Manchester Metropolitan University

Image Recognition and Classification for Road Signs with Uncertainty Estimation

Unit Code and Title:	6G7V0016 – Knowledge Representation and Reasoning
Group Members:	Aimen Lone - 22586132, Muhammad Seedat - 19090939, Tooba Zahid - 23721219
Assessment Set By:	Ismail Adeniran
Assessment ID:	1CWK100
Assessment Weighting:	100%
Assessment Title:	Decision Making System Report
Type:	Group/Individual
Weighting:	70%
Hand-in Deadline:	Friday 12th January 2024 21:00
Learning Outcomes assessed: 1. Explore and evaluate theorical issues underpinning the design of KBS. 2. Design and develop key elements of KBS reasoning systems and display an appreciation of the processes involved in building KBS. 3. Display a critical appreciation of KBS frameworks.	

Abstract

Using Convolutional Neural Networks (CNNs) and supervised learning, the project focuses on refining the interpretation of road signs, accompanied by uncertainty estimation to navigate image quality variations and communicate classification reliability.

The project adopts Imitation Learning, enabling the CNN to extract insights from pre-classified road sign images. It employs Maximum Likelihood Estimation to enhance accuracy by assigning probabilities to classifications. A user-friendly GUI is also developed with Gradio, to visualize predictions and associated uncertainty percentages.

The research successfully realizes its aim of creating a sophisticated road sign classification model, incorporating uncertainty estimation, and underscores the significance of decision-making approaches in enhancing model accuracy.

Table of Contents

1.0 Introduction

1.1 Project Overview

Road signs play a crucial role in ensuring road safety by conveying essential information regarding speed limits, potential hazards, and roadway regulations (Rosebrock, 2019).
The project will focus on creating a road sign detection model that will use convolutional neural networks to train and classify road signs and output an uncertainty estimation for the predicted classification. The project will use an existing dataset and employ supervised learning models. Within the context of decision-making, this project will address the challenge of interpreting the meaning of road signs, emphasizing the role of the algorithm in generating informed classifications. The inherent uncertainty stems from the diverse nature of road sign images, introducing variability that may affect the certainty of the algorithm in accurately classifying each sign. Therefore, the project will recognise the inherent uncertainties associated with image quality variations and will strive to quantify and communicate these uncertainties to users.

1.2 Approach Used

The approach outlined in the project report involves the use of Imitation Learning in conjunction with a Convolutional Neural Networks (CNN) to achieve precise road sign classification. The CNN is trained using pre-classified images of road signs to learn detailed features such as shapes, colours, and patterns, which allows it to adjust its parameters and improve accuracy. In addition, the project utilises a Maximum Likelihood approach to classify new images. In this approach, the CNN calculates probabilities for different road sign categories and selects the category with the highest probability as the final classification. This method guarantees the dependability of the system and offers valuable insights into the level of confidence in the predictions. Utilising CNNs for image processing and Maximum Likelihood estimation for decision-making is highly effective for the intricate task of road sign classification. This approach provides accurate classifications and also provides valuable information about the confidence levels associated with those classifications.

1.3 Aims and Objectives

Aim:

The aim of the report is to develop an advanced image recognition system that uses Convolutional Neural Networks (CNNs) to precisely categorise road signs. For improved system reliability, the project will aim to implement an advanced mechanism for estimating uncertainty.

Objectives:

- To research and select a road sign image dataset. To clean and preprocess the dataset.
- Choose a CNN algorithm with the most potential for accurate road sign classification.
- Customise and fine-tune the algorithm to meet road sign recognition needs.
- Train the chosen CNN algorithm on the prepared dataset for accuracy and robustness.
- To improve decision-making, use maximum likelihood estimation to evaluate picture categorization uncertainty and assign probability.
- Test the algorithm using unseen data to keep uncertainty low and acceptable. Display classification findings in graphs or tables for clarity.

2.0 Literature Review

2.1 Image Classification

Image classification is the process of categorising images into predefined groups or classes (Sanghvi, 2020). It constitutes a fundamental aspect of computer vision, wherein computers analyse each element of a photograph and allocate it to one or more categories based on observed characteristics and patterns (Odonohue, 2023). This entails training a machine learning model using labelled pictures to identify and differentiate various objects, scenes, or patterns within an image. In accordance with the learned patterns during training, the model becomes adept at assigning the most relevant label or category to new, unseen pictures (Das, 2020).

Temple matching was employed for Traffic sign detection and tracking by Torresen et al. (2004) and Greenhalgh et al. (2012). It has the benefits of being quick, simple, and precise (with a 90% accuracy rate on their own visual image dataset). The disadvantage of this technology is that it is extremely susceptible to noise and obstructions. Furthermore, it necessitates a different framework for each scale and orientation.

Another classification methodology is the genetic algorithm. It relies on a natural selection approach that is similar to biological evolution and was utilised in the early twentieth century (Safat, 2019). Armingol et al. (2001) and Hannan et al. (2008) applied this approach for traffic sign identification. These experiments demonstrated that this technology is effective in detecting traffic signs even when the signs have considerable form degradation or lighting problems. The downside of the genetic algorithm is its unpredictable work time and inability to ensure the optimal answer.

The random forest is another popular classification approach. It is a technique for machine learning that works by building many decision trees while training and then generating the class (Safat, 2019). This technique outperformed Support Vector Machine, and Multilayer Perceptron, and HOG-based classifiers in Greenhalgh et al. (2012) and Zaklouta et al. (2011), with the greatest precision and the shortest computing time. After analysing their individual dataset, the precision was roughly 94.2%, whilst the SVM performance is 87.8% and the MLP reliability is 89.2%. During a single classification, the SVM requires 115.87 ms, the MLP takes 1.45 ms, and the decision tree takes 0.15 ms.

