Can Data Augmentation Improve the Ability of CNN Models to Classify Obstructed Traffic Signs?

Tudor Haba
2420645

Table of Contents

1 Introduction *3*

1.1 Motivation 3

1.2 Aim and Objectives 4

2 Background Research *5*

2.1 Neural Networks 5

2.2 Convolutional Neural Networks 6

2.3 Data augmentation 8

2.4 Traffic Sign Detection and Recognition (TSDR) System 10

3 Project Plan *11*

3.1 Deliverables 11

3.2 Gantt Chart 11

3.3 Tools and Technologies 12

4 Bibliography *13*

1 Introduction

1.1 Motivation

Convolutional neural network (CNN) models are a good way to identify and classify images based on the features the model learns from them. They have proven to work very well in multiple real-world industries such as medical and more recent in the autonomous vehicle industry [1][2]. The autonomous vehicle industry is a fast-growing industry gaining a lot of interest in the media, especially since some manufacturers such as Tesla are already using their autopilot vehicles on roads in America. Although, some other manufacturers such as Waymo are using different technologies for their self-driving taxis, Tesla is known to be using CNN models as part of their decision-making software [3]. There is a clear rise in self-driving vehicle technologies that rely on systems like CNNs in the background, which process data from outside cameras that gather environment information from around them, to determine appropriate driving actions. Recent news show that the UK government is considering allowing testing of autonomous vehicles on the UK roads [4]. However, this technology must be safe enough to be on the road for the safety of pedestrians and other people on the road. The road could have a lot of variants that could cause some indecisiveness in CNN models such as rainy weather making it harder to classify traffic signs. Research from [2] has already looked at CNN models used in autonomous vehicles however, they've decided to use data augmentation to increase the training dataset size.

This project aims to train two CNN models on different datasets and compare the accuracies between them on some real-world based scenarios that have been created using data augmentation. One of the models will be trained with clear data of traffic signs, the perfect scenario that would be occurring on the roads, while the other model will be trained with an augmented dataset of traffic signs that adds more data with imperfections. Differently from [2] data augmentation will not only be used to increase the training dataset size but it will also be used to create some obstructed traffic signs. This aims for a more robust system at detecting real-life scenarios thus more appropriate to be used on roads.

1.2 Aim and Objectives

The overall aim of this project is to compare and evaluate whether data augmentation can help convolution neural network (CNN) models into more accurately classifying partially obstructed traffic signs based on real world scenarios. Analysing the accuracies of CNN models that have been trained with and without data augmentation by testing on unseen data.

The objectives of the project are:

- Design and implement a convolutional neural network (CNN) model capable to classify traffic signs, aiming to achieve a model accuracy of above 90% on the unseen test dataset.
- To develop and apply different data augmentation techniques on the image dataset. The aim is to produce real world scenarios from the available dataset that would make the model focus more on each feature of a traffic sign. Some of the scenarios that would be implemented are foggy weather, traffic signs being obstructed by different objects and images taken at different times of the day.
- Design and implement a new CNN model specifically trained on the augmented dataset, aiming to achieve a moderately high accuracy of above 70% for unseen test data. The augmented dataset is expected to increase the challenge to which the model is able to classify the dataset thus proving a more realistic accuracy in a real-world setting.
- Test both CNN models, the one not trained on the augmented dataset and the one trained on the augmented dataset, on unseen data from the augmented dataset. Analyse the results and compare the accuracies with which the models classify real world scenario data and evaluate whether a neural network's fault tolerance would allow it to correctly classify the images even though the neural network model has not been trained on the augmented dataset.

2 Background Research

2.1 Neural Networks

A neural network (NN) is a machine learning model that tries to simulate the way a human brain would work to process information [5]. A NN is made up of different layers, starting with the input layer, one or more hidden layers which decide the outcome of the data and a final output layer. A traditional neural network processes the information through the hidden layers of interconnected nodes all of which have a weight and a threshold value. Each node will then take the data and apply the weighting to it, if the resulting output is above the threshold value then the node is called an active node and it can pass data down to the other layers, Fig 1. Some neural network architectures do not pass the data forward if it has not met the threshold value while others replace the threshold value by an activation function that calculates the output of an node and passes it forward[5][6].

