

# Compare and Contrast LiDAR and Non-LiDAR Technology in an Autonomous Vehicle: Developing a Safety Framework

Benjamin Quito, Larbi Esmahi

School of Computing and Information Systems, Athabasca University, Alberta, Canada

Email: [bquito1@learn.athabascau.ca](mailto:bquito1@learn.athabascau.ca), [larbie@athabascau.ca](mailto:larbie@athabascau.ca)

**How to cite this paper:** Quito, B. and Esmahi, L. (2023) Compare and Contrast LiDAR and Non-LiDAR Technology in an Autonomous Vehicle: Developing a Safety Framework. *Open Journal of Safety Science and Technology*, 13, 101-131.

<https://doi.org/10.4236/ojsst.2023.133006>

**Received:** June 16, 2023

**Accepted:** September 1, 2023

**Published:** September 4, 2023

Copyright © 2023 by author(s) and

Scientific Research Publishing Inc.

This work is licensed under the Creative

Commons Attribution International

License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

## Abstract

Ensuring safety has always been of utmost importance in vehicular operations, where traditionally, human drivers have been solely responsible for driving. However, with the emergence of advanced technologies, we are now on the brink of a new era with the introduction of Autonomous Vehicles (AVs), in which control over the vehicle gradually shifts to Artificial Intelligence (AI). The safety of these vehicles has raised concerns among the public. In terms of safety, human drivers heavily rely on visual perception. Key elements that contribute to safe driving include situational awareness, vehicle control, reaction capabilities, and anticipation of potential hazards. With the introduction of AVs, these fundamental factors remain unchanged. AVs rely on two primary technologies, namely LiDAR and Non-LiDAR, to perceive their surroundings. This research focuses on three primary aspects. Firstly, it involves the development of an image classification model to assess the safety of AVs. This model determines whether the images captured by LiDAR and Non-LiDAR technologies can be accurately predicted using supervised learning. An algorithm is employed to identify the input images obtained from both LiDAR and Non-LiDAR technologies. The results demonstrate that the model achieved a high accuracy rate of 94.63% in predicting the images. Secondly, a Safety Framework is established to facilitate the subsequent proposal for Experimental Research, which is the third aspect. The Safety Framework incorporates the application of LiDAR and Non-LiDAR technologies mounted on a vehicle that is operated in diverse weather conditions by both a human driver and an autonomous system. The Scenario tables, which are currently in a blank state, will be populated upon completion of the Experimental Research. The Experimental Research aims to compare and contrast the performance of LiDAR and Non-LiDAR technologies on a vehicle driven by a human operator versus an Autonomous Vehicle. The findings of this re-

search will be documented in the Scenario table outlined within this study, ultimately shedding light on the safety implications of implementing LiDAR and Non-LiDAR technologies within an AV context.

## Keywords

AI, ML, Image Classification, Safety Frameworks, LiDAR, Non-LiDAR

---

## 1. Introduction

The idea of an Autonomous Vehicle (AV) started with Leonardo Da Vinci when he designed a self-propelled cart that could move without being pushed [1]. Google started its Autonomous Vehicle project in 2009 [2]. In 2015, Tesla released its autopilot [3]. Tesla's autopilot is not entirely AV; the feature is Autosteer that keeps the car in its current lane; the technology has four different kinds of sensors; forward radar, forward-facing camera, 360-degree ultrasonic sensors, and GPS in combination with high-resolution navigation maps designed to track individual lanes and features of the roads [4]. NHTSA further mentioned that self-driving vehicles would be integrated into U.S. roads by advancing through the six levels of driver assistance technology advancements. The six-level driver assistance technology advancements focus on who does what and when, *i.e.*, the driver and the Automated Driving systems (ADS).

### 1.1. Six Levels of Driver Assistance Technology Advancements

According to NHTSA, these are the levels of automation, who does what and when shown in **Table 1** [5].

Level 0 has no automation, and the human driver completely controls the vehicle. Level 1 has driver assistance; the vehicle is equipped with assistance functions such as lane-keeping assistance, and the human driver remains responsible for all driving tasks. Level 2 is partial automation; humans still drive the vehicle, and the vehicle combines automated functions such as lane-keeping and adaptive cruise control. The human driver must be prepared to take control of the vehicle at any time. Level 3 is conditional automation; an automated driving system drives the vehicle and can perform all aspects of driving under certain conditions; the human driver can intervene when prompted by the system or when specific conditions arise. Level 4 is high automation; an automated driving system drives the vehicle, and the system can operate autonomously without human intervention on specific operational conditions; the human driver is disengaged from driving responsibilities. Level 5 is full automation; an automated system drives the vehicle and is capable of all driving tasks under all conditions, and no human intervention is required.

Back in September 2016, the NHTSA and the United States Department of Transportation (USDOT) has released a Federal Automated Vehicles Policy (FAVP) to provide a safety approach on safety assurance and innovations [6]. In

**Table 1.** The six levels of automation of NHTSA.

Levels	Description
Level 0	The driver does all the driving without an advanced driver assistance system (ADAS) intervention.
Level 1	There is an ADAS onboard the vehicle which not most of the time assists the driver in steering, braking, or accelerating and not concurrently.
Level 2	There is an onboard ADAS that controls the steering, braking, or accelerating in each situation and the driver must always pay full attention, then do the rest of the task of driving.
Level 3	There is an ADS onboard and do all the driving task in each situation, the driver must be ready to take back the control at any time when the ADS request the driver to do so.
Level 4	The ADS performs all the tasks of driving, and monitors the environment in each situation, the driver does not need to pay attention at all.

September 2017, NHTSA and USDOT released the “Automated Driving Systems: A Vision for Safety 2.0” document [7], which gives a nonregulatory method on automated vehicle technology safety and safe testing of Levels 3-5. In October 2018, both NHTSA and USDOT released the document “Preparing for the Future of Transportation: Automated Vehicles 3.0” [8]; the paper was organized in three key areas such as the advancement of multi-modal safety, the reduction of policy uncertainty, and the outline of the process for working with USDOT. And on January 2020, the USDOT released a document called “Ensuring American Leadership in Automated Vehicle Technologies: Automated Vehicles 4.0” [9], which was based on “Preparing for the Future of Transportation: Automated Vehicles 3.0” and expanded 38 components and organized on three key areas. The United States Government Automated Vehicles (USG AV) principles and the administration’s efforts in supporting the Automated Vehicle (AV) growth and leadership. Including the United States Government (USG) activities and opportunities for collaboration.

## 1.2. Framework Measuring Automated Vehicle Safety

In 2018 RAND Corporation, through the research of Blonar *et al.* [10], wrote a report on automated vehicle safety; it was found that there was no definite standard that existed in terms of AV safety, but the information presented a framework wherein the safety of AV can be tested and measured. The framework developed by Blonar *et al.* [10] focused on the features of Level 4 of SAE J3016 [11] version levels of driving automation. The testing was performed within pre-specified conditions. Level 4, as explained on the Landmark Dividend website [12], is where the automated system can take over the driving task in each situation. NHTSA [5] mentioned that drivers do not need to pay attention.

In the research of Blanar *et al.* [10], safety was defined as the elimination, minimization, or management of harm to the public; this research focused on comparing AV and conventional vehicles. To further clarify, the comparison is based on the operational design domain (ODD) Blanard& Holliday [13], and ODD was described by Blanar *et al.* [10] in terms of geography, weather, lighting, roadway markings, previous experience on a specific roadway. **Table 2** shows the integrated safety framework developed by Blanar *et al.* [10], wherein Column 1 is the Setting, Column 2 is the Stage, and Column 3 is the Leading Measures, and Column 4 Lagging Measures.

The research can interest stakeholders such as academics, transportation regulators, transportation managers, industries, and the public. The future of logistics will be based on autonomous or semi-autonomous vehicles; this research will help develop a guide on such vehicles' rules and regulations on the public road. Businesses inclined to use Autonomous vehicles will have an informed choice of whether to go with the technology of LIDAR or Non-LIDAR, which are currently being developed.

The two-leading image-gathering technologies for AVs' image gathering: LiDAR and Non-LiDAR, are essential for the safety of AVs. It serves as the eyes to detect objects on the roads and surroundings; it is paramount that these technologies must assure the public that such detection of objects can be used to make the driving of the AVs safe. For society to trust their lives on AVs, there must be a proper safety measurement. The public needs to know the similarities and differences between the two technologies regarding safety. The businesses who invest and use the technology should ensure it is safe to use in any weather conditions. Therefore, there is a need to develop a safety framework to measure the two technologies' safety. In the research conducted by Blanar *et al.* [10], it was not mentioned that the AV used in the experiment uses LiDAR or Non-LiDAR. Using LiDAR on Non-LiDAR in terms of safety is essential because this will define the safety of AVs.

## 2. Background

On November 11, 2020, in Japan, Honda Motor Co., Ltd. announced the approval of an SAE Level 3 autonomous vehicle [14]. According to the SAE International standard, a level 3 autonomous vehicle means the driver is not driving, and the automated features of autonomous driving are engaged, even though the driver is in the driver's seat. According to the article Honda, the Ministry of Land, Infrastructure and Tourism (MILT) of Japan approved the Level 3 autonomous vehicle based on SAE standards; this will enable the vehicle to drive when there is traffic congestion on the expressway; Honda calls this as "Traffic Jam Pilot" and will be available to consumers on the first quarter of 2021. In March 2021, Honda made the Honda Legend available to the public, the first SAE Level 3 certified AV and considered the world's first [15]. In October 2021, an article on Car Buzz [16] stated that Honda is already testing Level 4

**Table 2.** The integrated safety framework.

Stage	Setting	Leading Measures	Lagging Measures
Development	Simulation	Infractions	Outcome
Demonstration	Closed course	Roadmanship	
Deployment	Public roads	Disengagement	

autonomous technology and compares it to Tesla being in Level 2 in autonomous technology.