2.2 Supervised Learning

Supervised Learning is an approach to machine learning in which computers acquire patterns from labelled training data to generate predictions on unseen data. The algorithm is trained on a dataset where the input data is paired with the correct output, as illustrated in Figure 1 (Murray, 2011).

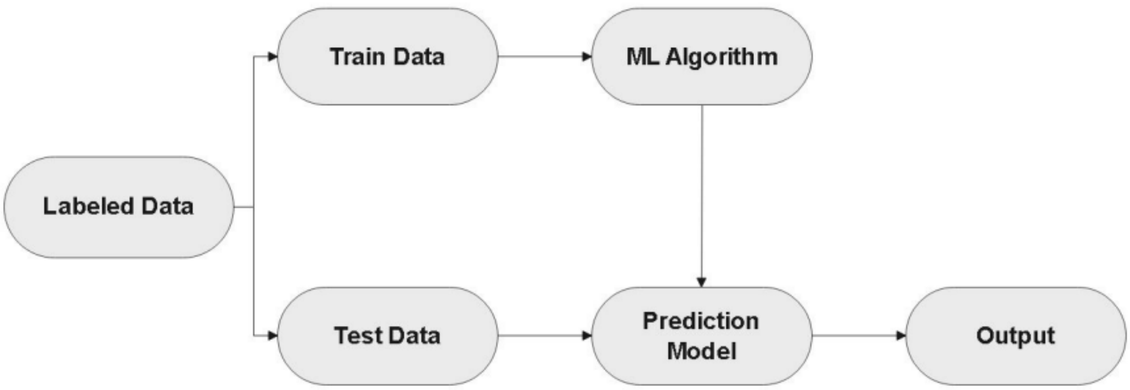

Figure 1 - Basic Architecture of Supervised Learning

This technique is commonly employed for road signs classification, which is a methodology that includes categorising and recognising various kinds of traffic signs on roadways utilising machine learning and computer vision (Salichs, 2022). The purpose is to correctly recognise and categorise diverse traffic signs, aiding autonomous cars in their navigation, supporting drivers by delivering pertinent data, and providing general roadway security by warning drivers to potential risks or legislative regulations on the road (Kondamari, 2021).

2.3 Road Sign Classification

Road sign classification assists autonomous cars in comprehending and following traffic laws, enabling them to recognise limitations on speed, stop signs, warning signs, and other road signs, making navigation more secure and effective (Rosebrock, 2019).

The most frequent methods rely on Support Vector Machines (SVM) (Yang, 2016), template matching, and presently on CNNs (Ruta, 2016).

Mogelmose and Li (2016) suggested using a convolutional neural network (CNN) to recognise and categorise speed limit signs in the United States. For the identification, they used an altered version of R-CNN, while for the classification, they employed a Cuda-convnet. They reported to have achieved a mean AUC of 93.89% for four classes (No Turn, Speed Limit, Stop, and Warning).

Gomez Moreno et al. (2005) and Timofte et al. (2010) studied the use of SVM for recognising speed limit signages. SVM combined with a Gaussian Kernel was employed to identify and recognise speed restriction signs. With a 90% efficiency rate, 134 blob pictures were retrieved from an image database.

In a recent work, Li et al. (2010) implemented a CNN by intentionally employing asymmetric kernels to optimise the performance. Furthermore, an inception module that combined the findings of two CNN branches along the channel axis was utilised to incorporate distinct geographical data. The CNN model, which was trained on the German Traffic Sign Recognition Benchmark, performed admirably, reaching 99.66% precision while remaining resilient (Serna, 2014).

2.4 Convolutional Neural Network

The Convolutional Neural Network (CNN) is a basic and highly sophisticated technique in the environment of traffic sign classification, serving an essential part in the advancement of self-driving automobile technologies (Satishkumar, 2022). CNNs are especially built for images identification tasks, and their use in road sign classification demonstrates their effectiveness in identifying complicated visual elements necessary to ensure secure and effective navigation (Credi, 2016). CNNs, as opposed to typical neural networks, use convolutional layers to apply filters to input photographs, enabling them to collect and analyse specific data such as borders, forms, and texturing. This style corresponds to the complex structure of road sign pictures, where diverse visual aspects indicate unique driving instructions (Wu, 2013).

Aghdam (2016) focused on the detection and classification of traffic signs using CNNs. Their convolutional layer algorithm assessed border, form, and texturing data for road sign recognition. He found that convolutional neural networks can detect road signs by analysing borders, contours, and texturing.

Figure 2. illustrates the CNN architecture, which is optimised for pattern recognition within images, efficiently learning spatial hierarchies of features.