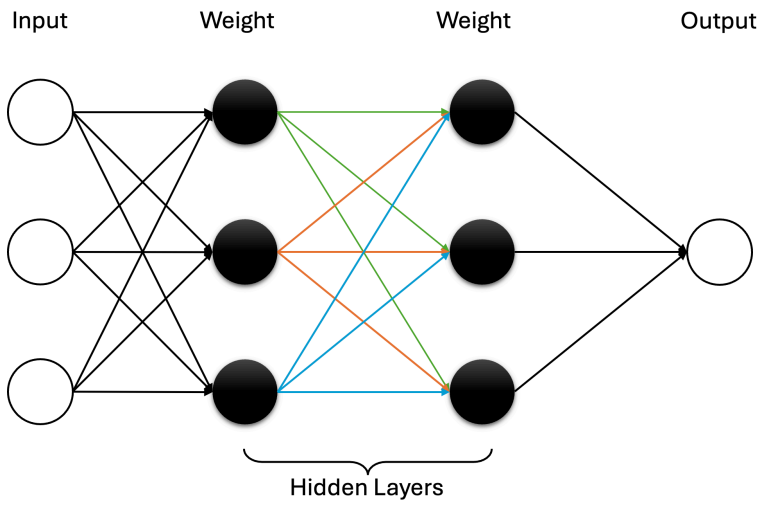

Fig 1. Simple representation of a neural network

The way a NN works is by providing it with some input data and applying the formula from Fig 2, with the resulting outcome getting passed into each node and checked against the threshold value, which typically has a value between 0 and 1, to determine which of the nodes will get to pass the data forward. This happens for every layer until we get an output set of data. Neural networks are good at learning patterns from datasets and they are good at improving fault tolerance over time, meaning that small errors in data do not have an effect to the output [5]. The project will make use of fault tolerance to initially create a model which recognises traffic signs and then test the model with a different dataset that introduces a various number of faults, seeing how well the model is at tolerating inaccuracies in the images. The project intends to use a specialised neural network, a convolution neural network (CNN), to process the image datasets and to extract the features of the images in order to identify the traffic signs.

$$\sum w_i x_i + bias = w_1x_1 + w_2x_2 + w_3x_3 + bias$$

$$output = f(x) = 1 \text{ if } \sum w_1x_1 + b >= 0; \ 0 \text{ if } \sum w_1x_1 + b < 0$$

Fig 2. Formula to calculate the output of a neural network node from its input

2.2 Convolutional Neural Networks

Convolutional neural networks (CNNs) are a type of neural network focused around solving deep learning problems specialising in pattern recognition. CNNs are great at extracting features out of image and audio input datasets and classifying them. They have been making a massive advancement in fields such as facial recognition and autonomous driving [7]. CNNs are complex neural networks made up of an input layer, one or more convolution layer followed by a pooling layer and more fully connected layers (FC) before producing an output, Fig 3.

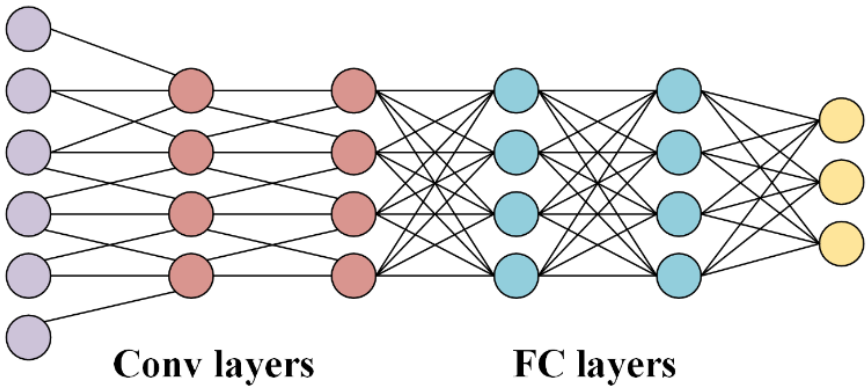

Fig 3. Example of a CNN model with convolutional layers and FC layers [7]

Convolutional layers are made up of filters. A filter or a feature detector is a matrix, Fig 4, that goes on top of the image and multiplies the pixel values from top left to bottom right until a new matrix is produced [7][8]. The resulting matrix is a score of how closely the pixels in the image match the pattern of the filter and is the activation map of the given image. The role of the filter is to find matching pattern in the image while at the same reducing the amount of data the network has to process. Using different feature detectors produces different activation maps which can be used by the CNN to identify different features in an image. The activation maps combined together produce a convolution layer that are used by the model to understand larger parts of the image [8].