### 2.1. LiDAR

LiDAR, according to Wasser [17], is an active remote sensing system, and an active system is a system that produces energy or light to quantify things on the ground. Wasser further explained that remote sensing is measuring things, not using our hands; sensors are used to obtain information regarding a landscape and record things to measure characteristics and conditions. Light is emitted from a quick-firing laser, which makes a LiDAR system; according to Wasser, this light goes to the ground, and the light bounces off obstacles like buildings or trees; the bounced light then goes back to the LiDAR sensor then records it.

LiDAR sensors can achieve mapping precision of up to 1 cm horizontally (x, y), and 2 cm vertically (z) and range accuracy of 0.5 to 10 mm relative to the sensor. As a result, they serve as particularly advantageous remote sensing equipment for mobile mapping. (Vectornav n.d.). Furthermore, LiDAR sensors may capture several returns from a single light pulse. This is due to the fact that as the light pulses travel from the sensor, they may come into contact with many objects that will reflect the pulse, such as the leaves and branches of a tree canopy. LiDAR sensors can record this data to provide a detailed picture of both the tree canopy and the underlying structure.

### 2.2. Non-LiDAR

This technology does not employ LiDAR on AV; it uses the same technology as other Avs, except it does not install LiDAR as part of autonomous driving but relies heavily on the onboard camera. On October 22, 2020, Washington Post published an online article about Tesla putting self-driving technology in their cars [18]. Before the Washington Post, on October 20, 2020, Elon Musk, the CEO of Tesla, tweeted that an FSD (Full Self-driving) rollout was happening [19]. The article in Washington Post mentions that according to some safety experts, Tesla's technology can detect vehicles and pedestrians on the road, as well as some objects such as trees, but it cannot always discern the real shape or depth of the obstacles it encounters. As it approached the rig from behind, the automobile might not be able to tell the difference between a box truck and a semi. Furthermore, according to the article, Tesla will use eight surrounding-view cameras connected to the car to gather information so it can steer through freeways, city streets, and traffic. This is made possible by enhancing the

software of the car to compensate for its hardware. This will create a virtual LiDAR using eight cameras connected to its neural network.

### 2.3. Current Research

Zywanowski, Banaszczyk, & Nowicki [20] compared camera-based (non-LiDAR) and 3D LiDAR-based image recognition across different weather conditions and processed all data inputs employing similar architecture of a neural network and worked for 50 weeks and found that there is a need for more research into place recognition with multi-sensory setups. Their study used the data set from the University of Sydney that has been recorded using cameras, 3D LiDAR u-blox GPS, an Inertial Measurement Unit (IMU), and others. The researchers drove the same route weekly for over a year and gathered 50 recordings of lighting conditions, weather conditions, infrastructural, environmental, and traffic conditions. In all these driving, they used cameras (non-LiDAR) and LiDAR and then compared the data collected. Zywanowski, Banaszczyk, & Nowicki [20] used transfer learning to train their networks; the camera and LiDAR intensity images were used in these training datasets. The researchers acquired the results based on weather conditions observed and divided them into six categories such as (S) sunny, (C) cloudy, (S/C) sunny/cloudy, (AR) after rain, (SS) sunset, and (VC) very cloudy then created a table comparing camera (non-LiDAR) and LiDAR.

A comparison review on LiDAR and Camera (Non-LiDAR) in AV conducted by Mugunthan *et al.* [21] states that in using LiDAR, the pulses are affected by heavy rains or hanging clouds and that these obstacles influence the refraction. Furthermore, the sun's angle also has a significant impact since laser pulses are based on the principle of refraction. Regarding the use of a camera, Mugunthan *et al.* [21] indicated that when using a camera, there are problems with changing lighting and weather conditions and that the depth data using algorithms are not as accurate as the LiDAR. From the research, the researchers concluded that LiDAR and cameras on their own are not safe and that it was suggested in the article that LiDAR and cameras should be used together. However, the research did not mention using image processing to compare LiDAR and camera; there was no baseline to test the findings. The study is all based on a literature review.

### 2.4. Image Recognition Model

The website of Fritz.ai [22] defines image recognition as a computer vision technique that allows machines to decode and classify what they see on an image or video. This is often called image classification or image labelling; in machine learning, Du, Guo, & Simpson [23] wrote an article on self-driving car steering angle prediction based on image recognition. In their research, they used a set of images with the steering angle captured during driving; the study explored two models to perform predictions based on photos using various deep learning techniques, which include Transfer Learning, 3D CNN (3D Convolutional Neural Network), LSTM (Long Short-Term Memory) and ResNet (Residual

Network). Furthermore, the website of Fritz.ai [22] explains that to build an image recognition model which will automatically detect an image, there are three basic steps one is to train the model, two to input the image or video, three is the output which will statistically predict the input image.

### 2.5. Image Recognition Architectures

Several neural network architectures are used in image recognition, and it is typical for the neural network to be used in different image recognition problems; it can also be tested on object detection or segmentation. Kumar [24] explained that CNN architectures are a well-known deep learning framework and the application of which ranges from computer vision to natural language processing (NLP); based on an article written by Dang [25], many technology companies developed research studying architectures in CNN such companies are Google, Microsoft, and Facebook these companies used CNN for analyzing image content, image segmentation, classification, detection, and retrieval. Further, the article explains that CNN builds a network in neurons in the early layer of the network, extracts visual topographies, then neurons in later layers put together all the topographies and create higher-order topographies. The layers stated in the article are convolutional, pooling, and fully connected.

## 3. Methodology

Since Karl Benz's invention of the automobile in Germany in 1885/1886 [26], human drivers have consistently assumed control of vehicles. Among the paramount safety requirements in driving is the ability to visually perceive the road, discern obstacles, and maintain situational awareness, all of which necessitate the use of our visual faculties. This fundamental principle also applies to autonomous vehicles (AVs) as they, too, require a means of "seeing." The integration of LiDAR and Non-LiDAR technology empowers AVs to perceive the road, identify obstacles, and maintain environmental awareness. Pertinent terminologies in the realm of AVs encompass perception, localization, prediction, and decision-making [27]. The focus of this research lies specifically in the realm of perception, which enables AVs to accurately identify and categorize the objects within their visual field. In terms of data acquisition for identification and categorization purposes, the two dominant technologies employed are the Camera, which falls under the Non-LiDAR category, and the LiDAR itself. Through an extensive literature review, this research endeavour seeks to juxtapose and examine the relative merits and limitations of these two technologies. Furthermore, the research entails a comparative evaluation of proposed experimental methodologies for data collection, which will serve as inputs for an image processing model. This model will be constructed based on publicly accessible datasets, facilitating a comprehensive analysis and comparison of the performance of the Camera and LiDAR technologies within the context of AV perception. Additionally, the research involves the development of a safety framework that will effectively contrast and evaluate the outcomes of the conducted experi-



ments. Ultimately, this research endeavour will contribute by proposing innovative experimental methodologies to enhance our understanding of AV perception capabilities and further refine the field's safety framework. **Figure 1** shows the concept of the research framework.

### 3.1. LiDAR for Object Detection and Image Classification

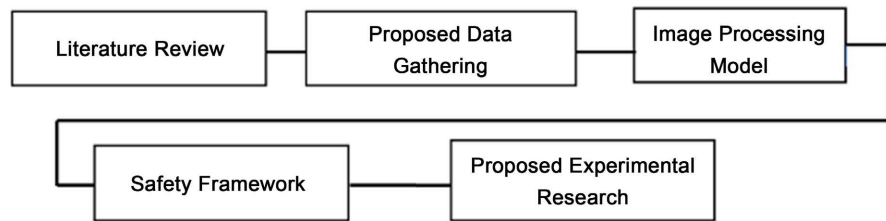
Acquiring LiDAR data presents significant challenges compared to visible image data, as highlighted by Wenhui and Fan [28]. These challenges include the sparseness of the LiDAR point cloud and the occurrence of mutual occlusion, where objects obstruct each other within close proximity, which poses a crucial obstacle in object detection and classification algorithms [29]. The recognition and classification of object structures based on point cloud data involve manipulating the distinct properties of objects, such as their non-uniform densities and non-structural distributions, which deviate from conventional methods of point cloud analysis. Furthermore, the accuracy and speed of LiDAR-based systems are hindered by the unorganized allocation of the LiDAR point cloud [30].

Zhang, Fu, & Dai [29] addressed the issue of mutual occlusion in their research on LiDAR-based object classification by employing Explicit Occlusion Modeling. Their investigation emphasized the crucial role of mutual occlusion between adjacent objects in object detection and classification, as it significantly affects accuracy. The authors' research approach involved explicitly modeling occlusion, defining a view volume in which the LiDAR camera is most likely to be positioned during runtime. However, it should be noted that in the research conducted by Zhang, Fu, & Dai [29], the LiDAR point cloud used for classification was incomplete and fragmented, leading to potential misclassifications.

In their research, Song *et al.* [31] highlighted the challenges arising from the unstructured distribution, disordered arrangement, and large volume of data in LiDAR point clouds, which result in high computational complexity and difficulties in classifying 3D objects. To address these issues, the authors proposed a CNN-based 3D object classification method that leverages the Hough space of LiDAR point clouds. The Hough Transform, a method commonly employed for isolating features of specific shapes in images, including lines, circles, and ellipses [32], was utilized to transform the object point cloud into Hough space. Subsequently, the Hough space was rasterized, transforming the electronic data into a sequence of evenly sized grids [33]. The count of accumulators in each grid was computed and fed into a CNN model for the classification of 3D objects. Moreover, the researchers developed a semi-automatic 3D object labeling tool to construct a LiDAR point cloud object labeling library. Following the initialization of the CNN model, the dataset from the object labeling library was employed to train the neural network. The outcome of this approach yielded an object classification accuracy of 93.3% [31].