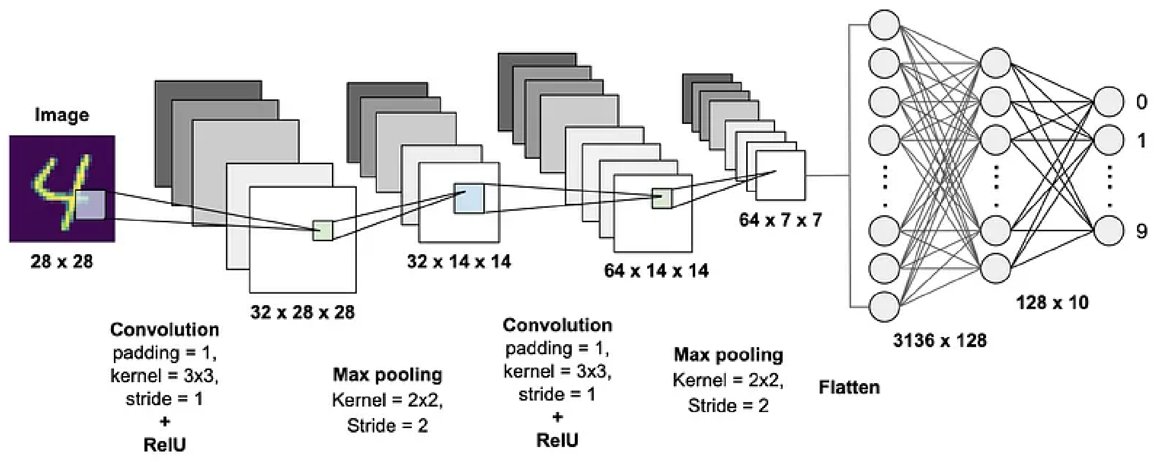

Figure 2- Convolutional and Pooling Layers

Luo (2018) trained CNN by showing it a vast set of annotated traffic sign images, improving its capacity to recognise sign category visuals. Their study found that training a CNN with a large set of labelled traffic sign images improves its capacity to recognise visual features for different sign categories.

Boujemaa et al. (2017) investigated the use of CNNs for traffic sign recognition, highlighting the significance of neural networks in deciphering the intricate structure of road sign images.

As the car meets new road signs while in execution, the trained CNN can generate precise predictions depending on its learnt characteristics, facilitating a real-time decision-making process, critical for the security and efficacy of self-driving automobiles (Pan, 2020).

Dhar (2017) and Agrawal (2018) highlighted how CNNs play a crucial role in developing resilient algorithms for traversing difficult traffic conditions, significantly advancing autonomous transportation systems. Researchers found that CNNs aid in developing reliable algorithms for navigating complex traffic conditions, paving the way for autonomous transportation systems.

The use of CNNs in traffic sign classification constitutes an evolutionary change, a significant step towards the realisation of less hazardous, more effective, and entirely autonomous modes of transportation (Wu, 2013).

2.5 Imitation Learning

Imitation learning has proven to be successful in various domains, including video games, autonomous driving, robotic simulations, and object manipulation. The recent enhancements have addressed various challenges, such as the ever-changing environment, frequent queries, and complex calculations. As a result, the system now achieves quicker convergence, greater resilience to noise, and more efficient learning processes (Verma,2022).

A simulator is designed for autonomous driving research and utilises imitation learning to train self-driving cars. The simulator combines computer vision, deep learning, and artificial intelligence to establish an authentic training environment for autonomous vehicles. Research has shown that incorporating both simulated and real data can significantly enhance the accuracy of AI models. Using simulated data can greatly decrease the amount of training needed on actual data to reach acceptable levels of accuracy (Ros, 2023).

Goyal (2022) researches and shows how to use advanced machine learning methods like CNNs to automatically classify and recognise traffic signs. This work is pertinent to the project as it focuses on using CNNs for traffic sign identification, a key part of the Imitation Learning technique.

Park (2022) proposes integrating vision and deep learning to guide a car to a drivable region in his paper titled "Look-ahead autonomous driving in semi-structured situations utilising imitation learning." To create a safe policy with a clear state action pattern linkage, imitation learning learns the vehicle's look-ahead point on a vision-based occupancy grid map. The proposed imitation learning method worked well in semi-structured situations.

2.6 Maximum Likelihood Estimation

Brownlee (2019) made the discussion on Maximum Likelihood Estimation (MLE) within the realm of machine learning. This text delves into density estimation and maximum likelihood estimation (MLE) as a fundamental framework for a range of machine learning algorithms, including deep learning neural networks. Brownlee's work highlights the importance of maximum likelihood estimation (MLE) in optimising parameters to improve the fit of models. This aligns with the paper's description of how MLE enhances neural network optimisation for challenging tasks such as image and speech recognition.

Agarwal (2023) offers a detailed examination of MLE. This text explores MLE as a fundamental statistical technique in data science and statistics, emphasising its significance in estimating parameters for different probability distributions. This guide is designed to be practical and applicable in improving the precision of predictive modelling in machine learning scenarios.

Rittmo (2021) is centred around image classification using the fashion-MNIST dataset and utilises maximum likelihood estimation (MLE). The methodology considers the dataset images as observations derived from multivariate Gaussian distributions. The project entails the establishment of a classifier using maximum likelihood and attains a training accuracy of 0.796, which, as previously stated, is considerably superior to chance. The methodology and results of this project align with the paper's description, emphasising the utilisation of MLE in image classification with specialised datasets such as fashion-MNIST.

3.0 Project Design and Implementation

The algorithm structure is based on an article by Gupta (2023: online) titled, "Traffic Signs Recognition using CNN and Keras in Python", which details the steps taken to build a model for traffic sign recognition using convolutional neural networks. The code and tools used to build the model in this paper were modified significantly from the article to achieve improved accuracy and better suit the aims and objectives of the paper.

3.1.0 Model Design

This section describes model building and visualization tools and libraries. The code's design and structure, including the data and CNN variant used to train the model, are described. The section also outlines the tools to deploy a user-friendly graphical user interface to visualize the model.

The python libraries used in this paper are:

- OpenCV
- NumPy
- Pandas
- TensorFlow
- Matplotlib
- Scikit-Learn
- Keras
- Pillow (PIL)
- Gradio

3.1.1 Dataset

The dataset chosen for the paper is the "German Traffic Sign Recognition Benchmark (GTSRB)", it contains 43 classes of various German traffic signs, and more than 50 000 images split into a training and testing sets. The training data contains 39 209 images, and the test data contains 12 630 images. This dataset was chosen because it is one of the largest datasets publicly available and it also contains a variety of images in different lighting conditions and of varying image quality and background (Stallkamp et al.,2012).

