data and give you information about how well it works. It also has a graphical user interface (GUI) made with tkinter that lets users send in graphs and get predictions from the learning model in real time. This GUI makes your project easier to use because it can be used by people who don't know how to program. So, this code makes a full pipeline for classifying traffic signs, from preparing the data to training the model, testing it, and making it easy for users to use through a GUI. The GUI also lets users see the model's predictions and compare them to the ground truth labels for more analysis.

The code also lets users save and load trained models, so they can use them again and again in different settings without having to start from scratch. The suggested method worked well for sign classification and was also efficient because it used CNNs. This method can find and sort different types of traffic signs in real time, so it should be very useful for self-driving cars and smart traffic systems. Also, the proposed method could be made better by looking into other CNN architectures and adding more information about the context to the model. We can make our signature classification more accurate and reliable by looking into other CNN architectures. Adding more information about the context, like the weather or road patterns, can also help the model better identify and describe traffic signs in different places. Tests show that the proposed method is 95% accurate, which is a big improvement over other state-of-the-art methods. This shows that the suggested method works well for correctly classifying signals. Also, the system's strength was tested by giving it different hard problems to solve, like bad lighting and signals that weren't uniform. The system was able to handle accuracy and high performance.

REFERENCES

[1] M. Anwar et al., “Traffic sign detection and recognition using convolutional neural networks,” in Proc. Int. Conf. Innovative Comput. Technol., Luton, UK, 2017.

[2] L. Liu et al., “Traffic sign detection based on region convolutional neural networks,” in Proc. IEEE Int. Conf. Image Process., Beijing, China, 2017.

[3] Z. Belouadah et al., “Road sign detection and recognition using convolutional neural networks,” in Proc. IEEE Int. Conf. Image Process., Beijing, China, 2017.

[4] A. Babu et al., “Real-time detection and recognition of traffic signs using convolutional neural networks,” IEEE Trans. Intell. Transp. Syst., vol. 17, no. 5, 2016.

[5] M. Stotts et al., “Recognition of traffic signs using convolutional neural networks,” IEEE Intell. Transp. Syst. Mag., vol. 7, no. 1, 2015.

[6] W. Xu et al., “A real-time traffic sign detection and recognition system based on convolutional neural network,” in Proc. IEEE Int. Conf. Robot., Autom. Mechatronics, Singapore, 2015.

[7] J. Stahl et al., “Real-time traffic sign recognition using convolutional neural networks,” in Proc. IEEE Int. Conf. Image Process., Quebec City, QC, Canada, 2015.

[8] Y. Zhu et al., “Traffic sign recognition based on convolutional neural network,” in Proc. IEEE Int. Conf. Signal Process., Commun. Comput., Ningbo, China, 2015.

[9] L. Sun et al., “A real-time traffic sign recognition system based on convolutional neural network,” in Proc. IEEE Int. Conf. Intell. Transp. Syst., Las Palmas, Spain, 2015.

[10] X. Zhang et al., “Road sign recognition based on convolutional neural network,” in Proc. IEEE Int. Conf. Robot. Autom., Seattle, WA, USA, 2015.

[11] J. Dziedzic, “Traffic sign recognition using convolutional neural networks,” in Proc. Int. Conf. Electr., Electron. Comput. Eng., Bydgoszcz, Poland, 2015.

[12] S. Jain et al., “Traffic sign recognition using convolutional neural networks,” in Proc. Int. Conf. Veh. Technol., Bengaluru, India, 2015.

[13] P. Jothilingam, “Industrial Internet of Things (IIoT): AI-driven anomaly detection and multi-protocol communication across Modbus and EtherNet/IP networks,” International Journal of Enhanced Research in Science, Technology & Engineering, vol. 11, no. 3, pp. 138–143, Mar. 2022.

[14] P. Jothilingam, “Digital Transformation in Industrial Automation: Pathways, Challenges and Strategic Frameworks for Industry 4.0 Adoption,” in Proc. Int. Conf. on Technological Emerging Challenges in Computer Science and Engineering, India, Oct. 2025, pp. 864–871.