The process of gathering data, as illustrated by the research conducted by Song *et al.* [31], involves transforming the object point cloud into Hough space using the Hough Transform algorithm. Subsequently, the Hough space is





**Figure 1.** Research concept framework.

rasterized into evenly sized grids, and the count of accumulators in each grid is computed. These accumulated counts are then utilized as input for a CNN model to classify 3D objects. Additionally, a semi-automatic 3D object labelling tool is employed to construct a LiDAR point cloud object labelling library. The CNN model is initialized, and the dataset from the object labelling library is used for training the neural network. The proposed methodology demonstrates promising results in achieving accurate object classification in LiDAR-based systems.

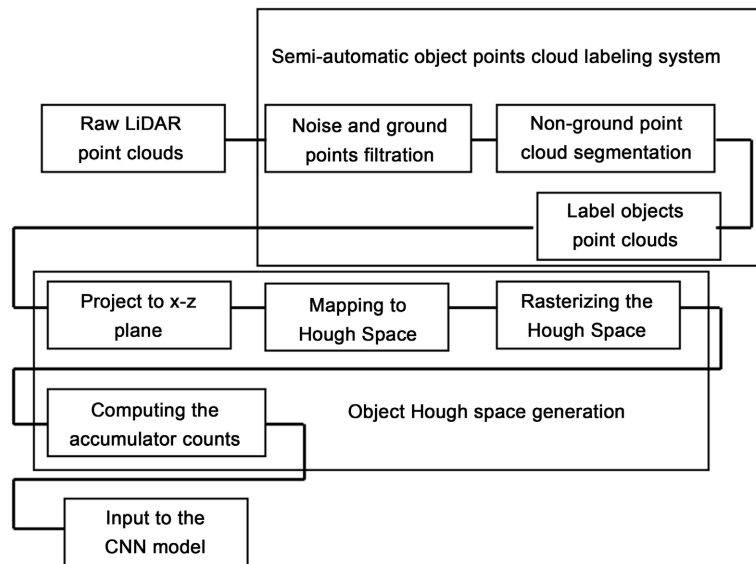
The process of gathering data based on the research of Song *et al.* [31]. **Figure 2** shows the process of gathering data of LiDAR in preparation for the CNN input [31].

### 3.2. Non-LiDAR for Object Detection and Image Classification

In May of 2021, Tesla, a prominent advocate of Non-LiDAR technology, made a significant transition to a vision-only model by removing all Radar sensors from their vehicles. This decision was aimed at demonstrating the company's belief that cameras alone are sufficient for computer vision applications. According to information from the Tesla website, the Model 3 and Model Y, manufactured in North America, became the first vehicles to rely on camera vision and neural net processing for their Autopilot self-driving capabilities [34].

Jahromi [35] provided an explanation of the process by which light emitted from an object passes through a lens and lands on the light-sensitive surface or image plane. The light-sensitive surface converts the rays into electrons, which are then transformed into voltage, amplified, and passed through an Analog-Digital Converter (ADC) to ultimately form a pixel.

In line with the research conducted by Fujiyoshi, Hirakawa, & Yamashita [36], the extraction of feature vectors or local features from 2D images is crucial for image recognition. In the context of autonomous vehicles, the detection of objects such as pedestrians is of particular interest. Fujiyoshi, Hirakawa, & Yamashita [36] suggested the use of histogram-oriented gradients (HOG) features in combination with a support vector machine (SVM) for this purpose. HOG features involve calculating the gradient orientation quantities in a portion of an image. The extracted image is then resized to dimensions of  $128 \times 64$  pixels, and the gradient is calculated by considering both magnitude and angle. The gradient matrices are divided into  $8 \times 8$  cells to form blocks, and from each block, a 9-point histogram is computed. These histograms with 64 different values represent the intensity of the gradient in each bin. After performing the



**Figure 2.** Process of data gathering of LiDAR in preparation for CNN input.

histogram calculations for all blocks, four blocks from the 9-point histogram matrix are combined to create a new block. This combination is achieved through an overlapping process with an 8-pixel stride. For each block, the 9-point histogram values from all four cells are concatenated to form a 36-feature vector, thus extracting a HOG feature [37].

An SVM is a machine learning algorithm utilized for searching a hyperplane in an N-dimensional space to classify data points. Hyperplanes act as decision boundaries for classifying data points, which can belong to different classes and can exist on either side of the hyperplane. The dimensions of the hyperplane depend on the number of features considered. In the case of two features, the hyperplane corresponds to a line, while for three features, it represents a two-dimensional plane [38]. Support vectors, which are data points close to the hyperplane, influence its position and orientation. By incorporating these support vectors, the classifier maximizes the margin, which refers to the distance between the hyperplane and the nearest data points. The removal of support vectors would alter the location of the hyperplane, making them integral to SVM [38].

### 3.3. Image Processing Model

The Literature Review establishes that CNN (Convolutional Neural Network) is considered the most effective approach for object detection and image classification. This is attributed to the limitations of traditional methods, such as their complexity in processing a large volume of data, insufficient accuracy, and inadequate processing speed. CNN has demonstrated significant advancements in various areas, including image classification, object detection, and image segmentation. Leonard [39] conducted comprehensive research on CNN and developed algorithms for image classification and object detection based on this

neural network architecture. The fundamental structure of CNN consists of several key components, including an input layer that receives input from both LiDAR and Non-LiDAR technologies, a convolutional layer, a pooling layer, a fully connected layer, and an output layer [39] (Figure 3).

Once the substitution of multiple convolutional layers and downsampling layers is performed, the Convolutional Neural Network (CNN) proceeds to utilize a fully connected network for the purpose of classifying the extracted features and obtaining a probability distribution based on the input. Figure 4 shows the basic structure of CNN with downsampling [39].

### 3.4. CNN Implementation

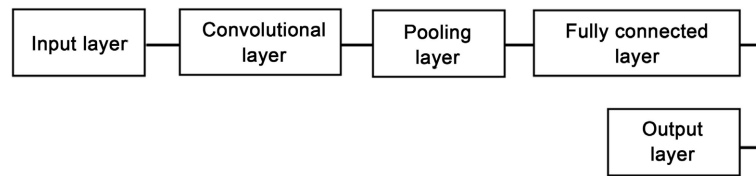
The implementation of Convolutional Neural Networks (CNNs) plays a crucial role in various fields, including computer vision and image recognition. CNNs are deep learning models specifically designed to process and analyze visual data, making them highly effective in tasks such as image classification, object detection, and facial recognition.

CNNs consist of multiple layers, including convolutional, pooling, zero padding, Dropout Layer, and fully connected layers. The convolutional layers perform the main computations by applying a set of learnable filters to input images, enabling the network to automatically extract hierarchical features. The pooling layers reduce the spatial dimensions of the features, helping to extract important information while preserving spatial invariance. Dropout is a regularization technique commonly applied to fully connected layers or convolutional layers in CNNs. It aims to prevent overfitting by randomly dropping out a proportion of the neurons during training. Zero padding is a technique used in convolutional layers to preserve the spatial dimensions of the input while applying convolutions. It involves adding zeros around the borders of the input feature maps before performing convolutions. Finally, the fully connected layers act as a classifier, making predictions based on the extracted features.

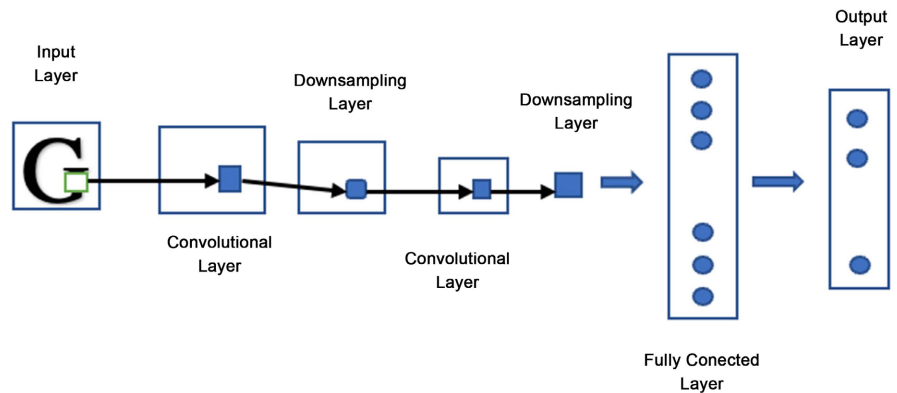
To implement CNNs, popular deep learning frameworks such as Tensor Flow [40] and PyTorch [41] provide extensive libraries and tools that simplify the process. These frameworks offer pre-defined CNN architectures, such as VGGNet [42], ResNet [43], and InceptionNet [44], which have achieved remarkable performance in various computer vision tasks.

For example, researchers have successfully employed CNNs in image recognition tasks. In a study by Krizhevsky, Sutskever, and Hinton [45], they introduced a CNN architecture known as AlexNet, which significantly improved the accuracy of image classification on the ImageNet dataset. The AlexNet model comprised multiple convolutional and pooling layers, followed by fully connected layers, and achieved state-of-the-art results at the time of its publication.

In summary, the implementation of CNNs is essential for various computer vision tasks, and popular deep learning frameworks such as TensorFlow and PyTorch provide the necessary tools and libraries to facilitate the development and training of CNN models. Numerous studies have demonstrated the



**Figure 3.** Basic structure of CNN.



**Figure 4.** Basic structure of CNN with downsampling.

effectiveness of CNNs, such as the pioneering work of Krizhevsky *et al.* [45] with the AlexNet architecture, which revolutionized the field of image recognition.