3.1.2 Python

"Python is a powerful, elegant programming language that is easy to read and to understand" (Yuill and Halpin, 2006: online). The programming language chosen for this paper was Python, the reason for which is because it is a widely popular programming language used in machine learning and is easily accessible and free to use. Additionally, Python has a vast number of resources and libraries that are available for free and are well maintained. The IDE used for the python codes was Visual Studio Code.

3.1.3 CNN Structure

Convolutional Neural networks consist of three main layers namely, the convolution layer, the pooling layer, and the fully connected layer (Guo et al., 2017). These layers can be modified, and additional layers can be added to enhance the accuracy of the model and make the model more efficient. The model used in this paper contains the following layers: The input layer, the convolution layers, the pooling layers, a flatten layer, and a fully connected layer.

The input layer is responsible for specifying the shape of the images being fed into the model to ensure they have the same shape as the training data. This is then followed by two convolutional layers with 32 filters each and a filter size of (3, 3), both using ReLU activation.

Thereafter, a MaxPooling layer with a pool size of (2, 2) reduces the pixel width and height of the image, keeping only the relevant information needed. Another set of two convolutional layers are then used this time applying 64 filters each with a filter size of (3,3), both using ReLU activation. A further MaxPooling layer follows with a pool size of (2,2) (Gholamalinezhad and Khosravi, 2020).

After the convolution and pooling layers, a flatten layer is added, which transforms the input into a 1D array to prepare for the Fully Connected Layer. The fully connected layer consists of: A Dense layer with 256 units and ReLU activation, a Dropout layer with a dropout rate of 0.5 to prevent overfitting, and a final Dense layer with 43 units to account for the 43 classes of images. The final dense layer uses softmax activation which is used to calculate the probability of each class (Basha et al., 2020).

The model was run for seven epochs which proved to be the ideal balance between accuracy and computational time. The images were then tested using the test data and the accuracy of the predictions calculated.

3.1.3 Approach Application

Two approaches were used as the basis of this paper, imitation learning and maximum likelihood estimation as discussed in section 1 and 2.

This paper uses imitation learning by feeding the model labelled photos. Thus, the model learns from human traffic sign recognition activities. The model then classifies and recognises photos using that information. Thus, by leveraging the past actions of humans from the training dataset, the model adopts an imitation learning approach.

The maximum likelihood approach was also applied to the model. This approach was used to calculate the probability of the predictions that the model made. To follow the maximum likelihood approach softmax activation and cross entropy loss were used. Softmax activation uses an exponential function to estimate the probabilities of each class to between 0 and 1 and the sum of all probabilities add up to 1 (Sharma et al., 2020).

"Cross-entropy loss refers to the contrast between two random variables. It measures the variables to extract the difference in the information they contain, showcasing the results." (365 Data Science, 2021: online). Cross entropy loss is usually used in conjunction with the softmax activation function to calculate how far the predicted class probability is from the actual probability is and adjust the weights accordingly. It works to minimise the loss of the model and improve accuracy (v7labs, no date: online)

As defined by Arora (2023: online), " Maximum Likelihood Estimation (MLE) is a statistical method used to estimate the parameters of a probability distribution that best explains the observed data", the softmax activation function and cross-entropy loss function fulfil the definition of the approach and aim to best estimate the probability of the classes of the image and maximise accuracy of the predicted class and minimising loss.

3.1.4 Deploying a GUI

In order to visualise the predictions in a user-friendly manner a GUI was created to upload and classify images along with displaying the predicted uncertainty percentage of each classification. The package used to achieve this was Gradio, Gradio is a python package that enables construction of a web application to deploy machine learning models. It is simple to build, and the model can be accessed by a link generated once the program is run.

3.2.0 Implementation

3.2.1 Data input Processing

The first step to building the model is to process the data to ensure it is compatible with the model. In order to assist in loading and processing the images the OpenCV library was utilised. The images in the train folder were loaded into a variable, they were then resized to 30 X 30 pixels using openCV's resize function. The images were then transformed into a Numpy array and their pixel values were scaled to be between 0 and 1 as this will make the model faster and more efficient.

Images are appended to a data array for storage after scaling to match the model. Class labels are appended to a labels array for future usage. The label and data arrays were transformed to NumPy arrays for better flexibility and data manipulation with functions.

After splitting the data, 80% is used for training and 20% for validation. The model will evaluate its performance on new data with validation data. The labels are one-hot encoded to transform categorical labels to numerical values for computation.

The data is ready for the model after these procedures. Implementing the model using this data is the next stage.

3.2.2 Model Implementation

The library used to build the CNN model was TensorFlow. TensorFlow is a powerful machine learning library that has functionality to define and build CNN models. In the model in this paper a sequential model was initialised which specifies a neural network passing through the layers in the order defined (D'Agostino, 2022: online). The layers of the model are then defined as discussed in section 3.1.3.

The model is compiled, and data is entered using model.fit once the layers are defined. The CNN model will use the NumPy array of images, target labels, and validation data. The number of epochs—the number of times the model will run on the data—is seven. The batch size, 32, determines how many samples will be sent through the network before updating the model's weights (Pramoditha, 2022: online).

After compilation, the model outputs loss and accuracy values for both the training and validation datasets across epochs, which are saved in anc. The model is retained in memory for future use and can be called in other programmes.