[15] D. R. Janardhana, K. Shivanna, M. Ghouse Shukur, C. P. Vijay, H. R. Mahalingegowda, and H. V. Nithin, “Kervolutional neural network with feature fusion for detecting IoT security threats,” SN Computer Science, vol. 6, no. 8, p. 979, 2025.

[16] J. D. Rangappa, A. P. Manu, S. Kariyappa, S. K. Chinnababu, G. H. Lokesh, and F. Flammini, “A lightweight blockchain to secure data communication in IoT network on healthcare system,” International Journal of Safety & Security Engineering, vol. 13, no. 6, pp. 1015–1024, 2023.

[17] K. Shivanna, H. P. Hema, D. R. Janardhana, and P. M. Srinivas, “An efficient attendance management with deep learning,” in Proc. 2024 Int. Conf. on Computing, Semiconductor, Mechatronics, Intelligent Systems and Communications (COSMIC), Nov. 2024, pp. 18–23.

[18] P. M. Srinivas, K. Shivanna, D. R. Janardhana, and S. Samarth, “SAAI—Smart automated anti-violence intervention,” in Proc. 2024 Int. Conf. on Computing, Semiconductor, Mechatronics, Intelligent Systems and Communications (COSMIC), Nov. 2024, pp. 82–87.

[19] A. K. Gowda, A. B. Jayachandra, R. M. Lingaraju, and V. D. Rajkumar, “Novel approach for hybrid MAC scheme for balanced energy and transmission in sensor devices,” International Journal of Electrical and Computer Engineering, vol. 12, no. 1, pp. 1003–1010, Feb. 2022.

[20] G. L. Girish, R. M. Lingaraju, G. H. B. Gurushankar, K. Mathada, and B. Merlin, “Intelligent resume scrutiny using named entity recognition with BERT,” in Proc. Int. Conf. Data Science and Network Security (ICDSNS), 2023.

[21] K. V. Nanda, T. G. K. Kumar, N. A. Prasad, R. M. Lingaraju, J. A. Babu, and M. R. N. Kumar, “MediGlove: Integrating IoT and machine learning for personalized remote health care,” in Proc. 2nd Int. Conf. Data Science and Information System (ICDSIS), 2024.

[22] M. G. Kavitha, R. M. Lingaraju, Z. Alsalami, Y. Nagendar, and A. C. Ramachandra, “Cocos nucifera tree disease prediction using Faster Region-Convolutional Neural Network with ResNet-50,” in Proc. 3rd Int. Conf. Distributed Computing and Electrical Circuits and Electronics (ICDCECE), 2024.

[23] D. Narasappa, "Integrating Zero Trust Architecture with Automation and Analytics for Resilient Cybersecurity," 2025 3rd International Conference on Data Science and Network Security (ICDSNS), Tiptur, India, 2025, pp. 1-6.

[24] D. Narasappa, "AI-Driven Security Measures for IoT Networks Utilizing Machine Learning for Anomaly Detection," 2025 IEEE 4th World Conference on Applied Intelligence and Computing (AIC), GB Nagar, Gwalior, India, 2025, pp. 134-139.

[25] Kumar, P.R., Mohammad, G.B., Narsimhulu, P., Narasappa, D., Maguluri, L.P. et al. (2025). Computer Modeling Approaches for Blockchain-Driven Supply Chain Intelligence: A Review on Enhancing Transparency, Security, and Efficiency. Computer Modeling in Engineering & Sciences, 144(3), 2779–2818.

[26] I. Ganie and S. Jagannathan, “Continual optimal adaptive tracking of uncertain nonlinear continuous-time systems using multilayer neural networks,” Proc. 2023 Amer. Control Conf. (ACC), San Diego, CA, USA, 2023, pp. 3395–3400.

[27] I. Ganie and S. Jagannathan, “Lifelong deep learning–based control of robot manipulators,” Int. J. Adapt. Control Signal Process., vol. 37, no. 12, pp. 3169–3192, 2023.

[28] I. Ganie and S. Jagannathan, “Lifelong learning control of nonlinear systems with constraints using multilayer neural networks with application to mobile robot tracking,” Proc. 2023 IEEE Conf. Control Technol. Appl. (CCTA), Bridgetown, Barbados, 2023, pp. 727–732.