### 3.5. Safety Framework

In order to assess the safety of LiDAR and Non-LiDAR technologies, it is necessary to establish parameters that allow for a fair comparison between the two. Therefore, this research employs specific parameters that evaluate the capability of autonomous vehicles (AVs) to perceive the road, detect obstacles, and maintain situational awareness under various weather conditions. The image recognition model utilized in this study aims to classify and identify the images it is presented with. To accomplish this, publicly available training and testing image datasets are employed.

In this research, the VGG-16 architecture, the same as that used by Zywanowski, K., Banaszczyk, A., & Nowicki, M. [20], is adopted to train the image recognition model. The experimental details of the safety framework are comprehensively presented in Chapter V, with particular emphasis on the image processing model. Both LiDAR and Non-LiDAR sources are utilized as inputs for the image recognition model in the convolutional neural network (CNN). The methodology for collecting the data is publicly available images, including the VGG16 dataset, are used. The selection of the VGG Neural Network Architecture is based on extensive literature research that supports its compatibility and effectiveness with CNNs, as demonstrated by Zywanowski, K., Banaszczyk, A., & Nowicki, M. [20].

#### 3.5.1. VGG Architecture

The architecture referred to as VGGNet, or VGG16, is widely recognized as a

classical convolutional neural network (CNN) architecture. Its primary purpose is to enhance the depth of CNN models, thereby improving their overall performance [46]. VGG16 consists of 16 layers designed to classify images across 1000 object categories, and it operates with an input size of  $224 \times 224$  pixels. This architecture is constructed based on the most effective and essential features of CNNs [46]. **Figure 5** shows the VGG net architecture [46].

### 3.5.2. VGG Architecture

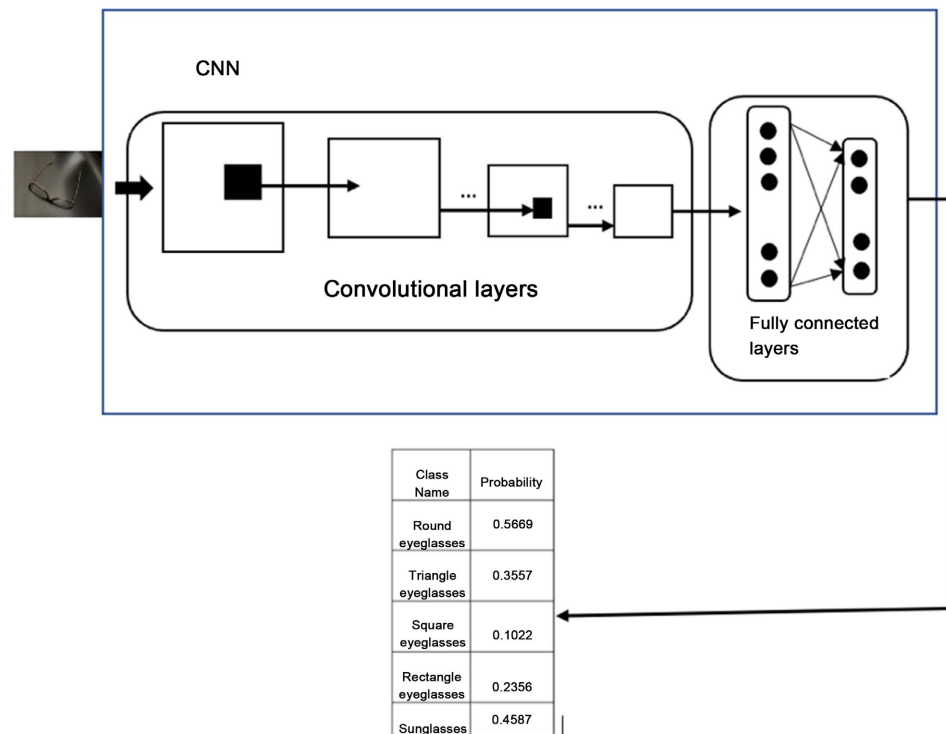
The VGG16 architecture consists of a total of 13 convolutional layers and three fully connected layers [46]. The key characteristics of VGG16 are as follows:

- **Input:** The network takes in an image with dimensions of  $224 \times 224$  pixels.
- **Convolutional Layers:** VGG16 employs  $3 \times 3$  convolution filters, which are the smallest possible size for convolutions. The input undergoes a linear transformation using  $1 \times 1$  convolution filters, followed by a Rectified Linear Unit (ReLU) activation function. The convolution stride is set to 1 pixel, ensuring that the spatial resolution remains unchanged after convolution.
- **Hidden Layers:** In VGG16, all the hidden layers utilize the ReLU activation function.
- **Fully Connected Layers:** VGG16 includes three fully connected layers. The first two layers have 4096 individual channels, while the third layer has 1000 channels, each corresponding to a specific class.

The VGG16 network is readily available in Keras, an image processing framework capable of recognizing 1000 categories of images. Instead of training VGG16 from scratch on the ImageNet dataset, this research leverages pre-trained weights from ImageNet. The image is converted into a NumPy array before being passed through the model, which processes batches of images, treating them differently across various channels. In this research, the safety measurement entails considering the top five outputs with the highest probability from the model. For instance, if the input is an image of a car, the image processing model should predict that it belongs to the “car” category among the top five model outputs.

### 3.6. Image Processing Model

The images used in this research are sourced from two technologies: LiDAR and Non-LiDAR. LiDAR utilizes point cloud data to recognize and classify images, employing the PointNet method to feed the data into the CNN [29]. On the other hand, Non-LiDAR uses camera images directly as input to the CNN. The objective of comparing and contrasting these two technologies is to identify the differences and similarities in their performance under different weather conditions. The hypothesis to be investigated by the image processing model is that there is no difference in the speed and accuracy of image recognition between the two technologies in various weather conditions for Autonomous Vehicles. The research aims to develop an image processing algorithm to be used in the proposed experimental research, which will test and validate this hypothesis.



**Figure 5.** VGG net architecture.

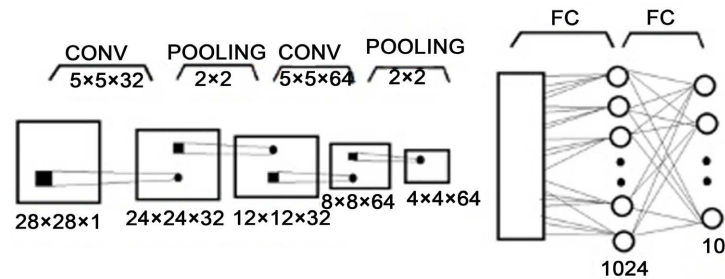
In the proposed experiment, both LiDAR and Non-LiDAR technologies will be tested by human drivers in controlled environments under different weather conditions. The data collected in this experiment will serve as the baseline for the subsequent experiment, where the vehicles will be driven autonomously without human intervention. The results of both experiments will be compared and analyzed to address the hypothesis.

For building the image processing model, the researcher employed supervised learning, a branch of machine learning and artificial intelligence. In supervised learning, the computer learns to perform functions based on labeled training data [47]. The image processing model utilizes CNN to create and compile the images. The images obtained from LiDAR and Non-LiDAR technologies undergo processing to generate an output. During the image processing stage, local receptive fields, which are small regions of input layer neurons, connect to neurons in the hidden layer. These receptive fields traverse the image, creating an input map from the input layer to the hidden layer neurons. The process of creating and compiling the image processing model is illustrated in **Figure 6**.

The programming language used for implementation is Python 3.10.4, in conjunction with TensorFlow. TensorFlow facilitates the building and deployment of a supervised machine learning model, where the model is trained by providing input data and corresponding expected results.

The following processes are employed in building the Supervised Learning Model:

- 1) ML Algorithm: Convolutional Neural Network (CNN)



**Figure 6.** Creating and Compiling the image processing model fully connected network.

2) Training: Loading the training data along with their expected output to train the model. The training data accounts for 80% of the dataset, following the Pareto principle of 80% effects and 20% causation, as used by Zhao *et al.* [48].

3) Testing: Loading separate testing data, distinct from the training data, to predict the correct results. The testing data constitutes 20% of the dataset.

4) Evaluation: Loading new data for evaluation purposes. This data differs from both the training and testing data. The evaluation process ensures that the model can predict outputs for data it has not encountered before, thereby enhancing its accuracy.

### 3.6.1. CNN Model

A fully connected neural network is not suitable for processing images due to the large number of parameters that arise when each pixel is considered as an input. To address this issue, smaller images are used in this research. The creation and compilation of the image processing model are illustrated in the figure below. The input image size is  $28 \times 28$  with one channel. The convolution operations employ a  $5 \times 5$  kernel with 32 filters. Following the pooling operation, the image size is reduced to  $12 \times 12$ . Subsequently, another convolution is applied with a  $5 \times 5$  kernel and 64 filters, followed by another pooling operation, resulting in a  $4 \times 4$  kernel size. The flattened image is then connected to a Fully Connected (FC) network. The FC network consists of 1024 nodes, and the final output layer comprises ten outputs [49]. **Figure 6** shows the creation and compilation of the image processing model of a fully connected network [49].

### 3.6.2. Preprocessing

The initial step in training the ML algorithm is to load the data, and TensorFlow offers a convenient method for loading datasets. TensorFlow provides a pipeline that efficiently manages the loading of data, making it well-suited for handling large volumes of datasets [49]. The algorithm begins by loading the necessary libraries, including operations such as convolution, max pooling, flattening, and the fully connected (dense) layer, which are essential components of the CNN. For this research, the Modified National Institute of Standards and Technology Database (MNIST) datasets are utilized, as they are readily available in the Keras library [49].