3.2.3 Model Testing

In order to test the accuracy of the word, the Test.csv file and the test folder in the dataset is used. These contain images that the model has not seen before and the images themselves have no attached label. The corresponding labels are stored in the test.csv file. The images are first extracted from their paths in the csv file and stored in an array. The corresponding labels are also extracted and stored in a separate array.
The images are then transformed using OpenCV and NumPy into the correct format for processing identical to in section 3.2.1. The model.predict method is then applied to the images to get the predicted classes for each image. The accuracy score is then calculated by comparing the predicted class labels to the actual labels that are stored. The model outlined in this paper received an accuracy score of 97.36%.

3.2.4 GUI Implementation

According to section 3.1.4, Gradio creates a GUI for model visualisation. The GUI lets users submit traffic sign images to run the CNN model and get the projected class name. In addition, the GUI will show the expected percentages of the top three classes.

For this, a second Python file is produced. Keras library loads the previously saved model into the file. To match class names to model label numerical values, a dictionary is built.

Next, a function processes the image and runs it through the model. First, the image is downsized to 30x30. Then, a NumPy array with pixel values between 0 and 1 is created. Processed images are run through the model using the predict method. The function returns all class labels and their predicted values.

To generate the GUI, a Gradio interface is established with the code above and an RGB image uploaded by the user. The final predicted class and top 3 highest predicted classes will be displayed.

4.0 Evaluation and analysis of results

The model achieved a high level of accuracy during the training process achieving up to 98.93% accuracy as shown by Figure 4.

```
Epoch 1/7
981/981 [==============================] - 35s 33ms/step - loss: 0.9356 - accuracy: 0.7385
Epoch 2/7
981/981 [==============================] - 37s 38ms/step - loss: 0.1410 - accuracy: 0.9571
Epoch 3/7
981/981 [==============================] - 39s 40ms/step - loss: 0.0821 - accuracy: 0.9755
Epoch 4/7
981/981 [==============================] - 31s 31ms/step - loss: 0.0578 - accuracy: 0.9831
Epoch 5/7
981/981 [==============================] - 27s 27ms/step - loss: 0.0492 - accuracy: 0.9849
Epoch 6/7
981/981 [==============================] - 28s 29ms/step - loss: 0.0412 - accuracy: 0.9869
Epoch 7/7
981/981 [==============================] - 31s 32ms/step - loss: 0.0356 - accuracy: 0.9893
```

Figure 4

The Graph in Figure 5 also shows the accuracy increasing through each epoch which shows that the model was learning and improving. The graph in Figure 6 displays the loss through the epochs which shows that it achieved minimal loss and with each epoch the loss reduced.

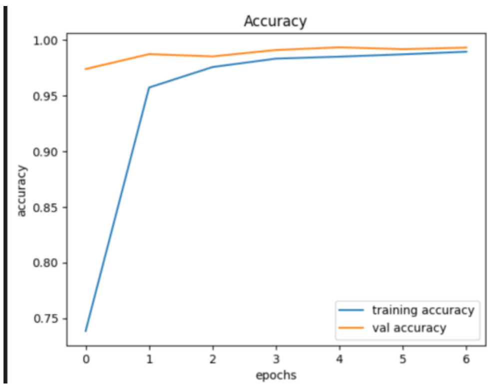

Figure 5

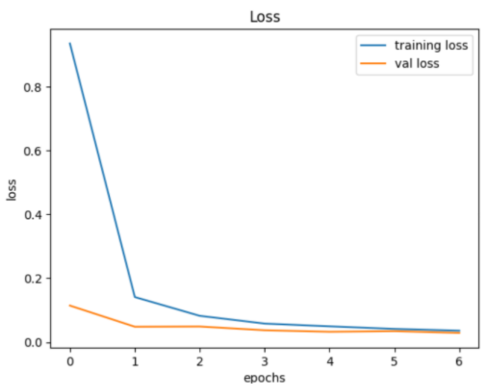

Figure 6

The accuracy of the test set is also extremely strong as it achieved a 97.36 % accuracy on test data as shown by Figure 7. This shows that the imitation learning approach and maximum likelihood estimation approach were successful in improving the model's accuracy and correctly predicting most images.

```
395/395 [==============================] - 3s 9ms/step
Accuracy on test set: 97.36%
```

Figure 7

As shown in Figure 8 the model correctly classified the image even considering that it was of poor quality and blurry.

Figure 8

There is room for improvement, as in some instances the model does not classify the image correctly, such as, in Figure 9, the image was classified as Yield when it should be Ahead only.

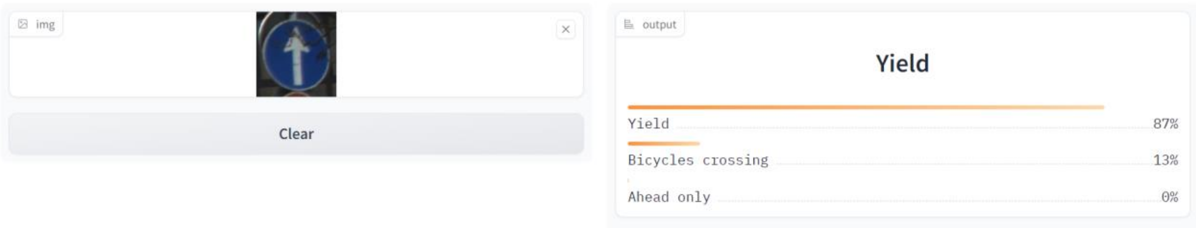

Figure 9

The project results show that by large the model and approach used were successful in classifying the images, but there is room for improvement and experimentation with other approaches that can be used for this problem.