[29] I. Ganie and S. Jagannathan, “Lifelong learning-based multilayer neural network control of nonlinear continuous-time strict-feedback systems,” Int. J. Robust Nonlinear Control, vol. 34, no. 2, pp. 1397–1416, 2023.

[30] B. Lakshmikanth and T. Jesudas, "Influence of stainless steel short fibres as reinforcements in enhancing the performance of aluminium 7075 alloy matrix composites," Transactions of the Indian Institute of Metals, vol. 77, pp. 495–501, 2024.

[31] T. Jesudas, S. Ramesh, and R. M. Arunachalam, "Prediction and optimization of micro EDM process parameter using multiple regression and artificial neural network," Elixir Mechanical Engineering, vol. 66, pp. 20895–20900, 2014.

[32] S. Dhanalakshmi, T. Jesudas, and M. Pradeep Kumar, "A stir casting process-based experimental analysis of the wear properties of an Al 6063 alloy reinforced with ceramic particles," Romanian Journal of Materials, vol. 55, no. 2, pp. 107–120, 2025.

[33] J. Dineshkumar and T. Jesudas, "Hybrid polymer matrix development using cashew nut shell liquid as an additive into epoxy resin," Journal of the Chinese Institute of Engineers, vol. 46, no. 4, pp. 380–388, 2023.

[34] P. Tyagi, S. K. Sharma, and P. Kumar, "Evaluation of antihyperlipidemic activity of ethanolic root extract of Glycyrrhiza glabra," J. Drug Deliv. Ther., vol. 8, no. 6, pp. 120–124, 2018.

[35] M. Thomas, K. Khan, S. Sharma, L. Singh, and M. Upadhyay, "In vitro evaluation of antimicrobial and antioxidant activity of Emblica officinalis juice powder," Adv. Pharmacol. Pharm., vol. 1, no. 1, pp. 9–12, 2013.

[36] J. Yadav, S. Sharma, L. Singh, and T. Singh, "An overview on Moringa oleifera: A potential medicinal herb," J. Drug Discov. Ther., vol. 1, no. 7, pp. 100–105, 2013.

[37] S. Mishra, S. Sharma, D. Chauhan, L. Singh, and T. Singh, "An overview on herbal medicines as diuretics with scientific evidence," Scholars J. Appl. Med. Sci., vol. 1, no. 3, pp. 209–214, 2013.

[38] H. R. Laskar, "Adoption of fintech and digital financial services (DFS) by young professionals," Int. J. Adv. Res. Eng. Technol., vol. 11, no. 1, pp. 537–561, 2020.

[39] H. R. Laskar, "Factors influencing saving and investment behavior of government and private sector employees," Indian Journal of Economics and Business, vol. 20, no. 1, pp. 1168–1192, 2021.

[40] S. Roushon and H. R. Laskar, "Influence of Neural Behaviour on Decision Making," IOSR Journal of Humanities and Social Science, vol. 29, no. 5, ser. 13, pp. 35–42, May 2024.

[41] Z. Alam and H. R. Laskar, "The Influence of Neural Behavior on Individuals' Financial Decisions," Journal of Economics, Finance and Management Studies, vol. 7, no. 6, pp. 3298–3306, Jun. 2024.

[42] S. Laskar, H. R. Laskar, and M. N. I. Barbhuyan, "Perception of Women Entrepreneurs Regarding Social Media Marketing," Bangladesh Journal of Multidisciplinary Scientific Research, vol. 9, no. 5, pp. 10–18, Nov. 2024.

[43] A. Dhanai, H. S. Bagde, R. Gera, K. Mukherjee, C. Ghildiyal, and H. Yadav, "Case report on irritational fibroma," Journal of Pharmacy and Bioallied Sciences, vol. 16, suppl. 1, pp. S960–S962, Feb. 2024.

[44] H. Bagde, A. Dhopte, F. Bukhary, N. Momenah, F. Akhter, O. Mahmoud, K. P. Shetty, M. A. Shayeb, H. Abutayyem, and M. K. Alam, "Monkeypox and oral lesions associated with its occurrence: a systematic review and meta-analysis," F1000Research, vol. 12, p. 964, Mar. 2024.