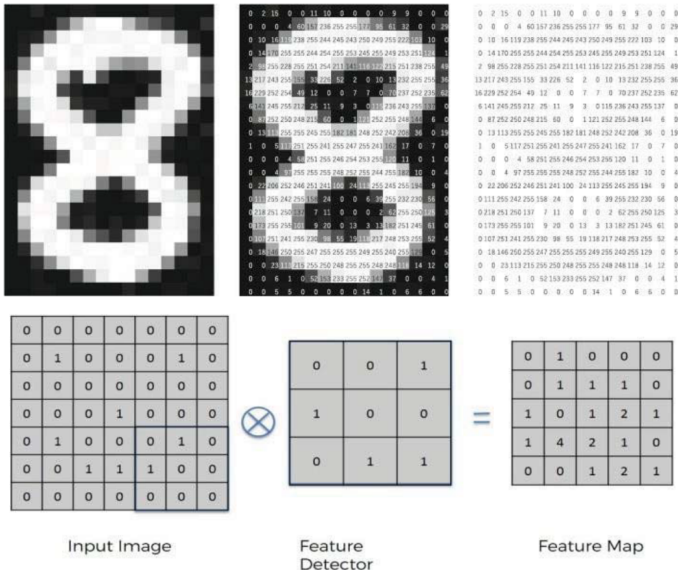

Fig 4. Detecting features out of an image using a feature detector matrix [8]

It is usual for CNNs to use a pooling layer after every convolution layer. A pooling layer reduces the size of the image and can be achieved by many different ways such as max pooling, average pooling, stochastic pooling, spatial pyramid pooling. Max pooling being the most commonly used in CNNs. Pooling works similarly to how the filters work, a matrix is slid on top of the image and depending on the method chosen a result is obtained, however, the pool filter does not cross on pixels that it has already worked on, thus being able to reduce the size of the image. A 2 x 2 max pooling for example, Fig 5, will take the maximum pixel value and only this value will be now present in the resulting feature map. This way pooling only keeps the important features of an image discarding the rest and reducing the size of the image for the next layers thus reducing the amount of time this will take overall [8] [9]. A CNN usually works by having a convolutional layer followed by a pooling layer and this is repeated until the resulting output out of a pooling layer is matrix fed into the FC layers which decide how to classify the image. Pooling also helps to resolve the overfitting problem that could happen to some models. This is where the model learns the training dataset and gives high accuracies only because it has memorised the images but fails to find patterns in new unseen images, resulting in poor model performance [10]. In most CNNs the data is flattened before being passed into the fully connected layers. This means that from a 2D or a 3D array the feature map gets converted into a 1D vector that can be understood by traditional neural network. The final flattened vector only contains the most import feature of the image [1]. Once the vector gets passed into the fully connected layers, the NN can make a decision about what it thinks the data should be classified as based on the learned features.

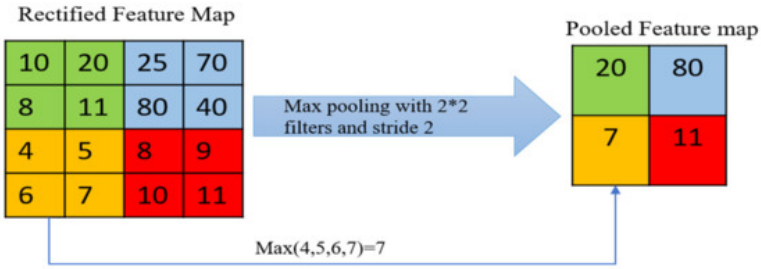

Fig 5. Max pooling being applied to a feature map [9]

Two CNNs models will be used in my project. The first model will be trained to identify traffic signs, and all the training data will have clean, unobstructed traffic signs. To test the model the dataset will be split into 70/30, with 70% of the data used for training and the rest of 30% used to test how accurate the model is. The second model will be trained with an augmented dataset that aims to introduce some obstruction in the images. Different architectures for the models will be tested until both models have a high accuracy of above 90% for the one not trained on the augmented data and 70% for the augmented data trained model.