5.0 Conclusion

The aim of the paper was to create a Road sign classification model and include an uncertainty estimation calculation, achieved to a high standard. The project was able to successfully present a CNN model which predicted road sign classifications with a high degree of accuracy and furthermore, presented a GUI to observe the uncertainty estimations. Although the paper was not purely a decision-making problem it incorporated decision making approaches such as imitation learning and maximum likelihood estimation to enhance the accuracy of the CNN model.

In conclusion, the paper achieved its main aims and objectives while incorporating decision making approaches. The paper also used industry leading methods and tools to create a model and visualize the model for the problem defined.

Group Contributions

Muhammad Seedat – Coded the CNN model and GUI and researched and wrote the Project Design and Implementation, Evaluation and Analysis of results and Conclusion Sections.

Aimen Lone – Conducted research on Image Classification, Supervised Learning, and Road Sign Classification. Wrote the abstract and introduction overview.

Tooba Zahid – Researched on CNN model, Maximum Likelihood Estimation, Imitation Learning. Wrote Approached Used, Aims and Objectives.

Reference List

365 Data Science. (2021). What Is Cross-Entropy Loss? [online] [Accessed on: 15th December 2023] https://365datascience.com/tutorials/machine-learning-tutorials/cross-entropy-loss/.

Aghdam, H. H., Heravi, E. J. and Puig, D. (2016) A Practical Approach for Detection and Classification of Traffic Signs Using Convolutional Neural Networks. *Science Direct*, 84, pp. 97-112. [Online] [Accessed on 19th December 2023] DOI: https://doi.org/10.1016/j.robot.2016.07.003

Agrawal, P. (2018) 'A Survey on Road Traffic Sign Recognition System Using Convolutional Neural Network.' *International Journal of Current Engineering and Technology*, 8(1), pp. 104-108. [Online] [Accessed on 20th December 2023] DOI: DOI: 10.14741/ijcet/v.8.1.21

Arora, A. (2023). Logistic Regression: Maximum Likelihood Estimation & Gradient Descent. Medium. [online] [Accessed on: 20th December 2023] https://medium.com/@ashisharora2204/logistic-regression-maximum-likelihood-estimation-gradient-descent-a7962a452332#:~:text=The%20softmax%20function%20applies%20a.

Basha, S.H.S., Vinakota, S.K., Pulabaigari, V., Mukherjee, S. and Dubey, S.R. (2021). AutoTune: Automatically Tuning Convolutional Neural Networks for Improved Transfer Learning. Neural Networks, 133, pp.112–122.

Bi, Y., Xue, B., Zhang, M. (2021) *Genetic Programming for Image Classification: An Automated Approach to Feature Learning*. [Online] New Zealand: Springer International Publishing. [Accessed on 5th October 2023] https://www.google.co.uk/books/edition/Genetic_Programming_for_Image_Classifica/sYYbEAAAQBAJ?hl=en&gbpv=1&kptab=overview

Boujemaa, K. S., Berrada, I., Bouhoute, A. and Boubouh K. (2017) 'Traffic Sign Recognition Using Convolutional Neural Networks.' *International Conference on Wireless Networks and Mobile Communications,* pp. 1-6. [Online] [Accessed on 20th December 2023] DOI: 10.1109/WINCOM.2017.8238205

Bouti, A., Mahraz, M. A. and Tairi, H. (2019) 'A Robust System for Road Sign Detection and Classification Using LeNet Architecture Based on Convolutional Neural Network.' Soft Computer, 24, pp. 6721-6733. [Online] [Accessed on 17th December 2023] DOI: https://doi.org/10.1007/s00500-019-04307-6

Brownlee, J. (2019) A Gentle Introduction to Maximum Likelihood Estimation for Machine Learning. Machine Learning Mastery. [Online] [Accessed on 8th January 2024] https://machinelearningmastery.com/what-is-maximum-likelihood-estimation-in-machine-learning/

Chuan Fei Guo, Pleiss, G., Sun, Y. and Weinberger, K.Q. (2017). On Calibration of Modern Neural Networks. Cornell University. Proceedings of the 34th International Conference on Machine Learning.Vol.70.6th-11th August 2017. Precup, Doina and Teh, Yee Whye. (eds.) PMLR, pp 1321-1330.

Credi, J. (2016) *Traffic Sign Classification with Deep Convolutional Neural Network*. Masters' Thesis in Complex Adaptive Systems. Chalmers University of Technology.

D'Agostino, A. (2023). Get started with TensorFlow 2.0 — Introduction to deep learning. Medium. [online] [Accessed on: 25th November 2023] https://towardsdatascience.com/a-comprehensive-introduction-to-tensorflows-sequential-api-and-model-for-deep-learning-c5e31aee49fa#:~:text=The%20TensorFlow%20APIs%20are%20based.

Das, R. (2020) *Content-Based Image Classification: Efficient Machine Learning Using Robust Feature Extraction Technique*. [Online] 1st ed., Florida: CRC Press. [Accessed on 5th October 2023] https://www.google.co.uk/books/edition/Content_Based_Image_Classification/gfcFEAAAQBAJ?hl=en&gbpv=1&kptab=overview

Dhar, P., Abedin, M. Z., Biswas, T. and Datta, A. (2017) 'Traffic Sign Detection – A New Approach and Recognition Using Convolutional Neural Network.' *IEEE Transactions on Intelligent Transportation Systems*, pp. 416-419. [Online] [Accessed on 20th December 2023] DOI: 10.1109/R10-HTC.2017.8288988

Geeks For Geeks. (2023) *Supervised and Unsupervised Learning*. [Online] [Accessed on 10th October 2023] https://www.geeksforgeeks.org/supervised-unsupervised-learning/

Gholamalinezhad, H. and Khosravi, H. (2020). Pooling Methods in Deep Neural Networks, a Review. [online] [Accessed on: 5th January 2024] DOI: https://doi.org/10.48550/arXiv.2009.07485.

