

# Object Detection System for Vehicles



---

Prepared by

İsmail ÖKSÜZ	150119516
Barış HAZAR	150118019



# Outline

---

- Project Definition
  - Project Usage
  - Milestones
  - Used Frameworks/ Technologies
  - Scope & Constraints
  - Dataset Details
  - Implementation Details
- 
- Data Augmentation
  - Dropout Layer
  - Final Model
  - Test Outputs
  - Summary
  - List of References



# Project Definition

---

This is an image classification project for different types of land vehicles.

It classifies a vehicle from the photograph of the vehicle.

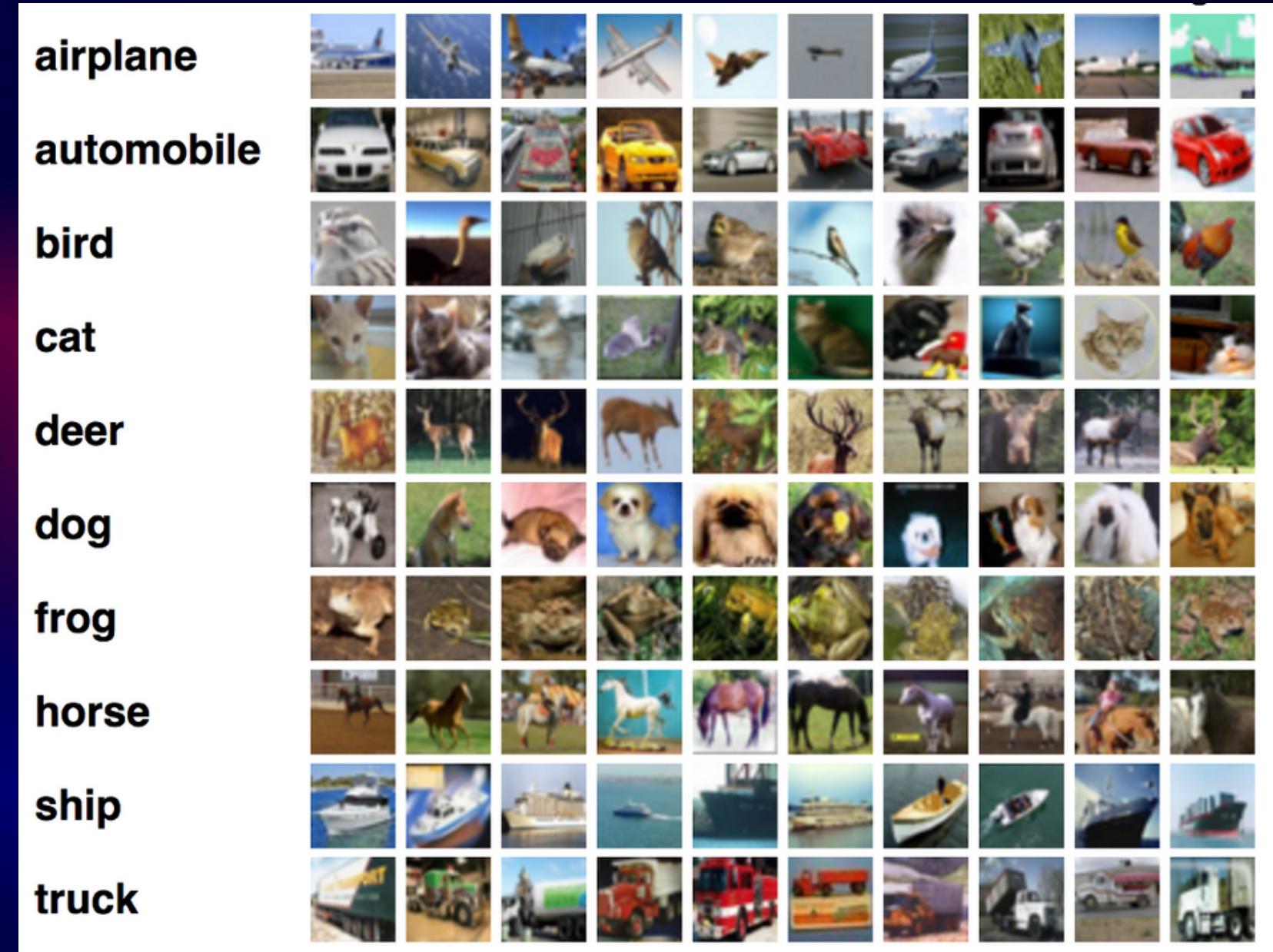


Figure 1. Example of image classification

# Project Usage

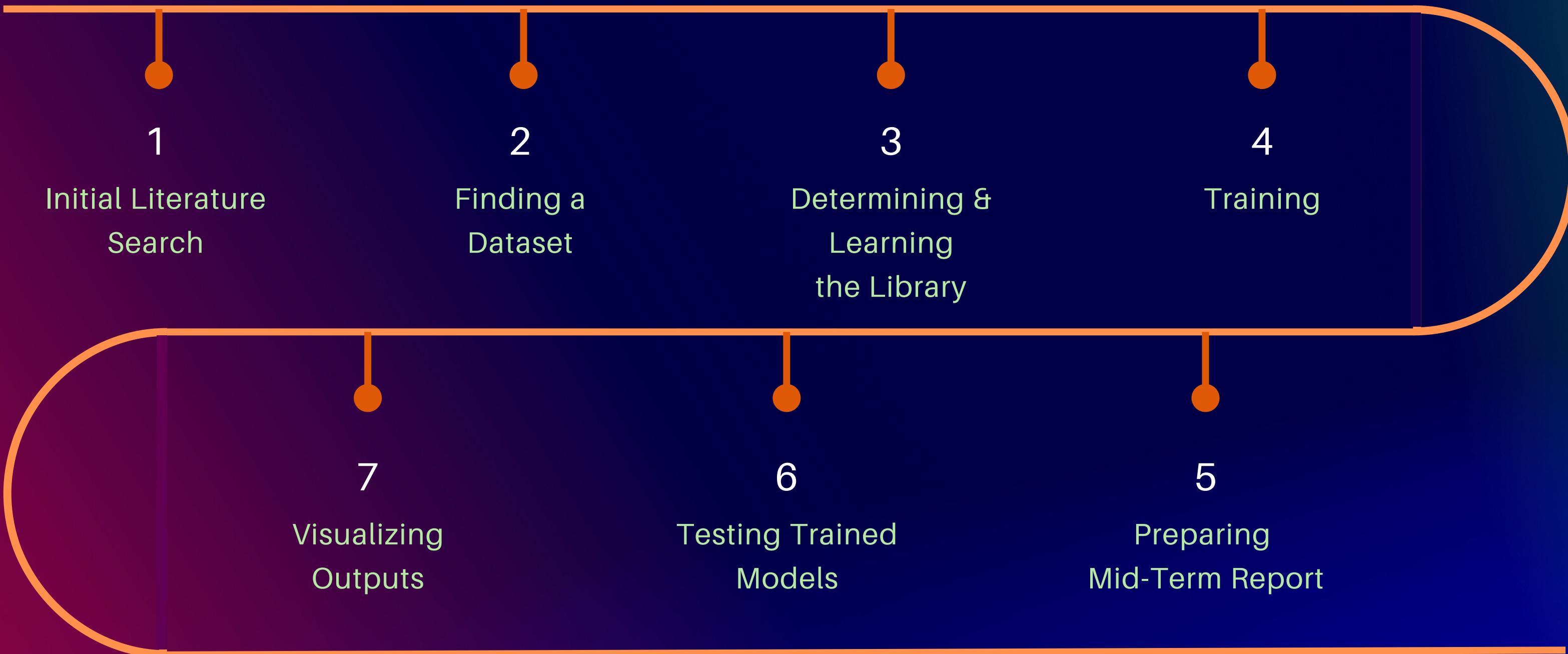
---

It can be used in toll booths or parking lots.