2.3 Data augmentation

Data augmentation is a technique to artificially generate more data to train a model on from an already existing dataset. It is beneficial in the case the dataset is not large enough to train the model or if it would take a long time to gather new data and it improves the overall performance of the model training on it. The way data augmentation works is by taking existing data and applying different convolutions and filters to modify the data and produce new samples. The new samples that are being produced should have the same corresponding label as the input sample [11][12]. There are a number of techniques that could be used to augment a set of data in the case of images:

- Geometric transformations are a set of operations that include image rotation, image scaling and flipping. Image scaling being the most useful one for the case of traffic signs, it could help the model by zooming in and out of an image giving a more realistic interpretation of how a human would perceive a traffic sign at different distances [13].
- Colour augmentation allows for a shift in the image's colour or a change in brightness [11]. The change in an image's brightness could mimic the time of day the picture was taken, forcing the model to focus on specific feature of a traffic sign rather than its colours.
- Kernel filters are convolutional operations that could be applied to an image to highlight certain features or hide them. A Gaussian blur, Fig. 6, could be applied to an image to hide some of its distinctive feature and make it more challenging for a model to identify thus training it on more niche outliers [13].

- Random erasing, this is when random parts of the image get masked by rectangles in order to reduce the number of features, Fig. 6. This in result will enhance the performance of the model by being able to recognise and classify some of the harder images with missing information [11].

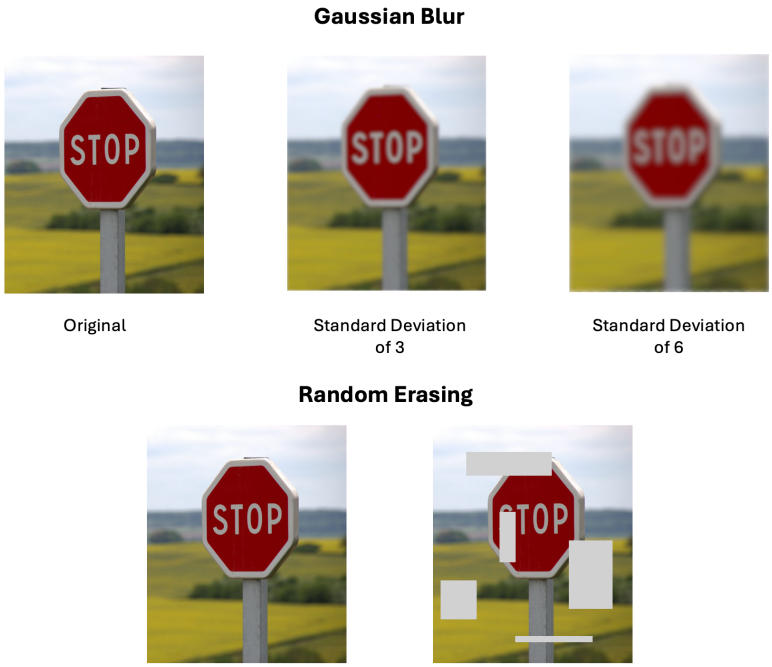

Fig 6. Data augmentation techniques visualised

The project would make use of a combination of data augmentation techniques to modify the dataset the second CNN model will be trained on. Using a combination of one or more techniques the dataset could achieve some more unique real-life scenarios and test whether the model can identify the traffic signs with real life circumstances. One of the real-life events the project wants to train and test the model on would be foggy weather. Using the filters that could easily be simulated by making use of the Gaussian blur and potentially making use of image scaling to make the traffic sign appear further away. Another scenario the project wants to test would be partial sign recognition. Using the random erasing which could simulate tree branches being in the way of the sign or the traffic sign being dirty or defaced and harder to read. Once the augmented datasets are generated and the new data is available, it is intended to split the dataset into 70/30 with 30% being kept for testing purposes. Out of the test data the same image will be fed into both models, the one not trained on the augmented dataset and the one that is specifically trained with the new dataset. This is done to make a comparison between the accuracies and analyse what the models classifies the data as.