Gillig, A. S. (2023) *Unsupervised Learning*. TechTarget. [Online] [Accessed on 5th October 2023] https://www.techtarget.com/searchenterpriseai/definition/unsupervised-learning

Greenhalgh, J., Mirmehdi, M. (2012) 'Real-time Detection and Recognition of Road Traffic Signs.' *International IEEE Conference on Intelligent Transportation Systems*, *13*, pp. 1498–1506.

Gupta, S. (2021*). Traffic Signs Recognition using CNN and Keras in Python*. [online] [Accessed on 25th November 2023] https://www.analyticsvidhya.com/blog/2021/12/traffic-signs-recognition-using-cnn-and-keras-in-python/.

Hannan, M., Hussain, A., Samad, S., Ishak, K. and Mohamed, A. (2008) 'A Unified Robust Algorithm for Detection of Human and Non-human Object in Intelligent Safety Application.' *World Academic Science Engineering Technology International Journal of Computer and Information Engineering*, *2*(11), pp. 3838–3845.

IBM. (2023) What Is Unsupervised Learning? [Online] [Accessed on 7th October 2023] https://www.ibm.com/topics/unsupervised-learning

J. Stallkamp, M. Schlipsing, J. Salmen, and C. Igel. The German Traffic Sign Recognition Benchmark: A multi-class classification competition. In Proceedings of the IEEE International Joint Conference on Neural Networks, pages 1453–1460. 2011.

Kondamari, P. S., Itha, A. (2021) *A Deep Learning Application for Traffic Sign Classification*. Bachelor of Science in Computer Science. Blekinge Institute of Technology.

Lavanya, K., Shakthi, P. V., Pavithra, S. T. and Prabha, B. (2023) 'Traffic Sign Classification Using Convolutional Neural Network.' *International Journal of Research Thoughts,* 11(5), pp. 476-484.

Li, S., Song, W., Fang, L., Chen, Y., Ghamisi, P. and Benediktsson, A. (2019) 'Deep Learning for Hyperspectral Image Classification: An Overview.' *IEEE Transactions on Geoscience and Remote*

Sensing, 9(57), pp. 6690-6709. [Online] [Accessed on 5th October 2023] DOI: 10.1109/TGRS.2019.2907932

Luo, H., Yang, Y., Tong, B., Wu, F. and Fan, B. (2018) 'Traffic Sign Recognition Using a Multi-Task Convolutional Neural Network.' *IEEE Transactions on Intelligent Transportation Systems*, 19(4), pp. 1100-1111. [Online] [Accessed on 20th December 2023] DOI: 10.1109/TITS.2017.2714691

Mishra, J. and Goyal, S. (2022) 'An Effective Automatic Traffic Sign Classification and Recognition Deep Convolutional Network.' *Multimedia Tools and Applications: An International Journal,* 81, pp. 18915–18934. [Online] [Accessed 17th November 2023] DOI: https://doi.org/10.1007/s11042-022-12531-w

Mishra, J., Goyal, S. (2022) 'An Effective Automatic Traffic Sign Classification and Recognition Deep Convolutional Network.' *Multimedia Tools and Applications: An International Journal*, 81, pp. 18015-18934. [Online] [Accessed on 14th December 2023] DOI: https://doi.org/10.1007/s11042-022-12531-w

Murray, R. F. (2011) 'Classification Images: A Review.' *Journal of Vision*, 11(5):2, pp. 1-25. [Online] [Accessed on 5th October 2023] DOI: 10.1167/11.5.2.

Nandi, D., Saif, A. S., Prottoy, P., Zubair, K. M., Shubho, S. A. (2018) Traffic Sign Detection based on Color Segmentation of Obscure Image Candidates: A Comprehensive Study. *International Journal of Modern Education Computer Science*, *10*, pp. 35-46. [Online] [Accessed on 7th January 2024] DOI: 10.5815/ijmecs.2018.06.05

Obaid, K. B., Zeebaree, S. R. and Ahmed, O. M. (2020) 'Deep Learning Models Based on Image Classification: A Review.' *International Journal of Science and Business*. 4(11), pp. 75-81. [Online] [Accessed on 5th October 2023] DOI: 10.5281/zenodo.4108433

Olaide, A., Naghdy, G. and Todd, C. (2014) 'Unsupervised Classification of Images: A Review.' *International Journal of Image Processing*, 8(5), pp. 325-342.

Pan, Y., Kadappa, V. and Guggari, S. (2020) 'Identification of Road Sign Using Novel Convolutional Neural Network.' *Science Direct*, 1, pp. 319-337. [Online] [Accessed on 20th December 2023] DOI: https://doi.org/10.1016/B978-0-12-819443-0.00015-5

Pramoditha, R. (2023). Determining the Right Batch Size for a Neural Network to Get Better and Faster Results. Data Science 365.[online][Accessed on: 2nd January 2024] https://medium.com/data-science-365/determining-the-right-batch-size-for-a-neural-network-to-get-better-and-faster-results-7a8662830f15#:~:text=Default%20value%20%3D%2032&text=But%20we%20are%20limited%20to.

Pykes, K. (2023) *Introduction to Unsupervised Learning*. Datacamp. [Online] [Accessed 7th October 2023] https://www.datacamp.com/blog/introduction-to-unsupervised-learning

Rana, Md. M. and Hossain, K. (2021). 'Connected and Autonomous Vehicles and Infrastructures: A Literature Review'. International Journal of Pavement Research and Technology, 16, pp.264-284

Reddy, P. M., Jindam, S., Sayeed, U. M. and Reddy A. V. (2021) 'Recognition of Traffic Sign Using CNN and Deep Learning.' *International Journal of Latest Technology in Engineering, Management & Applied Science*, 10(12), pp. 21-25.

