**Week 1: Understanding ARAS and learning ML Fundamentals**

In the first week, I researched and familiarized myself with the following ARAS Features and Machine Learning Fundamentals:

ARAS Features

1. **Blind Spot Detection (BSD)**
2. **Lane Departure Warning (LDW)** and **Lane Keeping Assistance (LKA)**
3. **Lane Change Assist (LCA)**
4. **Forward Collision Warning (FCW)**
5. **Autonomous Emergency Braking (AEB)**
6. **Rear Collision Warning (RCW)**
7. **Rear Cross Traffic Alert (RCTA)**
8. **Traffic Sign Recognition (TSR)** and **Traffic Light Detection**
9. **Rider Drowsiness / Fatigue Monitoring System**
10. **Traction Control System (TCS)**
11. **Anti-lock Braking System (ABS)** and **Cornering ABS**
12. **Hill Hold Control (HHC)**
13. **Adaptive Headlights**
14. **V2X Communication**
15. **Adaptive Cruise Control (ACC)**

ML Fundamentals

* Types of ML (Supervised, Unsupervised, Reinforcement, Semi-supervised)
* Concepts of Classification, Regression

Referred links

For ARAS Features:

https://www.researchgate.net/publication/339819211\_PASSENGER\_DETECTION\_AND\_COUNTING\_FOR\_PUBLIC\_TRANSPORT\_SYSTEM

https://viso.ai/applications/pothole-detection/

https://github.com/hoanglehaithanh/Traffic-Sign-Detection

https://www.analyticsvidhya.com/blog/2021/12/traffic-signs-recognition-using-cnn-and-keras-in-python/

<https://ieeexplore.ieee.org/document/8662019>

For ML Fundamentals

https://developers.google.com/machine-learning/intro-to-ml

https://developers.google.com/machine-learning/crash-course/linear-regression

https://www.geeksforgeeks.org/machine-learning/types-of-machine-learning/

**Week 2: Neural Networks and CNNs**

Studied Feedforward Neural Networks and MLPs

* Components: Input, Hidden, Output layers
* Forward and Backward Propagation
* Activation functions: Tanh, Sigmoid, ReLU, Softmax
* Loss functions: MSE, MAE, Cross-Entropy
* Optimization: Gradient Descent (SGD, Batch, Mini-Batch)

Learnt about Convolutional Neural Networks (CNNs)

* Explored LeNet-5 architecture by Yann LeCun:
  + Layers: Convolution, Pooling, Fully Connected
  + Activation Functions used: Tanh, Softmax
  + Implementation steps: Data preprocessing, model building, compiling, training, evaluation

REFERRED LINKS

https://www.datacamp.com/tutorial/introduction-to-convolutional-neural-networks-cnns

<https://www.geeksforgeeks.org/machine-learning/neural-networks-a-beginners-guide/>

<https://paravisionlab.co.in/lenet-5-architecture/>

<https://www.youtube.com/watch?v=aircAruvnKk>  
<https://www.youtube.com/watch?v=YRhxdVk_sIs>  
<https://www.youtube.com/watch?v=5tvmMX8r_OM>

[Hot Dog or Not Hot Dog – Convolutional Neural Network Course for Beginners](https://www.youtube.com/watch?v=nVhau51w6dM)

**Week 3: Image Classification with CNN**

The objective was to implement a binary classification model for the two classes: Pedestrian and Auto-rickshaw.

**Implementation:**

1. Data Collection:

Images were collected from Google Images and Kaggle.