2.4 Traffic Sign Detection and Recognition (TSDR) System

There have been a number of previous research papers looking at the traffic sign recognition problem. One of them being [2] where they proposed to improve on an already existing 7 layers CNN model called LeNet-5. The original model was made up of three convolutional and three subsampling layers followed by the fully connected layers producing the output. The model they proposed contained 12 layers with additional work on image preprocessing before feeding the data into the model to train on. Part of the image preprocessing techniques they suggested were colour to grayscale conversion, histogram equalization, unsharp masking, and normalization followed by data augmentation to increase the available dataset. While this project has a similar idea to the project proposed in this paper, the way data augmentation will be used is different. Where in [2] they used it to increase the number of images, the intention of this project would be to create new images containing some form of obstructions. Not only would this increase the size of the dataset, but the aim is to generate some images closer to real world data and harsher environment to improve the robustness of the CNN model. The project was successful with high accuracies in detecting and classifying the traffic signs, of above 99% [2]. As future work they have identified a need to create a model trained on different weather conditions and during different times of the day. This is how the proposed project in this paper intends to digitally alter images to create some of the scenarios stated in the above future work.

The work of [14] has identified multiple current challenges that TSDR systems face. Some of the challenges identified by the paper are variable lighting conditions. Traffic signs are harder to detect and classify when the lighting conditions vary and make it harder to see the sign in the image. This is partly due to the fact that during the pre-image processing phase approached in [14], they have used colour segmentation to reduce the features of an image. This combined with poor lighting conditions will also affect the features a model can identify from an image potentially deleting some of them. The paper has also identified that the state in which a traffic sign is in affects the capacity for a model to learn its features. The example they identified were traffic signs that have been vandalised and traffic signs that are partially obstructed. The random erasing technique will be able to simulate this scenario without the need to gather new data and do the labelling process of the data. Both papers [2] and [14] have identified weather playing a major factor into the ability of a model to detect traffic signs. The effects of weather such as rain on the TSDR system make it difficult to accurately detect signs and they lead to an increase in false sign recognition. A suggestion to overcome the weather-based problems were to enhance the quality of the image using pre-processing techniques to make the features in an image clearer for a model to identify.

3 Project Plan

3.1 Deliverables

The project is made up of the following deliverables that are expected to be completed by the end of December:

- CNN model trained on non-augmented dataset with a high accuracy of classifying traffic signs, above 90%.
- CNN model trained on augmented dataset with a moderately high accuracy, above 70%, of classifying obstructed traffic signs.
- Augmented dataset with all the images clearly labelled to show what traffic sign they represent. The dataset will be used in the training of one of the CNN models and in the testing of both of them.
- A report comparing the accuracy between testing obstructed traffic signs on the model not trained on the augmented dataset and the model specifically trained on the augmented dataset.

3.2 Gantt Chart

The Gantt Chart from Fig 7. outlines the predicted timeline and duration of the project, with the project starting in September and expecting to be finished by the end of December. Each block in the diagram represents one working week. The first 2 weeks are for catching up on the language the models will be written in, Python, and the research and set up of all the needed libraries and the dataset. After the initial set up is complete, the first CNN can be developed and trained using the dataset which brings the project to the first milestone. By M1 it is expected to have a working CNN that can accurately classify traffic signs. The next part of the project would be to set up and create the augmented data set, which is expected to take around 3 weeks and after the dataset is created, the first CNN can be tested with normal data and with augmented data. By milestone 2, M2, a second CNN is expected to be created and trained with the augmented dataset. At the end of the project is when testing for the second CNN happens along with further testing and a comparison and analysis between both of the models. It is expected that it would take around a week at the end of the project to prepare a presentation on the findings of the project ready for January. Lastly, the dissertation will start being worked on at the same time as the first CNN, this is to allow time to write down any results and difficulties faced during the project.