# Milestones

---

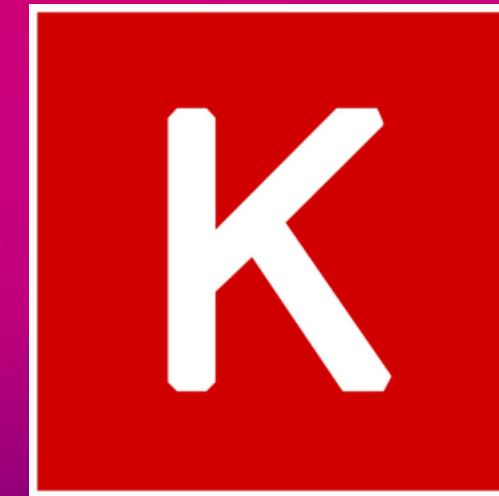


# Used Frameworks/ Technologies

---



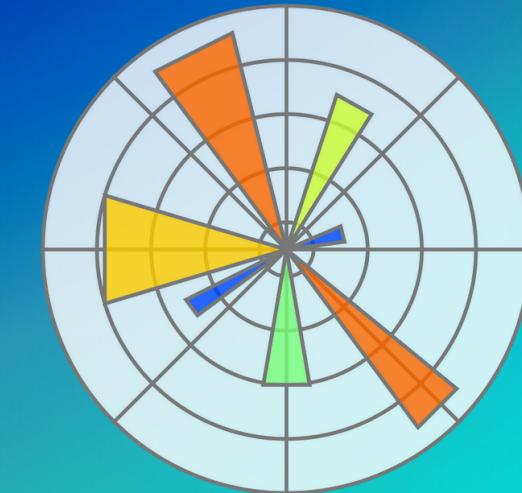
Tensorflow [1]



Keras [3]



Python [2]



Matplotlib [4]



# Scope & Constraints

---

## Scope

The model will classify  
8 types of vehicles.

## Constraints

It will only classify vehicle  
from the images, not from  
any video or livestream.



# Dataset Details

---

We found a dataset from kaggle.com [5].



This dataset contains various types of essential and non-essential vehicles (car, truck, motorcycle, ambulance etc.)

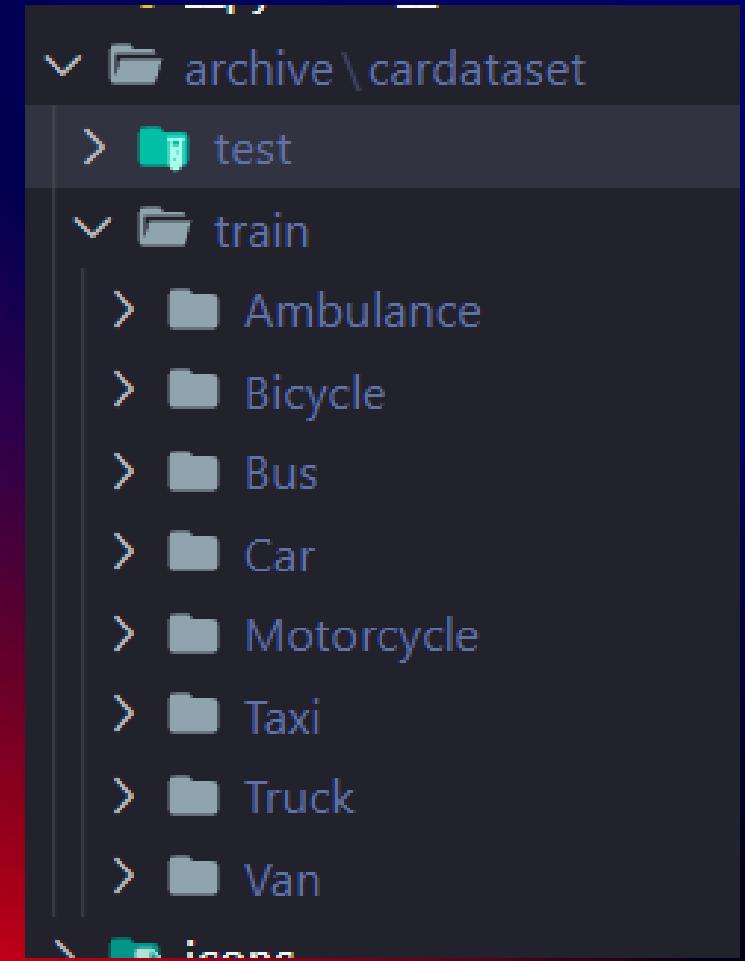


Figure 2. Dataset folder structure

It is more than 5 GB.