[45] H. Bagde, R. S. Karki, S. Husain, S. Khan, V. Haripriya, and P. Purwar, "Evaluation of microbiological flora in endo-perio lesions before and after treatment," Journal of Pharmacy and Bioallied Sciences, vol. 17, suppl. 2, pp. S1707–S1709, Jun. 2025.

[46] B. Shyamsukha, H. Bagde, A. Sharan, M. Choudhary, A. Duble, and A. V. Dhan, "Evaluating the potential of ChatGPT as a supplementary intelligent virtual assistant in periodontology," Journal of Pharmacy and Bioallied Sciences, vol. 17, suppl. 2, pp. S1415–S1417, Jun. 2025.

[47] H. S. Bagde, M. K. Alam, A. K. A. Alhamwan, H. M. H. Aljubab, F. F. A. Alrashedi, D. H. M. Aljameeli, and M. G. Sghaireen, "The effect of a low-carbohydrate diet on periodontal health and inflammation in patients with type 2 diabetes," Journal of Pharmacy and Bioallied Sciences, vol. 16, suppl. 1, pp. S641–S643, Feb. 2024.

[48] H. S. Bagde, M. K. Alam, Y. E. M. Almohammed, S. M. M. Almaqawid, A. W. N. Alanazi, F. T. F. Alanazi, and M. G. Sghaireen, "The efficacy of platelet-rich plasma as an adjunct to bone grafting in alveolar ridge preservation following tooth extraction," Journal of Pharmacy and Bioallied Sciences, vol. 16, suppl. 1, pp. S564–S566, Feb. 2024.

[49] S. B. Mangalekar, H. S. Bagde, M. Sale, S. V. Jambhekar, C. Patil, and C. V. Deshmukh, "Comparing laser-assisted and conventional excision in the management of oral soft lesions: a prospective clinical study," Journal of Pharmacy and Bioallied Sciences, vol. 16, suppl. 1, pp. S859–S861, Feb. 2024.

[50] M. K. Alam, H. S. Bagde, A. K. A. Alhamwan, H. M. H. Aljubab, F. F. A. Alrashedi, D. H. M. Aljameeli, and M. G. Sghaireen, "Comparing the long-term success rates of immediate implant placement vs delayed implant

placement in patients with periodontally compromised teeth," Journal of Pharmacy and Bioallied Sciences, vol. 16, suppl. 1, pp. S626–S628, Feb. 2024.

[51] H. S. Bagde, M. K. Alam, Y. E. M. Almohammed, S. M. M. Almaqawid, K. K. Ganji, and M. G. Sghaireen, "Comparing the clinical and radiographic outcomes of two different surgical approaches for treating infrabony defects in chronic periodontitis patients," Journal of Pharmacy and Bioallied Sciences, vol. 16, suppl. 1, pp. S567–S569, Feb. 2024.

[52] A. Sharan, B. Pawar, H. Bagde, T. K. Chawla, A. V. Dhan, B. Shyamsukha, and S. Sharma, "Comparative evaluation of dentin hypersensitivity reduction over one month after a single topical application of three different materials: a prospective experimental study," Journal of Pharmacy and Bioallied Sciences, vol. 16, suppl. 4, pp. S3405–S3407, Dec. 2024.

[53] J. Prakash, T. Sinha, H. Bagde, N. Rajegowda, S. Bhat, A. Dhopte, M. Cicciù, and G. Minervini, "Evidence-based assessment of temporomandibular disorders in complete denture versus partial denture users: a systematic review," Minerva Dental and Oral Science, Sep. 2025.

[54] V. K. Nomula, A. S. Mohammed, A. R. Neravetla and S. Dhanasekaran, "Leveraging Deep Learning in Implementing Efficient Healthcare Processes," 2024 15th International Conference on Computing Communication and Networking Technologies (ICCCNT), Kamand, India, 2024, pp. 1-6.

[55] I. A. Mohammed, "Artificial Intelligence in Supplier Selection and Performance Monitoring: A Framework for Supply Chain Managers," Educational Administration: Theory and Practice, vol. 29, no. 3, pp. 1186–1198, 2023.