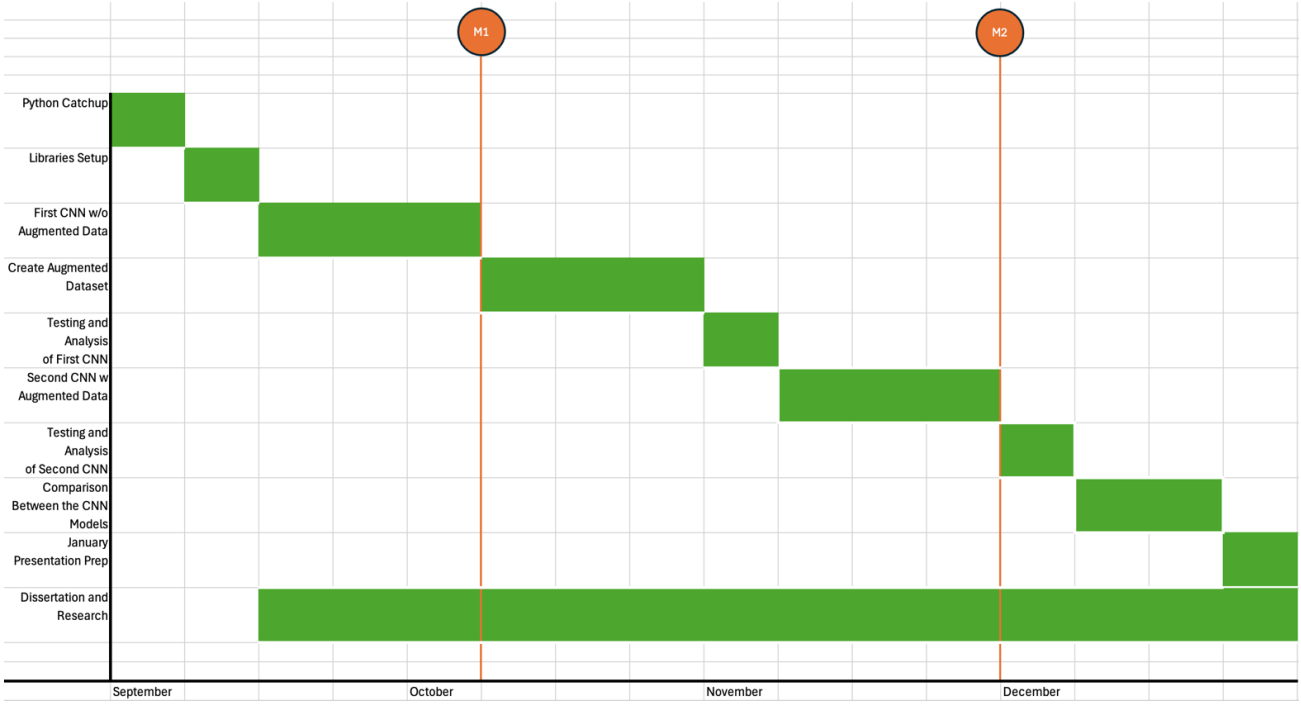

Fig 7. Ghantt Chart depicting the project time frame

3.3 Tools and Technologies

One of the needed tools to write the project is a library called PyTorch. PyTorch is an open-source deep learning framework used to build, train and evaluate neural networks. It will be used to write the CNN models to detect features and classify the images [15]. However, as a contingency plan in case anything goes wrong with training the model on a Mac laptop, such as the machine not being able to train CNN models because of the lack of available GPUs, there are other tools available online that could be used to train the models on, such as Google Colab. Google Colab allows for online GPU usage and is a good back up plan in case of any challenges of running the models locally. Also, it has no set up time and it uses Python so if there is a need to use it, it will be a quick switch between the local PyTorch implementation and Colab [16].

Since the language used to write the architecture for the CNN models is Python, to keep it cohesive for the project, Python will also be used to write the data augmentation system used to add obstructions to the images. Albumenations [17] is an image augmentation library that could be used to generate new sets of traffic images to use for model training. It allows for a good range of transformation techniques that will be used to create some real world scenarios such as fog or objects blocking traffic signs. Git and GitHub will be used for version control throughout the development of the project. Version control, whilst working with CNN models, has the benefit of allowing to experiment with different architectures and compare the performance of the models to find the most efficient and accurate one.