# Implementation Details

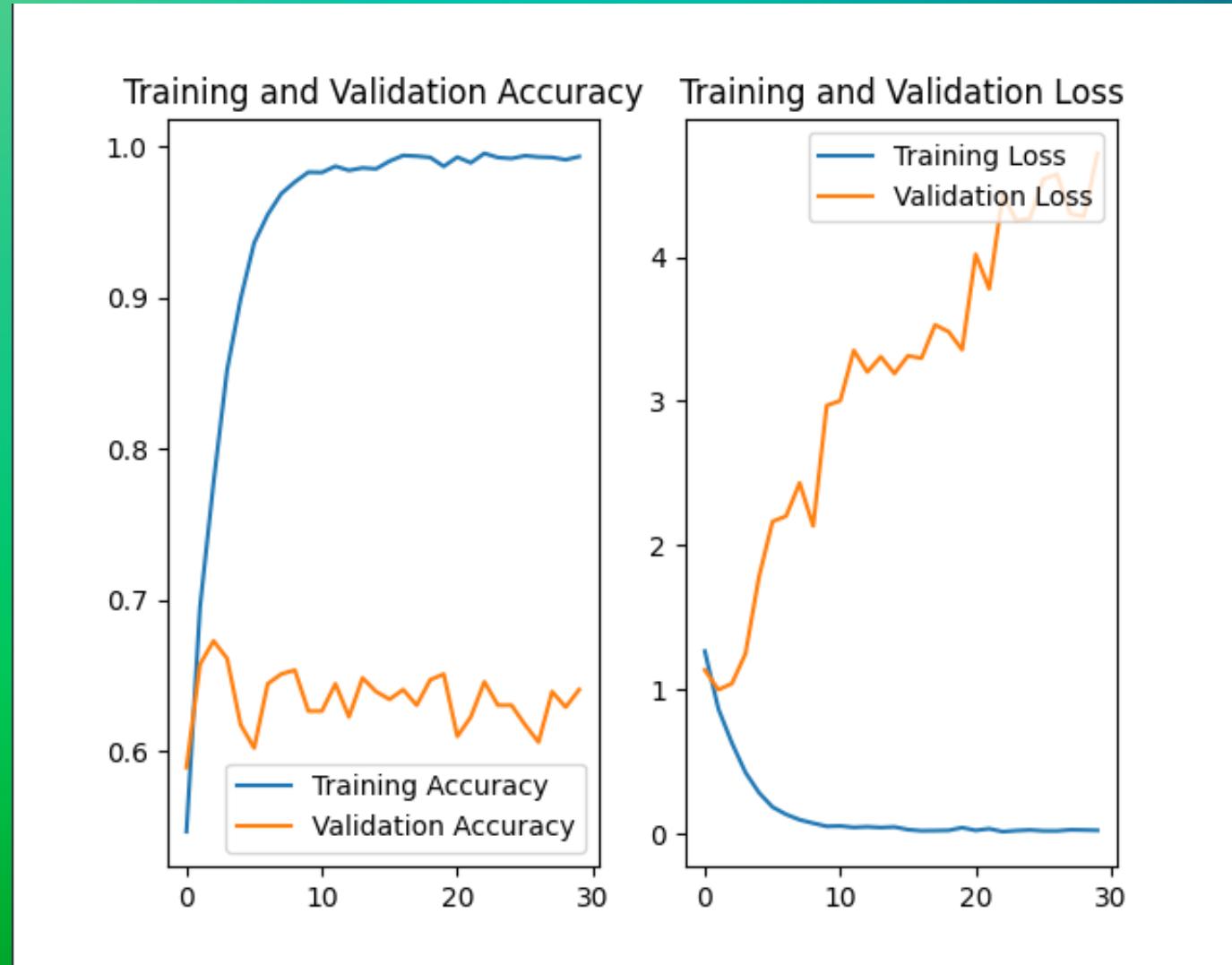


Figure 3. An example of overfitting

At first, when we train our model, using CNN's we had an overfitting issue, because of our small dataset.

We've overcome that issue by using data augmentation and dropouts.