Rittmo, J. Ö. (2021) 'Hands-On ML - Project 2: Image Classification in Python Using Maximum Likelihood.' Jonathan Ö. Rittmo's Blog. [Online] [Accessed on 7th January 2024] https://jorittmo.rbind.io/post/2021-11-27-hands-on-ml-project-2-image-classification-fashion-mnist-using-maximum-likelihood/

Rosebrock, A. (2019) *Traffic Sign Classification with Keras and Deep Learning*. [Online] [Accessed on 17th November 2023] https://pyimagesearch.com/2019/11/04/traffic-sign-classification-with-keras-and-deep-learning/#:~:text=Traffic%20sign%20classification%20is%20the,to%20build%20%E2%80%9Csmarter%20cars%E2%80%9D

Safat, B. W., Majid, A. A., Mahammad, A. H., Aini, H., Salina, A. S., Pin, J. K. and Muhammad, B. M. (2019) 'Vision-Based Traffic Sign Detection and Recognition Systems: Current Trends and Challenges.' *Sensors*, 19(9) [Online] [Accessed on 7th January 2024] DOI: https://doi.org/10.3390/s19092093

Saiteja, T., Kumar, K. R. (2021) 'Traffic Sign Recognition and Classification using CNN.' *International Research Journal of Engineering and Technology*, 8(9), pp. 1924-1927.

Salichs, M. A. (2014) Traffic sign detection for driver support systems. *Research Gate*.

Sanghvi, K. (2020) *Image Classification Techniques*. Medium. [Online] [Accessed on 5th October 2020] https://medium.com/analytics-vidhya/image-classification-techniques-83fd87011cac

Satishkumar, V. E., Sharmila, C., Santhiya, S., Poongundran, M., Sanjeet, S. and Pranesh, S. (2023) 'Convolutional Neural Networks for Traffic Sign Classification Using Enhanced Colours.' *Communications in Computer and Information Science*, 1719, pp. 34-43. [Online] [Accessed on 17th December 2023] DOI: https://doi.org/10.1007/978-3-031-27622-4_4

Sharma, S., Sharma, S. and Athaiya, A. (2020). ACTIVATION FUNCTIONS IN NEURAL NETWORKS. International Journal of Engineering Applied Sciences and Technology, 04(12), pp.310–316. [Online] [Accessed on: 3rd January 2024] https://www.ijeast.com/papers/310-316,Tesma412,IJEAST.pdf

Stallkamp, J., Schlipsing, M., Salmen, J. and Igel, C. (2012). 'Man vs. computer: Benchmarking machine learning algorithms for traffic sign recognition'. Neural Networks, 32, pp.323–332.

Torresen, J., Bakke, J. W., Sekanina, L. (2004) 'Efficient recognition of speed limit signs.' *International IEEE Conference on Intelligent Transportation Systems*, pp. 652–656. [Online] [Accessed on 7th January 2023] DOI: 10.1109/ITSC.2004.1398978

Triki, N., Karray, M. and Ksantini, M. (2023) 'A Real-Time Traffic Sign Recognition Method Using a New Attention-Based Deep Convolutional Neural Network for Smart Vehicles.' *Applied Science*, 13(8), pp. 1-21. [Online] [Accessed on 17th December 2023] DOI: https://doi.org/10.3390/app13084793

Vidhya, A. (2023) Maximum Likelihood Estimation - A Comprehensive Guide. [Online] [Accessed on 8th January 2024] https://jorittmo.rbind.io/post/2021-11-27-hands-on-ml-project-2-image-classification-fashion-mnist-using-maximum-likelihood/

Wu, Y., Liu, Y., Li, J., Liu, H. and Hu, X. (2013) Traffic Sign Detection Based on Convolutional Neural Network. *Proceeding of International Joint Conference on Neural Network,* pp. 212-217. [Online] [Accessed 19th December 2023] DOI: 10.5220/0010567100003161

www.v7labs.com. (n.d.). Cross Entropy Loss: Intro, Applications, Code. [online] [Accessed on: 5th January 2024] https://www.v7labs.com/blog/cross-entropy-loss-guide#:~:text=Entropy%20is%20a%20measure%20of.

Yuill,S. And Halpin,H.(2006).Python.[Online] [Accessed on 10th December 2023] https://citeseerx.ist.psu.edu/document?repid=rep1&type=pdf&doi=1f2ee3831eebfc97bfafd514ca2abb7e2c5c86bb

Zaibi, A., Ladgham, A. and Sakly, A. (2021) 'A Lightweight Model for Traffic Sign Classification Based on Enhanced LeNet-5 Network.' *Journal of Sensors*, 2021, pp. 1-13. [Online] [Accessed on 19th December 2023] DOI: https://doi.org/10.1155/2021/8870529

Zaklouta, F. and Stanciulescu, B. (2011) 'Real-time Traffic Sign Recognition Using Spatially Weighted HOG trees.' *International IEEE Conference on Intelligent Transportation Systems*, pp. 61–66.

Zheng, B., Verma, S., Zhou, J., Tsang, I., & Chen, F. (2021) Imitation Learning: Progress, Taxonomies and Challenges. arXiv.org. [Online] [Accessed on 8th January 2024] https://arxiv.org/abs/2106.12177

Appendix

The code used in the project can be accessed by the following link: https://github.com/moma049/ImageClassification