4 Bibliography

[1] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, 'Convolutional neural networks: an overview and application in radiology', *Insights Imaging*, vol. 9, no. 4, pp. 611–629, Jun. 2018, doi: 10.1007/s13244-018-0639-9.

[2] M. Q. Kheder and A. A. Mohammed, 'Improved traffic sign recognition system (itsrs) for autonomous vehicle based on deep convolutional neural network', *Multimed. Tools Appl.*, vol. 83, no. 22, pp. 61821–61841, Jul. 2024, doi: 10.1007/s11042-023-15898-6.

[3] C. Isidore, 'Elon Musk says his next product will be the most important Tesla has ever produced. And he says it's (finally) almost here | CNN Business', CNN. Accessed: Jun. 25, 2025. [Online]. Available: https://www.cnn.com/2025/06/12/business/musk-tesla-robotaxi-rollout-austin

[4] 'Uber brings forward trialling driverless taxis in UK', BBC News. Accessed: Jun. 25, 2025. [Online]. Available: https://www.bbc.com/news/articles/cwynm4wl1j7o

[5] Y. Wu and J. Feng, 'Development and Application of Artificial Neural Network', *Wirel. Pers. Commun.*, vol. 102, no. 2, pp. 1645–1656, Sep. 2018, doi: 10.1007/s11277-017-5224-x.

[6] 'What is a Neural Network? | IBM'. Accessed: Jun. 14, 2025. [Online]. Available: https://www.ibm.com/think/topics/neural-networks

[7] Z. Li, F. Liu, W. Yang, S. Peng, and J. Zhou, 'A Survey of Convolutional Neural Networks: Analysis, Applications, and Prospects', *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 12, pp. 6999–7019, Dec. 2022, doi: 10.1109/TNNLS.2021.3084827.

[8] A. Ajit, K. Acharya, and A. Samanta, 'A Review of Convolutional Neural Networks', in *2020 International Conference on Emerging Trends in Information Technology and Engineering (ic-ETITE)*, Feb. 2020, pp. 1–5. doi: 10.1109/ic-ETITE47903.2020.049.

[9] A. Zafar *et al.*, 'A Comparison of Pooling Methods for Convolutional Neural Networks', *Appl. Sci.*, vol. 12, no. 17, Art. no. 17, Jan. 2022, doi: 10.3390/app12178643.

[10] X. Ying, 'An Overview of Overfitting and its Solutions', *J. Phys. Conf. Ser.*, vol. 1168, no. 2, p. 022022, Feb. 2019, doi: 10.1088/1742-6596/1168/2/022022.

[11] K. Nanthini, D. Sivabalaselvamani, K. Chitra, P. Gokul, S. KavinKumar, and S. Kishore, 'A Survey on Data Augmentation Techniques', in *2023 7th International Conference on Computing Methodologies and Communication (ICCMC)*, Feb. 2023, pp. 913–920. doi: 10.1109/ICCMC56507.2023.10084010.

[12] 'What is Data Augmentation? - Data Augmentation Techniques Explained - AWS', Amazon Web Services, Inc. Accessed: Jun. 20, 2025. [Online]. Available: https://aws.amazon.com/what-is/data-augmentation/

[13] K. Maharana, S. Mondal, and B. Nemade, 'A review: Data pre-processing and data augmentation techniques', *Glob. Transit. Proc.*, vol. 3, no. 1, pp. 91–99, Jun. 2022, doi: 10.1016/j.gltp.2022.04.020.

[14] S. B. Wali *et al.*, 'Vision-Based Traffic Sign Detection and Recognition Systems: Current Trends and Challenges', *Sensors*, vol. 19, no. 9, Art. no. 9, Jan. 2019, doi: 10.3390/s19092093.

[15] 'What is PyTorch?', NVIDIA Data Science Glossary. Accessed: Jun. 23, 2025. [Online]. Available: https://www.nvidia.com/en-gb/glossary/pytorch/

[16] ‘colab.google’, colab.google. Accessed: Jun. 23, 2025. [Online]. Available: http://0.0.0.0:8080/
[17] ‘Albumentations: fast and flexible image augmentations’, Albumentations. Accessed: Jun. 23, 2025. [Online]. Available: https://albumentations.ai/