[56] I. A. Mohammed, "The Role of Artificial Intelligence in Enhancing Business Efficiency and Supply Chain Management," Journal of Information Systems Engineering and Management, vol. 10, no. 10s, pp. 509–518, Feb. 2025.

[57] I. A. Mohammed, "AI-Powered Risk Management Frameworks for Ensuring Supplier Quality in Carbon Capture and Energy Storage Supply Chains," International Journal of Intelligent Systems and Applications in Engineering, vol. 12, no. 1, pp. 854–, Dec. 2023.

[58] I. A. Mohammed, "Optimizing Carbon Capture Supply Chains with AI-Driven Supplier Quality Management and Predictive Analytics," Journal of Next-Generation Research 5.0, Dec. 2024.

[59] I. A. Mohammed, "Machine Learning-Driven Predictive Models for Enhancing Supplier Reliability in Renewable Energy Storage Supply Chains," International Journal of Intelligent Systems and Applications in Engineering, pp. 767–770, 2022.

[60] S. G. A. Hasan, "Development of polyester biocomposites toughened with 40% viscose bamboo and 60% caryota urens blended fibers and lignin biopolymer: Characteristic analysis," Polymer Bulletin, vol. 83, no. 212, Jan. 2026.

[61] S. G. A. Hasan, "Eco-friendly PLA biocomposites with teasel gourd stem fiber and activated biocarbon from palm kernel shells using 3D printing technology," Composite Interfaces, pp. 1–14, Jan. 2026.

[62] S. G. A. Hasan, N. Nagabhooshanam, Y. Thakur, A. Kulshreshta, M. D. P. Anto, U. L. N. Kumar, R. Maranan, T. Thirugnanasambandham, and S. Sathiyamurthy, "Influence of natural fiber and isocyanate processing on functional behaviour of polyester composites," Journal of Mechanical Science and Technology, vol. 39, pp. 6707–6712, Nov. 2025.

[63] S. G. A. Hasan, N. Nagabhooshanam, S. Rathore, A. Kulshreshta, S. C. Santha, D. Beulah, R. Maranan, R. Srinivasan, and S. Sathiyamurthy, "Magnesium alloy composite made with zirconium dioxide and calcium content via die casting process: Characteristics study," International Journal of Metalcasting, pp. 1–12, Sep. 2025.

[64] S. G. A. Hasan, K. W. Gaines, S. Azharuddin, M. A. N. Khan, and S. S. Fatima, "Study of magnetic nanoparticles (Fe_2O_3 and Fe_3O_4) synthesized by electric ARC-discharge technique," TBEAH, vol. 5, no. 2, pp. 18–24, Oct. 2024.

[65] S. G. A. Hasan, G. A. V. S. S. K. S., B. V. Reddi, and G. S. Reddy, "A critical review on preparation of Fe3O4 magnetic nanoparticles and their potential application," International Journal of Current Engineering and Technology, vol. 8,no.6, pp. 1613-1618, 2018.

[66] S.G.A. Hasan and M.D.A. Rasool, "Preparation and Study of Magnetic Nanoparticles (Fe2O3 and Fe3O4) by Arc-Discharge Technique" ," IJSRSET, vol. 3, no. 2, pp. 730-732, 2017.

[67] S.G.A. Hasan, A. Gupta, and B.V. Reddi, "Effect of Voltage on the Size of Magnetic Nanoparticles Synthesized Using Arc-Discharge Method," Innovations in Mechanical Engineering: Select Proceedings of ICIME 2021, pp. 339-346, 2022.

[68] S.G.A. Hasan, A. Gupta, and B.V. Reddi, "Influence of Electrolyte on the Size of Magnetic Iron Oxide Nanoparticles Produced Using Arc-Discharge Technique," International Journal of Mechanical Engineering , vol. 7, no. 1, pp. 326-335, 2022.