# Data Augmentation

---

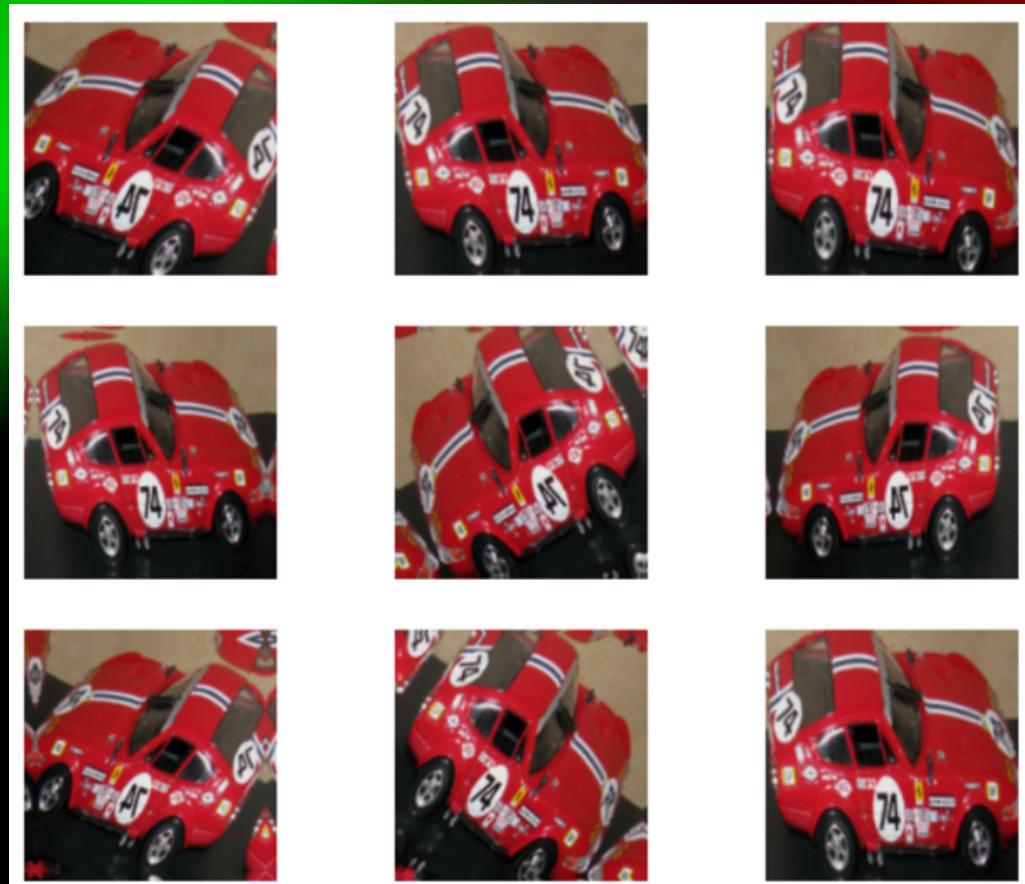


Figure 5. Data augmentation for car

```
# Use Data augmentation
data_augmentation = keras.Sequential([
    layers.RandomFlip("horizontal",
                       input_shape=(img_height,
                                   img_width,
                                   3)),
    layers.RandomRotation(0.2),
    layers.RandomZoom(0.1),
])
```

Figure 4. Data augmentation in code



Figure 6. Data augmentation for bicycle

# Dropout Layer

Dropout is a regularization technique that randomly drops out output units during training to prevent overfitting.



```
model = Sequential([
    data_augmentation,
    layers.Rescaling(1./255, input_shape=(img_width, img_height, 3)),
    layers.Conv2D(16, 3, padding='same', activation='relu'),
    layers.MaxPooling2D(),
    layers.Conv2D(32, 3, padding='same', activation='relu'),
    layers.MaxPooling2D(),
    layers.Conv2D(64, 3, padding='same', activation='relu'),
    layers.MaxPooling2D(),
    layers.Dropout(0.35), # Add dropout
    layers.Flatten(),
    layers.Dense(128, activation='relu'),
    layers.Dense(num_classes, name="outputs")
])
```