A sequential model is employed in the algorithm, as it allows for a linear ar-



rangement of neural network layers. The algorithm utilizes two “categoricals” to reshape the data and output the labeled data into ten categories or bins. The training dataset consists of 60,000 images, while the validation dataset contains 10,000 images for assessing the performance of the model.

Data loading results:

```
(60,000, 28, 28)
(10,000, 28, 28)
(60,000,)
(10,000,)
```

Pre-processing the data results:

```
(60,000, 28, 28, 1)
(10,000, 28, 28, 1)
(60,000, 10)
(10,000, 10)
```

### 3.6.3. Creating and Compiling

The sequential model was constructed by following a series of steps. First, the convolution model was added to the model. Next, padding was applied using the “same” mode, ensuring that the input image was fully covered by the filter. This was done to ensure that all input sequence data had the same length, which required padding. The activation function employed in the algorithm was “relu”. A convolution layer was added, followed by a Maxpooling layer. Another convolution layer and a subsequent Maxpooling layer were added. The network was then flattened to prepare it for the fully connected network, also known as the dense layer.

The fully connected network consisted of 1024 nodes with the “relu” activation function. Another fully connected layer was added to serve as the output layer, which had ten bins or classes. The “relu” activation function outputs zero for any value of x that is less than zero, and for values of x that are equal to or greater than zero, it returns the input value x. Thus, ReLU produces an output of zero for all negative inputs and preserves positive inputs [50].

During the compilation of the model, the “adam” optimizer was employed as a parameter. For the loss function, categorical cross-entropy was used since the output had ten possible classes. The accuracy metric was chosen to evaluate the model’s performance [50].

Creating and compiling results:

Model: “sequential”

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 28, 28, 32)	832
max_pooling2d (MaxPooling2D)	(None, 14, 14, 32)	0
conv2d_2 (Conv2D)	(None, 14, 14, 64)	51264

max_pooling2d_1 (MaxPooling	(None, 7, 7, 64)	02D)
flatten (Flatten)	(None, 3136)	0
dense (Dense)	(None, 1024)	3212288
dense_1 (Dense)	(None, 10)	10250

=====  
Total params: 3,274,634

Trainable params: 3,274,634

Non-trainable params: 0

### 3.6.4. Training and Evaluation

After the creation and compilation of the model, the next crucial stages are training and evaluation. The training stage involves feeding the dataset into the image processing model. During training, the model analyzes the data, draws conclusions, and predicts the results. This process generates the output of the image processing model, which includes the input data and the corresponding output.

To assess the accuracy of the model, an evaluation is conducted. The evaluation phase tests the suitability of the given dataset and algorithm for the image processing model. Through evaluation, the accuracy of the input data is determined, and the model predicts the outcomes. The accuracy of the model plays a significant role in the image processing model; a higher accuracy indicates a more accurate prediction to some extent.

The training accuracy is utilized to train the model, while the validation accuracy is employed to evaluate the performance of the model. In the obtained results, it is observed that the training accuracy is higher than the validation accuracy. This difference highlights significant disparities between the data used for training the model and the data used for evaluation purposes.

**Figure 7** shows the plot of the training accuracy. The y-axis is the accuracy percentage, while the x-axis shows the number of epochs.

**Figure 8** shows the plot of training accuracy and training validation accuracy. The blue line represents the training accuracy, and the orange line shows the training validation accuracy.

**Figure 9** shows the comparison of the training accuracy, the training validation accuracy, and the model's loss function. The loss function evaluates how well the image processing model algorithm performs; it measures the model in predicting the expected outcome. The top blue line is the training accuracy, and the blue line is the training validation accuracy. The bottom green line is the loss.

The plot shows that the loss function is low; it is also indicated on the 25<sup>th</sup> epoch with the value of 0.0024; this means that the image processing model works well.

Evaluation result: The evaluation result shows that the model has an accuracy of 99.29% and a loss of 0.69% or 0.07%. This indicates that after running the evaluation, the model is accurate.

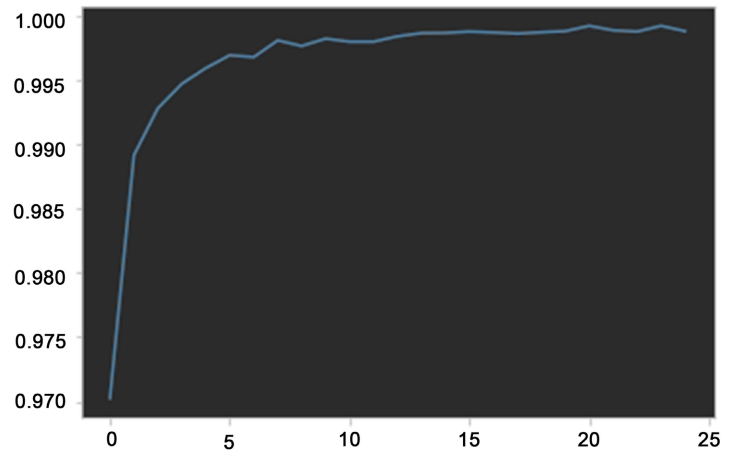


Figure 7. Training accuracy plot (image derived from Jupyter Notebook).

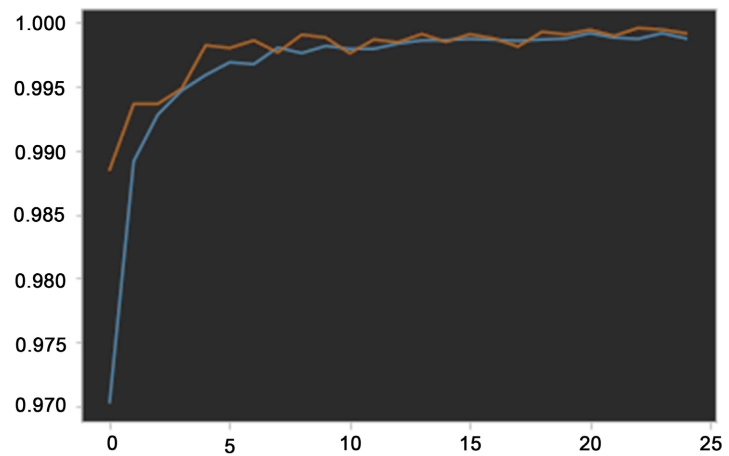


Figure 8. Evaluation result plot (image derived from Jupyter Notebook).

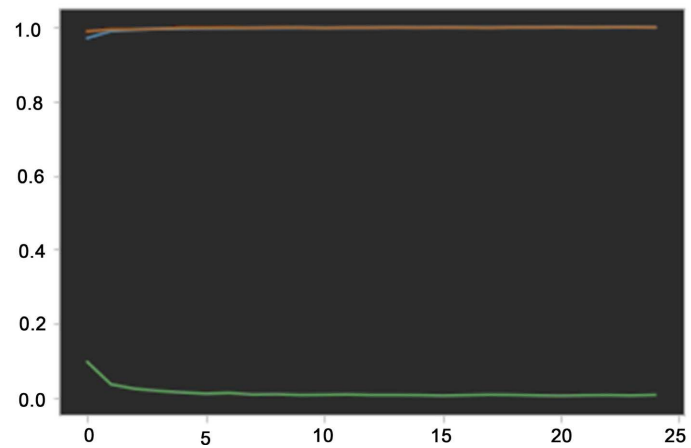


Figure 9. Comparison of training accuracy, training validation accuracy, and the loss function (image derived from Jupyter Notebook).

313/313 [=====] - 2 s 6 ms/step - loss: 0.0695 - accuracy: 0.9929

Result: These numbers represent a loss of 0.06949 and an accuracy of 0.9929; this shows that there is a low loss and a high accuracy, which indicates that the model is performing well.

[0.06949020177125931, 0.992900013923645]

The CNN training and evaluation yielded highly accurate results, with an accuracy rate of 99.29%. However, one of the challenges encountered in the image processing model is the presence of numerous parameters, which can potentially lead to overfitting. Overfitting occurs when the model is trained extensively on the dataset to the extent that it learns irrelevant information or “noise.” Consequently, the model becomes excessively tailored to the training data and fails to perform optimally as intended.

### 3.6.5. Enhancing the Model

To mitigate the issue of overfitting, a technique called “Dropout” is incorporated into the model. Dropout involves randomly deactivating neurons in a layer during the training phase. By applying a dropout probability (commonly set to 0.5), approximately half of the neurons are deactivated during training, preventing the network from relying too heavily on any individual neuron. It is important to note that dropout is only applied during the training process and not during evaluation or prediction.

To augment the image data and increase the diversity of the training set, a technique called image augmentation is employed. Image augmentation involves applying various transformations to the existing images in the dataset, resulting in additional transformed copies of each image. This augmentation process introduces more image variations into the training data, making the model more robust and stable.

In this research, two image generators are used to create different versions of the training dataset. These generators apply modifications such as changes in height and width by 0.2, zooming by 0.2, and image flipping. By running these image generators, the images undergo transformations and generate additional datasets. As a result, the research confirms that there are eight images available, with four images in the “knights” class and four images in the “nurses” class, effectively representing the two classes in the dataset.

Result:

Found 8 images belonging to 2 classes.

Checking the sample images results:

Found 8 images belonging to 2 classes.

The process of image augmentation resulted in the generation of additional datasets containing variations of the original eight images. This approach was implemented to enhance the stability of the model and enable it to effectively identify and classify images with diverse variations. By exposing the model to different versions of the same image, it becomes more adept at recognizing and correctly categorizing images despite inherent variations or changes in their visual characteristics. This ensures that the model is capable of generalizing its

learned features and effectively handles variations within the dataset, leading to improved overall performance.