[Pedestrians Dataset](https://www.kaggle.com/datasets/alincijov/penn-fudan)

[IIT Delhi Campus Pedestrian Dataset (Detection)](https://www.kaggle.com/datasets/aryangarg01/iit-delhi-campus-pedestrian-dataset-detection)

2. Implementation Frameworks:

- TensorFlow with Keras API  
- PyTorch

3. Implementation in Keras

Hyperparameters

Image Size – 256\*256

Batch size – 32

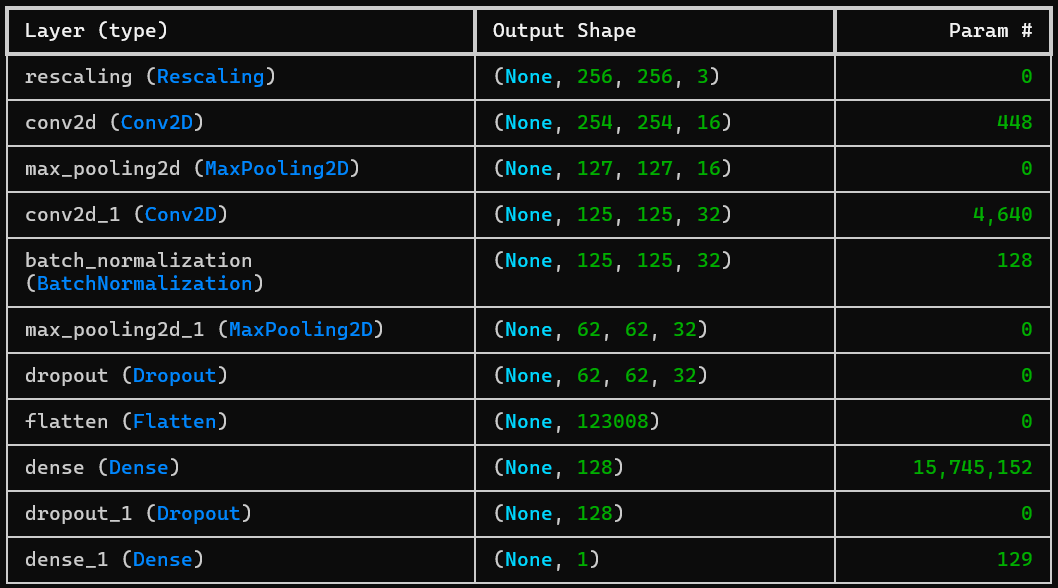
Data split - Split into 70% training, 20% validation, 10% testing.