Figure 7. Dropout application



# Final Model

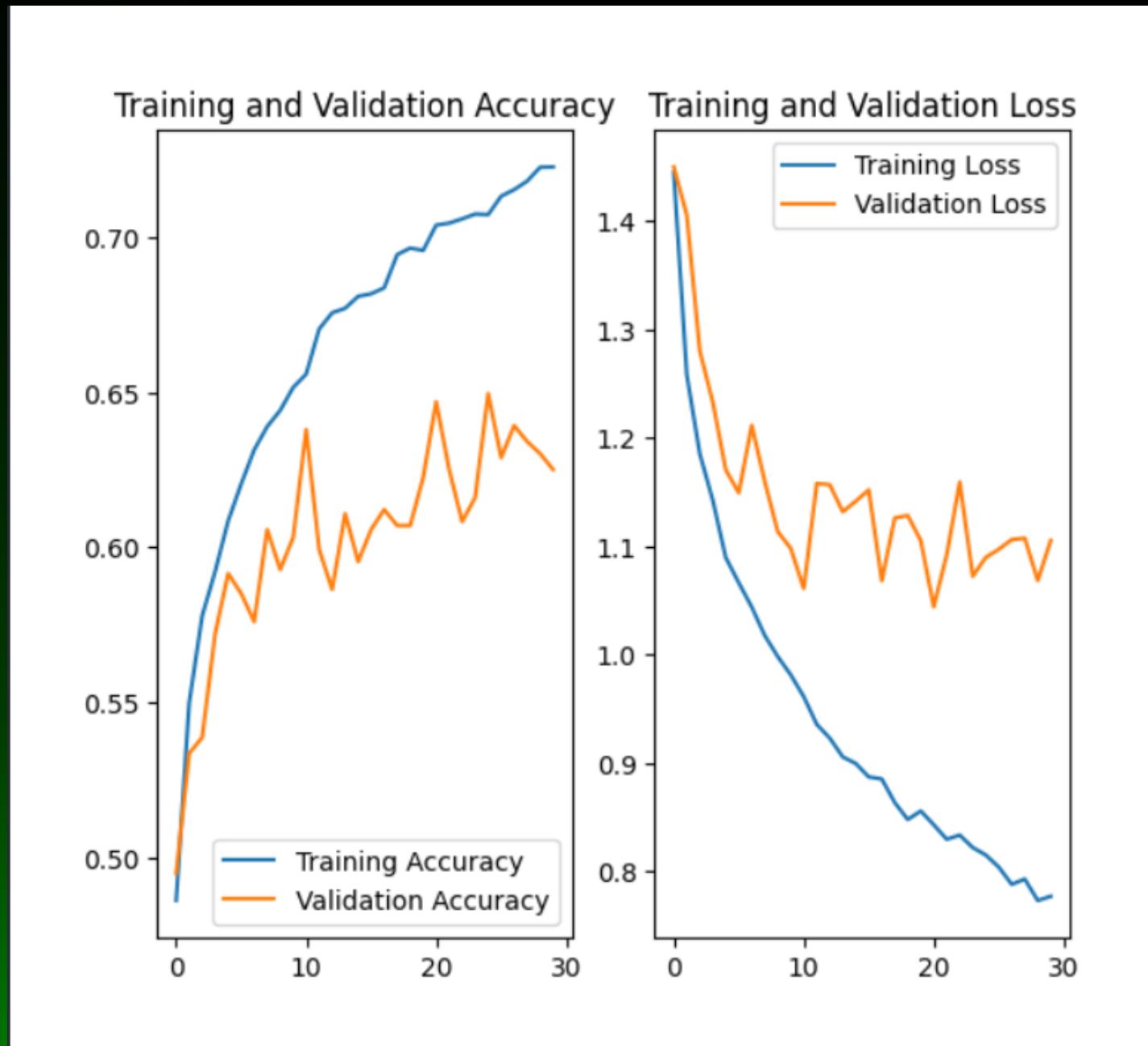


Figure 8. Accuracy of the final model

After applying data augmentation and adding a dropout layer, our model became much better as can be seen from figure 6.

Our final validation accuracy is around almost **65%** after **30** epochs.

# Test Outputs

---



The program guessed the motorcycle correctly with  
**70.06%** confidence.

# Test Outputs

---

The program guessed the truck correctly with **99.18%** confidence.

Top 3 Guesses:  
Truck: 99.18%  
Bus: 0.56%  
Car: 0.20%



# Test Outputs

---



The program guessed the van correctly with  
**36.22%** confidence.



# Test Outputs

---

The program guessed the bus correctly with **90.74%** confidence.



# Test Outputs

---



The program guessed the bus on the left correctly with **60.17%** confidence.

# Test Outputs

---

The program guessed  
the car correctly with  
**91.75%** confidence.

Top 3 Guesses:  
Car: 91.75%  
Truck: 5.58%  
Van: 2.67%



# Test Outputs

---



The program guessed the taxi correctly with **76.51%** confidence.

# Test Outputs

The program guessed the ambulance correctly with **89.95%** confidence.

Top 3 Guesses:  
Ambulance: 89.95%  
Van: 7.89%  
Truck: 1.17%



# Test Outputs

---



The program could not guess  
the ambulance on the left.

# Test Outputs

---

The program guessed the bicycle correctly with **86.58%** confidence.

Top 3 Guesses:  
Bicycle: 86.58%  
Motorcycle: 13.28%  
Car: 0.13%



# Summary

---

In this project, we used Tensorflow Keras library to perform image classification

We learned a lot about Convolutional Neural Networks. And what to do when overfitting occurs, what is data augmentation etc.



# List of References

---

1. <https://www.tensorflow.org/>
2. <https://www.python.org/>
3. <https://www.tensorflow.org/tutorials/images/classification>
4. <https://matplotlib.org/>
5. <https://www.kaggle.com/datasets/iamsandeepprasad/vehicle-data-set?resource=download>



THANKS FOR  
LISTENING