### 3.7. VGG16

The Image Processing Model employed in this study was based on the VGG16 architecture, developed by the Visual Geometry Group (VGG) at Oxford University, renowned for its success in winning the 2014 ImageNet competition. VGG16 is a relatively smaller and faster model compared to its counterpart, VGG19. Leveraging the capabilities of VGG16, the model created in this research was designed to recognize objects present in random images. To facilitate the implementation of VGG16, the model imported relevant libraries from Keras. As this was the first time VGG16 was utilized in this specific model, the pre-trained weights were obtained from the Imagenet database.

Given that VGG16 requires images of size  $224 \times 224$  as input, the model ensured that the images provided to VGG16 were appropriately resized by specifying the target size within the model. The images were converted into NumPy arrays, enabling training and classification in batches using CNN. To ensure compatibility with the model's requirements, an additional dimension was added to the images using the "expand\_dims" function available in NumPy. This process of expanding the array shape helps prevent errors during prediction by ensuring that the input data possesses the expected four dimensions as prescribed by the model. This was achieved by expanding the array shape and inserting a new axis with a value of zero at the specified position.

Output on Converting to a NumPy array:

(224, 224, 3)

Output after expanding the dimensions:

(1, 224, 224, 3)

### 3.8. Preprocessing the Image

After expanding the dimensions of the input images, the next crucial step is pre-processing, which involves encoding batches of images. In this study, the images utilized are sourced from the VGG16 package provided in Keras. The image processing model leverages these images from the VGG16 package to recognize various objects. The primary purpose of incorporating the VGG16 package is to utilize the ImageNet dataset as the training data for the model.

Subsequently, the images are converted into a NumPy array since Python employs the format of height, width, and channel for image representation. This conversion ensures uniformity in size for all images fed into the model. Following the execution of the model, the output obtained is a NumPy array comprising the probability values associated with the image's classification into one of the 1000 categories present in the Caffe Distribution [51].

To determine the prediction made by the model, the decode\_predictions method is employed, providing the top ten predictions. To evaluate the model's

performance on an unseen image, a random image of a fork is utilized, which has not been encountered by the model during training.

Regarding the preprocessing output, this step involves encoding a batch of images and processing them according to the specified channel order, which can be either channel first or channel last. The VGG16 package includes the necessary preprocess input images for this purpose. The subsequent numbers presented represent the numerical representations of the array, which serves as a data structure capable of storing multiple items of the same type. In the context of the image processing model algorithm, this data structure plays a crucial role.

Top ten prediction results:

```
[[('n04270147', 'spatula', 0.19171521),
 ('n04154565', 'screwdriver', 0.13400662),
 ('n03804744', 'nail', 0.07883306),
 ('n04208210', 'shovel', 0.044209614),
 ('n03481172', 'hammer', 0.037919484),
 ('n03658185', 'letter_opener', 0.03411597),
 ('n03759954', 'microphone', 0.02908787),
 ('n03532672', 'hook', 0.025673749),
 ('n04367480', 'swab', 0.022219818),
 ('n02906734', 'broom', 0.019698625)]]
```

Upon subjecting the fork image to prediction using the VGG16 model, it was determined that the image bears a 19.17% resemblance to a “spatula.” This outcome can be attributed to the absence of a specific “fork” category within the synsets provided. In order to further assess the model’s performance, another image is subsequently tested. This time, the image selected belongs to one of the categories encompassed by the synsets presented in the study by Jia *et al.* [51]. The image under consideration depicts a bee.

Another image prediction results:

```
[[('n02206856', 'bee', 0.9463092),
 ('n02190166', 'fly', 0.049911786),
 ('n01773549', 'barn_spider', 0.0010194371),
 ('n03530642', 'honeycomb', 0.0009426607),
 ('n01773797', 'garden_spider', 0.0004893371),
 ('n07730033', 'cardoon', 0.00036843063),
 ('n02219486', 'ant', 0.00022792102),
 ('n02177972', 'weevil', 0.00021030655),
 ('n01833805', 'hummingbird', 0.00013077597),
 ('n02169497', 'leaf_beetle', 7.780898e-05)]]
```

The evaluation of the image depicting a bee using the image processing model yielded a prediction accuracy of 94.63%. This indicates a high level of confidence in the model’s ability to correctly identify the image. Additionally, the model assigned a prediction probability of 7.780898e-05, which is equiva-

lent to 0.0072571%, to the leaf\_beetle category.

## 4. Safety Framework

In the proposed experimental research, a vehicle will be equipped with both LiDAR and Non-LiDAR technologies. The hardware used should have easy installation and removal capabilities. To ensure unbiased data collection, the environment and driver will remain consistent across the experiments when creating a table for weather conditions. The study will involve data collection under six different weather conditions: sunny (S), cloudy (C), daytime rainy (DR), foggy (F), nighttime rainy (NR), and snowy (SW). Two scenarios will be examined: one with LiDAR technology mounted on the vehicle and the other with Non-LiDAR technology. The testing locations for both technologies will be identical. A driving course with obstacles will be set up for the experiment. Since the focus of the experiment is on weather conditions, the data collection period will span 52 weeks.

The input data will be obtained from both LiDAR and Non-LiDAR technologies, and an interpreter software will be employed to feed this data into the Image Processing Model. The experimental research aims to evaluate the accuracy of the LiDAR and Non-LiDAR hardware in terms of the data they produce. The Image Processing Model will then predict the outcomes based on this data. Given that the model operates on supervised learning principles, the images to be predicted will already be labeled. The objective of the test is to determine whether the model can correctly identify the labeled images using inputs from the LiDAR and Non-LiDAR technologies. The challenge lies in assessing how effectively these two technologies can capture and feed the image data to the model, considering the specific weather conditions outlined in the tables. Each weather condition will be subjected to 25 runs for each technology, ensuring a substantial amount of data is collected for the experiment.

### 4.1. Scenario 1—LiDAR

The LiDAR technology will be installed on the vehicle and driven through the designated driving course under various weather conditions, including sunny (S), cloudy (C), daytime rainy (DR), foggy (F), nighttime rainy (NR), and snowy (SW). A table will be utilized to record the prediction accuracy percentage for each labeled image. For instance, during a sunny day (S), the image captured by the LiDAR and processed by the image processing model achieved a prediction accuracy of 98%. This value will be entered into the table corresponding to the “S” category. As there will be a total of 25 runs, 25 tables of this nature will be generated. **Table 3** shows the LiDAR weather experiment test blank table for the experimental research.

This table summarizes the average of all 25 tables in **Table 4**. E.g., on a sunny day (S), the average image captured and fed to the image processing model by the LiDAR was 98% accurate; the 98% will be inputted into the table under “S,” and the mean will be calculated.



**Table 3.** LiDAR weather experiment test.

LiDAR	S	C	R	F	DR	NR	SW
S							
C							
R							
F							
DR							
NR							
SW							

**Table 4.** LiDAR weather experiment results.

LiDAR	S	C	R	F	DR	NR	SW	Mean
S								
C								
R								
F								
DR								
NR								
SW								

## 4.2. Scenario 1—Non-LiDAR

The camera is securely mounted on the vehicle and utilized during the vehicle's traversal of the designated driving course under various weather conditions: sunny (S), cloudy (C), daytime rainy (DR), foggy (F), nighttime rainy (NR), and snowy (SW). A table will be employed to record the percentage of accurate predictions for the labelled images. For instance, when encountering a sunny day (S), the captured image will be processed by the camera and fed into the image processing model. If the resulting accuracy of the prediction is 98%, this value will be recorded in the respective table entry for the "S" category. As the experiment will be conducted 25 times, a total of 25 tables will be generated to encompass the outcomes of each run. **Table 5** shows the Camera weather experiment test blank table for the experimental research.

**Table 5** presents a comprehensive summary that calculates the average values derived from the 25 individual tables. For example, considering a sunny day (S), the average accuracy of the images captured and processed by the camera and subsequently fed into the image processing model was found to be 98%. This average accuracy value of 98% will be recorded within the corresponding entry in the table under the "S" category. The mean value will be calculated based on the collected data from all the tables, providing a consolidated representation of the overall performance (**Table 6**).

**Table 5.** Camera weather experiment test.

Camera	S	C	R	F	DR	NR	SW
S							
C							
R							
F							
DR							
NR							
SW							

**Table 6.** Camera weather experiment results.

Camera	S	C	R	F	DR	NR	SW	Mean
S								
C								
R								
F								
DR								
NR								
SW								

The outcomes of this experiment will serve as the foundation for conducting a comparative analysis between LiDAR and Non-LiDAR technologies implemented in a driver-operated vehicle. The collected data will be utilized for the purpose of comparing and contrasting these technologies with an Autonomous Vehicle system. By examining and contrasting the performance of LiDAR and Non-LiDAR in the context of a human-driven vehicle, insights can be gained regarding their effectiveness and potential advantages when compared to autonomous driving systems.

### 4.3. Safety Framework for Autonomous Vehicle

The forthcoming experiment will employ the same weather conditions and scenarios as the previous one, albeit without the involvement of a human driver. The focus will be on evaluating the performance of LiDAR and Non-LiDAR technologies within an autonomous vehicle (AV) context. To facilitate this research, a collaboration with Waymo and Tesla, proponents of LiDAR and Non-LiDAR technologies respectively, is proposed. These companies possess extensive training and test data, thereby minimizing the training duration since their AVs are already trained. The experiment will be conducted under comparable conditions as those involving a human driver. Subsequently, the results from both experiments will be compared and contrasted. The primary objective is to establish a baseline for assessing whether an AV can effectively navigate

around obstacles on the road, akin to the actions of a human driver. While a human driver would slow down or halt upon encountering an obstruction, LiDAR and Non-LiDAR will record obstacle data for comparison with AV performance.