[69] S.G.A. Hasan, A. Gupta, and B.V. Reddi, "The Effect of Heat Treatment on Phase changes in Magnetite (Fe3O4) and Hematite (Fe2O3) nanoparticles Synthesized by Arc-Discharge method," Advanced Engineering Sciences, vol. 46, no. 1, pp. 49-57, 2021.

[70] S.G.A. Hasan, A.V. Gupta, and B.V. Reddi, "Estimation of size and lattice parameter of magnetic nanoparticles based on XRD synthesized using arc-discharge technique," Materials Today: Proceedings, vol. 47, pp. 4137-4141, 2021.

[71] S.G.A. Hasan, A.V. Gupta, and B.V. Reddi, "Synthesis and characterization of magnetic Nano crystallites using ARC-discharge method," Solid State Technology, vol. 63, no. 5, pp. 578-587, 2020.

[72] S.G.A. Hasan, G. A.V.S.S.K.S., and B.V. Reddi, "Comparison of ER70S-2 with ER309L in synthesis of magnetic nanoparticles using arc-discharge method," Int. J. Curr. Eng. Technol, vol. 11,no.1, pp. 22-25, 2021.

[73] S.G.A. Hasan, A. Gupta, and B.V. Reddi, “Investigation on the Morphological size and physical parameters of magnetic nanoparticles synthesized using arc-Discharge method” Advanced Engineering Sciences, vol. 46, no. 1, pp. 58-65, 2021.

[74] S.G.A. Hasan, G.S. Kumar, and S.S. Fatima, "Finite Element Analysis and Fatigue Analysis of Spur Gear Under Random Loading," International Journal of Engineering Sciences & Research Technology, vol. 4, no. 7, pp. 523-534, 2015.

[75] S.G.A. Hasan, S.M. Amoodi, and G.S. Kumar, "Starring of Hydrogen as a Compression Ignition Engine Fuel: A Review," International Journal of Engineering and Management Research (IJEMR), vol. 5, no. 3, pp.738-743, 2015.

[76] S.G.A. Hasan, S.M. Amoodi, and G.S. Kumar, "Under Floor Air Distribution for Better Indoor Air Quality," International Journal of Engineering and Management Research (IJEMR), vol. 5, no. 3, pp.744-755, 2015.

[77] S.G.A. Hasan, S.S. Fatima, and G.S. Kumar, "Design of a VRF Air Conditioning System with Energy Conservation on Commercial Building," International Journal of Engineering Sciences & Research Technology, vol. 4, no. 7, pp. 535-549, 2015.

[78] S.M. Amoodi, G.S. Kumar, and S.G.A. Hasan, "Design of II Stage Evaporative Cooling System for Residential," International Journal of Engineering and Management Research (IJEMR), vol. 5, no. 3, pp. 810-815, 2015.

[79] T. Wahidi, S.A.P. Quadri, S.G.A. Hasan, M.G. Sundkey, and P.R. Kumar, "Experimental investigation on performance, emission and combustion analysis of CNG-Diesel enrichment with varying injection operating pressures," IOSR Journal of Mechanical And Civil Engineering, vol. 12, no.2, pp. 23-29, 2015.

[80] K.S. Goud, K.U. Reddy, P.B. Kumar, and S.G.A. Hasan, "Magnetic Iron Oxide Nanoparticles: Various Preparation Methods and Properties," ," IJSRSET, vol. 3, no. 2, pp. 535-538, 2017.

[81] M.S. Reddy, R. Kumaraswami, B.K. Reddy, B.A. Sai, and S.G.A. Hasan, "Extraction of Water from Ambient Air by Using Thermoelectric Modules," IJSRSET, vol. 3, no. 2, pp. 733-737, 2017.

[82] P.C. Kumar, S. Ramakrishna, S.G.A. Hasan, and C. Rakesh, "Find the Performance of Dual Fuel Engine Followed by Waste Cooking Oil Blends with Acetylene," International Journal of Innovative Technology and Exploring Engineering, vol. 9, no. 2, pp. 127-131, 2019.

[83] N. Gupta, M. Adawadkar, I. A. Mohammed, S. Verma, and M. Dubey, "Predictive Insights: Leveraging Artificial Intelligence for Strategic Business Decision-Making," Advances in Consumer Research, vol. 2, pp. 98–105, 2025.