The proposed experimental research will adhere to the safety principles outlined in Blanar *et al.*'s [10] study on Responsibility Sensitive Safety (RSS). These principles encompass maintaining a safe distance in front of and laterally to obstacles, respecting right-of-way, and exercising caution when encountering roadblocks or signs. Comparable to the actions of a human driver, these principles will be contrasted with AV behavior.

The experimental research aims to enhance the existing Image Processing Model developed in this study. The enhancements include incorporating distance and speed measurements, evaluating the time taken to stop relative to obstacle distance, utilizing GPS measurements for navigating or avoiding obstacles, determining the time required for turning or obstacle avoidance, and identifying and adhering to road signs. These enhancements will significantly contribute to the safety capabilities of AVs in the future.

In summary, the proposed experimental research serves as a suggestion for the future advancement of the image processing model. The model established in this study forms a foundational framework for subsequent research. While the model primarily focuses on image identification—an essential attribute for AV safety—the suggested enhancements will bolster AV safety measures. The results obtained from the developed image processing model will serve as the groundwork for formulating the safety framework for AVs. However, these outcomes alone are insufficient to instill trust and confidence among drivers, pedestrians, and society at large. Since LiDAR and Non-LiDAR technologies serve as the “eyes” of AVs, similar to human vision, it is crucial to compare the reactions of AVs with those of human drivers based on established rules. This comparative analysis will aid in establishing a certain level of trust in AVs. The tables presented within the safety framework will provide tangible and verifiable data, facilitating a clear comparison and contrast between the two technologies through the developed image processing model in this research.

## 5. Conclusions

In the introductory section, the study provided an overview of different LiDAR and Non-LiDAR technologies, as well as an exploration of Driver Assistance Technologies and the Framework on Measuring Automated Vehicle Safety. The background section examined existing research on the two prominent technologies in Automated Vehicles, namely LiDAR and Non-LiDAR, and highlighted current advancements in the field. Additionally, insights into Image Recognition models and architectures, including Convolutional Neural Networks (CNNs), were discussed.

The methodology section established the research framework and presented the theoretical background on Machine Learning and Artificial Intelligence, with

a specific focus on Image Processing. Furthermore, the Image Processing Model was elaborated upon, involving the development of an algorithm to identify input images from both LiDAR and Non-LiDAR technologies.

Subsequently, the images collected from the two technologies were fed into the algorithm for prediction. Given that the algorithm employed a supervised learning model, the images were appropriately labeled, enabling accurate measurement of input accuracy for each technology. The collected images were then organized into tables based on weather conditions, and each technology underwent 25 runs under various weather conditions. The results from these 25 runs were averaged and compiled into tables, facilitating a comprehensive comparison and contrast between the two technologies.

The image processing model developed in this research served as the foundation for data collection from both technologies using a vehicle driven by a human driver in a simulated environment. The same tests and scenarios outlined in the Safety Framework were also conducted with Autonomous Vehicles (AVs). The resulting data was subsequently compared and contrasted, taking into account the safety framework known as Responsibility Sensitive Safety (RSS). The RSS rules, which encompass maintaining safe distances, respecting right-of-way, and exercising caution in the presence of obstacles or road signs, were employed to evaluate the safety of AVs in comparison to human drivers. The primary objective of the safety framework was to establish trust and confidence among drivers, pedestrians, and society, thereby facilitating the acceptance of AVs on the road.

The research introduced a safety framework to measure the image input of LiDAR and Non-LiDAR technologies. This measurement was derived from the prediction of the image classification model developed in this study. The model's results were incorporated into safety framework tables, which collated the identified images under specific weather conditions. The weather conditions considered were sunny (S), cloudy (C), daytime rainy (DR), foggy (F), nighttime rainy (NR), and snowy (SW). The safety framework tables encompassed two scenarios: scenario 1 involved LiDAR-based technology, while scenario 2 involved Non-LiDAR-based technology. The aggregated results from these scenarios were then compared and contrasted, serving as a baseline for AVs. The AVs, using the same simulation and vehicle but without a human driver, underwent the same scenarios, and the corresponding data was collected and processed in a manner similar to that of a human-driven vehicle. However, improvements to the image processing model, as mentioned earlier in this chapter, were made to enhance security and align with the principles of Responsibility Sensitive Safety (RSS). The results obtained from the AVs' simulation were subsequently compared to those of the human-driven vehicle simulation, thereby addressing the hypothesis that there is no difference in the speed and accuracy of image recognition based on the two technologies under varying weather conditions in an Autonomous Vehicle.

This study proposes several areas for enhancing the image processing model

in order to provide a more comprehensive analysis. These recommendations include:

1) Incorporating Distance Measurement: The image processing model should be expanded to include the ability to accurately measure distances between objects and obstacles in the captured images.

2) Adding Speed Measurement: Enhancements should be made to enable the image processing model to measure the speed of objects and vehicles present in the images.

3) Considering Time of Stop Relative to Obstacle Distance: The model should be improved to capture and analyze the time duration for which a vehicle remains stationary in relation to the distance of the obstacle encountered.

4) Integrating GPS Measurement for Turning or Obstacle Avoidance: The image processing model should incorporate GPS data to accurately measure the trajectory of a vehicle during turning maneuvers or when avoiding obstacles.

5) Analyzing Time of Turning or Obstacle Avoidance: Enhancements should be made to enable the model to analyze and measure the time taken by a vehicle to complete turning maneuvers or navigate around obstacles.

6) Identifying and Tracking Road Signs: The image processing model should be enhanced to identify and track road signs present in the captured images, allowing for more comprehensive analysis of the environment.

Simulation Comparison and Responsibility Sensitive Safety (RSS). The proposed simulation involving a human driver serves as a valuable baseline for comparing and contrasting the performance of LiDAR and Non-LiDAR technologies. The image processing model, which is based on supervised learning, plays a crucial role in facilitating this comparison. However, in order to effectively compare the simulation involving a human driver with that of an Autonomous Vehicle (AV), further enhancements to the image processing model are necessary. The current focus of the developed image processing model on image classification is insufficient for a comprehensive comparison and contrast with a human driver. Therefore, additional enhancements are required.

Once the image processing model has been enhanced to address these limitations, AVs can be tested following the Responsibility Sensitive Safety (RSS) rules outlined in this research. By adhering to these rules, the performance of AVs can be evaluated and compared to that of human drivers. This evaluation aims to measure the safety of AVs in order to build trust and confidence among drivers, pedestrians, and society, ultimately promoting the wider acceptance of AVs on the road.

## Acknowledgement

I would like to thank my former professors for their guidance. Their unrelentless tenacity and patience are valuable contributions to this manuscript. For my colleagues for expanding my horizon regarding Artificial Intelligence and Machine Learning. To the Library Staff who guided me in my pursuit of research. To my family, who tirelessly supports me in all of my endeavours. To Brother Eduardo

V. Manalo for his spiritual guidance. And above all else, to our Almighty God, who made this journey possible and attainable.

### Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

### References

- [1] Da Vinci Inventions (2022) Self-Propelled Cart. <https://www.da-vinci-inventions.com/self-propelled-cart>
- [2] Hartmans, A. (2016) How Google's Self-Driving Car Project Rose from a Crazy Idea to a Top Contender in the Race toward a Driverless Future. Business Insider. <https://www.businessinsider.com/google-driverless-car-history-photos-2016-10>
- [3] Hawkins, A.J. (2017) How Tesla Changed the Auto Industry Forever. Autopilot, Electrification, and a Boot-to-Bonnet Commitment to Technology. The Verge. <https://www.theverge.com/2017/7/28/16059954/tesla-model-3-2017-auto-industry-influence-elon-musk>
- [4] Ziegler, C. (2015) Tesla's Cars Can Drive Themselves Starting Tomorrow. The Verge. <https://www.theverge.com/2015/10/14/9533539/teslas-cars-can-drive-themselves-starting-tomorrow>
- [5] NHTSA: National Highway Traffic Safety Administration (2022) Automated Vehicles for Safety. What Does This Mean for You as a Driver? <https://www.nhtsa.gov/technology-innovation/automated-vehicles-safety#topic-road-self-driving>
- [6] NHTSA: National Highway Traffic Safety Administration (2022) Automated Vehicles for Safety. NHTSA Is Dedicated to Advancing the Lifesaving Potential of New Vehicle Technologies. <https://www.nhtsa.gov/technology-innovation/automated-vehicles-safety#topic-road-self-driving>
- [7] NHTSA: National Highway Traffic Safety Administration (2022) Automated Driving Systems 2.0 Voluntary Guidance. <https://www.nhtsa.gov/document/automated-driving-systems-20-voluntary-guidance>
- [8] United States department of Transportation (2019) Preparing for the Future of Transportation: Automated Vehicles 3.0. <https://www.transportation.gov/av/3>
- [9] United States department of Transportation (2020) Ensuring American Leadership in Automated Vehicle Technologies. Automated Vehicles 4.0. <https://www.transportation.gov/sites/dot.gov/files/docs/policy-initiatives/automated-vehicles/360956/ensuringamericanleadershipav4.pdf>
- [10] Blonar. L.F., Blumenthal. M.S., Anderson. J.M. and Kalra. N. (2018) Rand Corporation. Measuring Automated Vehicle Safety. [https://www.rand.org/content/dam/rand/pubs/research\\_reports/RR2600/RR2662/RAND\\_RR2662.pdf](https://www.rand.org/content/dam/rand/pubs/research_reports/RR2600/RR2662/RAND_RR2662.pdf)
- [11] SAE (2018) SAE International Releases Updated Visual Chart for Its "Levels of Driving Automation" Standard for Self-Driving Vehicles. <https://www.sae.org/news/press-room/2018/12/sae-international-releases-updated-visual-chart-for-its-levels-of-driving-automation-standard-for-self-driving-vehicles>

- [al-chart-for-its-%E2%80%9Clevels-of-driving-automation%E2%80%9D-standard-for-self-driving-vehicles](#)
- [12] Landmark Dividend (2022) Self-Driving Car Technology: How Do Self-Driving Cars Work? Landmark Dividend, News & Insights. <https://www.landmarkdividend.com/self-driving-car/#:~:text=Self%2Ddriving%20vehicles%20employ%20a.navigate%20safely%20on%20our%20roads>
- [13] Blonar, L.F. and Holliday, M. (2020) Approaching Operational Design Domains for Automated Vehicles. RAND. [https://www.rand.org/blog/2020/08/how-state-and-local-governments-might-consider-approaching.html#:~:text=An%20operational%20design%20domain%20\(ODD,is%20designed%20to%20operate%20safely](https://www.rand.org/blog/2020/08/how-state-and-local-governments-might-consider-approaching.html#:~:text=An%20operational%20design%20domain%20(ODD,is%20designed%20to%20operate%20safely)
- [14] Honda (2020) Honda Receives Type Designation of Level 3 Automated Driving in Japan. Global Honda, Newsroom. <https://global.honda/newsroom/news/2020/4201111eng.html>
- [15] Sugiura, E. (2021) Honda Launches World's First Level 3 Self-Driving Car. Nikkei Asia. <https://asia.nikkei.com/Business/Automobiles/Honda-launches-world-s-first-level-3-self-driving-car>
- [16] Bigg, M. (2021) Honda Is Beating Tesla in a Driverless Car Race. Car Buzz. <https://carbuzz.com/news/honda-is-beating-tesla-in-driverless-car-race>
- [17] Wasser, L.A. (2020) The Basics of Lidar-Light Detection and Ranging-Remote Sensing. Neon Science. <https://www.neonscience.org/lidar-basics>
- [18] Siddiqui, F. (2020) Tesla Is Putting 'Self-Driving' in the Hands of Drivers Amid Criticism the Tech Is Not Ready. Washington Post. <https://www.washingtonpost.com/technology/2020/10/21/tesla-self-driving/>
- [19] Musk, E. (2020) Tweet: FSD  $\beta$  Rollout Happening Tonight. Twitter. <https://twitter.com/elonmusk/status/1318678258339221505?s=20>
- [20] Zywanowski, K., Banaszczyk, A. and Nowicki, M. (2020) Comparison of Camera-Based and 3D LiDAR-Base Place Recognition across Weather Conditions. 2020 16th International Conference on Control, Automation, Robotics and Vision (ICARCV), Shenzhen, 13-15 December 2020, 886-891. <https://doi.org/10.1109/ICARCV50220.2020.9305429>
- [21] Mugunthan, N., Balaji, S.B., Harini, C. and Prasanna, V.V. (2020) Comparison Review on LiDAR vs Camera in Autonomous Vehicle. *International Research Journal of Engineering and Technology (IRJET)*, 7, 4242-4246.
- [22] Fritz ai (2022) Image Recognition Guide. Fritz ai.
- [23] Du, S., Guo, H. and Simpson, A. (2019) Self-Driving Car Steering Angle Prediction Based on Image Recognition. Cornell University. arXiv: 1912.05440.
- [24] Kumar, A. (2021) Different Types of CNN Architectures Explained: Examples. Data Analytics. <https://vitalflux.com/different-types-of-cnn-architectures-explained-examples/>
- [25] Dang, A. (2021) Top 10 CNN Architectures Every Machine Learning Engineer Should Know. <https://towardsdatascience.com/top-10-cnn-architectures-every-machine-learning-engineer-should-know-68e2b0e07201>
- [26] Library of Congress (2022) Who Invented the Automobile? <https://www.loc.gov/everyday-mysteries/item/who-invented-the-automobile/>
- [27] Barla, N. (2021) Self-Driving Cars with Convolutional Neural Networks (CNN).



- Neptuneblog.  
<https://neptune.ai/blog/self-driving-cars-with-convolutional-neural-networks-cnn>
- [28] Wenhui, Y. and Fan, Y. (2017) Lidar Image Classification Based on Convolutional Neural Networks. 2017 *International Conference on Computer Network, Electronic and Automation (ICCNEA)*, Xi'an, 23-25 September 2017, 221-225.
- [29] Zhang, X., Fu, H. and Dai, B. (2019) Lidar-Based Object Classification with Explicit Occlusion Modeling. Cornell University. arXiv: 1907.04057.
- [30] Song, W., Zou, S., Tian, Y., Fong, S. and Cho, K. (2018) Classifying 3D Objects in LiDAR Point Cloud with Back-Propagation Neural Network. *Human-Centric Computing and Information Sciences*, **8**, Article No. 29.
- [31] Song, W., Zhang, L., Tian, Y., Fong, S., Liu, J. and Gozho, A. (2020) CNN-Based 3D Object Classification Using Hough Space of LiDAR Point Clouds. *Human-Centric Computing and Information Sciences*, **10**, Article No. 19.  
<https://doi.org/10.1186/s13673-020-00228-8>
- [32] HIPR2 (2004) Hough Transform. Image Processing Learning Resources.  
<https://homepages.inf.ed.ac.uk/rbf/HIPR2/hough.htm>
- [33] Rouse, M. (2017) Rasterization. Techopedia.  
<https://www.techopedia.com/definition/13169/rasterization>
- [34] Tesla (2022) Tesla Vision Update: Replacing Ultrasonic Sensors with Tesla Vision.  
[https://www.tesla.com/en\\_CA/support/transitioning-tesla-vision#:~:text=Beginning%20with%20deliveries%20in%20May,and%20certain%20active%20safety%20features](https://www.tesla.com/en_CA/support/transitioning-tesla-vision#:~:text=Beginning%20with%20deliveries%20in%20May,and%20certain%20active%20safety%20features)
- [35] Jahromi, B.S. (2019) Camera Technology in Self-Driving Cars.  
<https://medium.com/@BabakShah/camera-technology-in-self-driving-cars-610371db4b0b>
- [36] Fujiyoshi, H., Hirakawa, T. and Yamashita, T. (2019) Deep Learning-Based Image Recognition for Autonomous Driving. *IATSS Research*, **43**, 244-252.  
<https://doi.org/10.1016/j.iatssr.2019.11.008>
- [37] Tyagi, M. (2021) HOG (Histogram of Oriented Gradients): An Overview.  
<https://towardsdatascience.com/hog-histogram-of-oriented-gradients-67ecd887675f>
- [38] Gandhi, R. (2018) Support Vector Machine—Introduction to Machine Learning Algorithms.  
<https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47>
- [39] Leonard, J. (2019) Image Classification and Object Detection Algorithm Based on Convolutional Neural Network. *Science Insights*, **31**, 85-100.  
<https://doi.org/10.15354/si.19.re117>
- [40] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G.S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Zheng, X., *et al.* (2016) TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. arXiv: 1603.04467.
- [41] Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J. and Chintala, S. (2019) PyTorch: An Imperative Style, High-Performance Deep Learning Library. In: Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E. and Garnett, R., Eds., *Advances in Neural Information Processing Systems*, Neural Information Processing Systems Foundation Inc. (NeurIPS), Vancouver, 8024-8035.

- [42] Simonyan, K. and Zisserman, A. (2015) Very Deep Convolutional Networks for Large-Scale Recognition. arXiv: 1409.1556.
- [43] He, K., Zhang, X., Ren, S. and Sun, J. (2016) Deep Residual Learning for Image Recognition. 2016 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, 27-30 June 2016, 770-778.  
<https://doi.org/10.1109/CVPR.2016.90>
- [44] Szegedy, C., Liu, W., Jia, Y., Sermanet, P., Reed, S., Anguelov, D., Erhan, D., Vanhoucke, V. and Rabinovich, A. (2015) Going Deeper with Convolutions. 2015 *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, 7-12 June 2015, 1-9. <https://doi.org/10.1109/CVPR.2015.7298594>
- [45] Krizhevsky, A., Sutskever, I. and Hinton, G.E. (2012) ImageNet Classification with Deep Convolutional Neural Networks. *Proceedings of the 25th International Conference on Neural Information Processing Systems*, Carson City, 3-6 December 2012, 1097-1105.
- [46] Boesch, G. (2022) VGG Very Deep Convolutional Networks (VGGNet)—What You Need to Know.  
<https://viso.ai/deep-learning/vgg-very-deep-convolutional-networks/>
- [47] IBM (2020) Machine Learning. IBM Cloud Learn Hub.  
<https://www.ibm.com/cloud/learn/machine-learning>
- [48] Zhao, J., Liang, B. and Chen, Q. (2018) The Key Technology toward the Self-Driving Car. *International Journal of Intelligent Unmanned Systems*, **6**, 2-20.  
<https://doi.org/10.1108/IJIUS-08-2017-0008>
- [49] Fernandes, J. (2018) Neural Networks and Convolutional Neural Networks Essential Training.  
<https://www.linkedin.com/learning/neural-networks-and-convolutional-neural-networks-essential-training/what-you-should-know?autoSkip=true&autoplay=true&resume=false&u=2212217>
- [50] Great Learning (2020) What Is Rectified Linear Unit (ReLU)? Introduction to ReLU Activation Function.  
<https://www.mygreatlearning.com/blog/relu-activation-function/>
- [51] Jia, Y., Shelhamer, E., Donahue, J., Karayev, S., Long, J., Girshick, R., Guadarrama, S. and Darrell, T. (2014) Dafee: Convolutional Architecture for Fast Feature Embedding. *Proceedings of the 22nd ACM International Conference on Multimedia*, New York, November 2014, 675-678.