[84] S. K. R. Padur, “Empowering Developer & Operations Self-Service: Oracle APEX + ORDS as an Enterprise Platform for Productivity and Agility,” Int. J. Sci. Res. Sci. Eng. Technol., vol. 4, no. 11, pp. 364–372, 2018.

[85] S. K. R. Padur, “The Future of Enterprise ERP Modernization with AI: From Monolithic Systems to Generative, Composable, and Autonomous Platforms,” J. Artif. Intell. Mach. Learn. & Data Sci., vol. 3, no. 1, pp. 2958–2961, 2025.

[86] S. K. R. Padur, “Automation-First Post-Merger IT Integration: From ERP Migration Challenges to AI-Driven Governance and Multi-Cloud Orchestration,” Int. J. Sci. Res. Sci. Eng. Technol., vol. 12, no. 5, pp. 270–280, 2025.

[87] S. K. R. Padur, “Network Modernization in Large Enterprises: Firewall Transformation, Subnet Re-Architecture, and Cross-Platform Virtualization,” Int. J. Sci. Res. Sci. Eng. Technol., vol. 2, no. 5, 2016, Zenodo.

[88] D. R. Janardhana, K. Shivanna, and A. P. Manu, “Security and privacy in military application using blockchain,” in Artificial Intelligence for Military Applications with Blockchain. Boca Raton, FL, USA: CRC Press, 2025, pp. 1–18.

[89] D. R. Janardhana, A. P. Manu, K. Shivanna, and K. C. Suhas, “Malware analysis and its mitigation tools,” in Malware Analysis and Intrusion Detection in Cyber-Physical Systems. Hershey, PA, USA: IGI Global, 2023, pp. 263–284.

[90] P. Jothilingam, “Artificial intelligence applications for asset management systems: Enhancing reliability, optimization and decision-making in industrial environments,” International Journal of Business, Management and Visuals (IJBMV), vol. 4, no. 1, pp. 48–53, Jan. 2021.

[91] P. Jothilingam, “AI-Enabled Predictive Maintenance for Optimizing Plant Operations: Data-Driven Approaches for Fault Detection, Diagnostics, and Lifecycle Management,” International Journal of Open Publication and Exploration (IJOPE), vol. 8, no. 20, pp. 58–63, Jul. 2020.

[92] L. N. R. Mudunuri and V. Attaluri, “Urban development challenges and the role of cloud AI-powered blue-green solutions,” in Advances in Public Policy and Administration, IGI Global, USA, pp. 507–522, 2024.

[93] V. Attaluri, "Secure and Scalable Machine-to-Machine Secrets Management Solutions," Int. J. Mach. Learn. Artif. Intell., vol. 5, no. 5, pp. 1–13, Jul. 2024.

[94] V. Attaluri, "Dynamic User Permission Locking Based on Database Role Changes," Int. J. Adv. Eng. Res., vol. 27, no. 1, pp. 1–9, 2024.

[95] A. B. Naeem et al., “Heart disease detection using feature extraction and artificial neural networks: A sensor-based approach,” IEEE Access, vol. 12, no.3, pp. 37349–37362, 2024.

[96] R. Tsarev, B. Senapati, S. H. Alshahrani, A. Mirzagitova, S. Irgasheva, and J. Ascencio, “Evaluating the effectiveness of flipped classrooms using linear regression,” in Data Analytics in System Engineering, Cham: Springer International Publishing, Switzerland, pp. 418–427, 2024.

[97] V. Attaluri, "Real-Time Monitoring and Auditing of Role Changes in Databases," Int. Numer. J. Mach. Learn. Robots, vol. 7, no. 7, pp. 1–13, Nov. 2023.

[98] V. Attaluri, "Securing SSH Access to EC2 Instances with Privileged Access Management (PAM)," Multidiscip. Int. J., vol. 8, no. 1, pp. 252–260, Dec. 2022.

[99] R. Boina, “Assessing the Increasing Rate of Parkinson’s Disease in the US and its Prevention Techniques”,” International Journal of Biotechnology Research and Development, vol. 3, no. 1, pp. 1–18, 2022.