Optimizer – Adam

Loss function – Binary cross Entropy

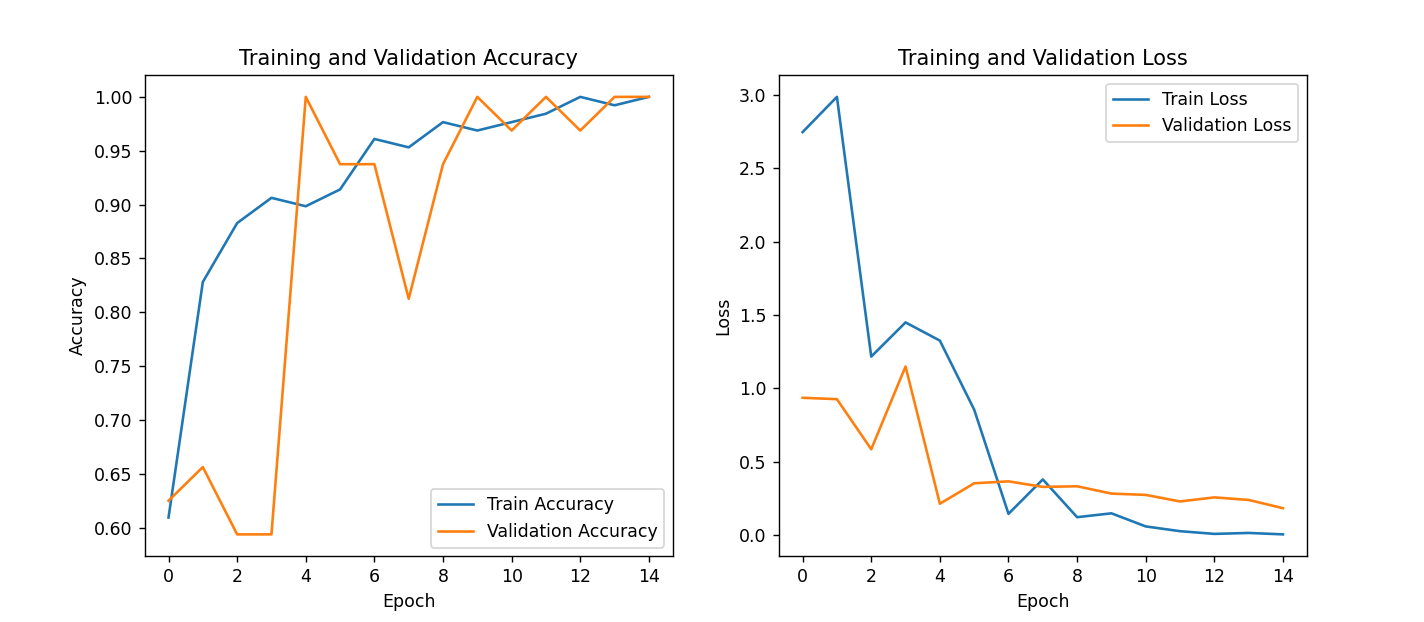
Epochs – 20

Model:

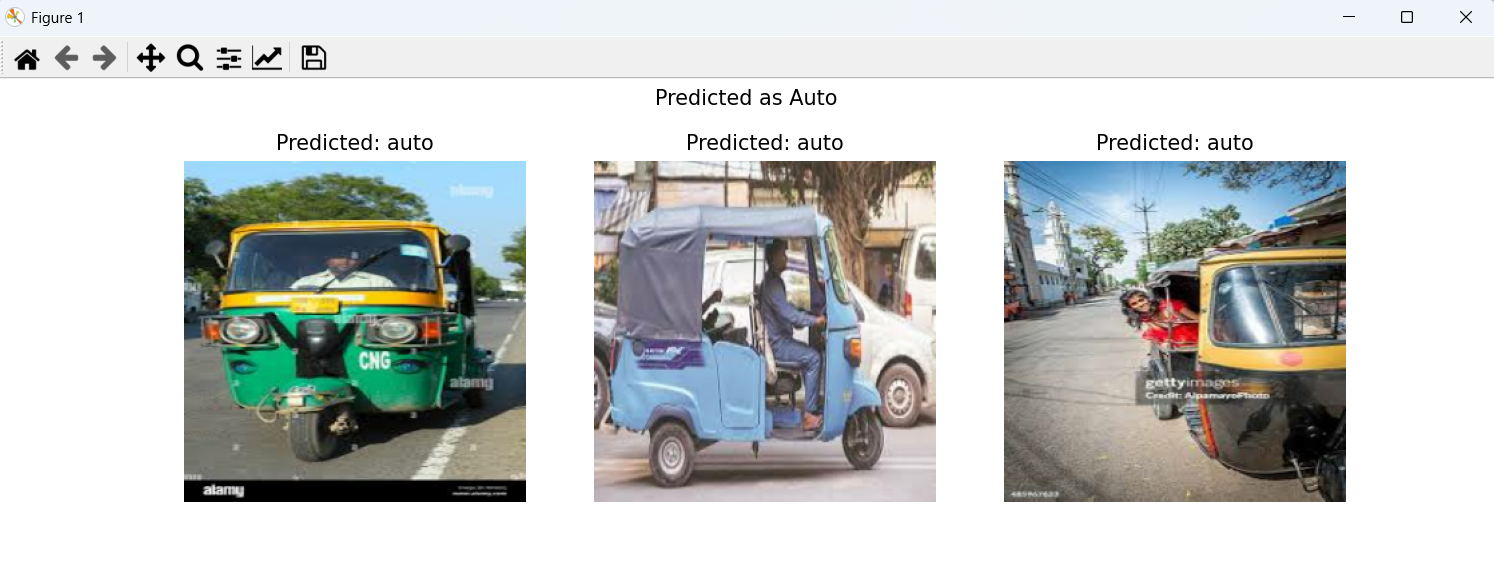


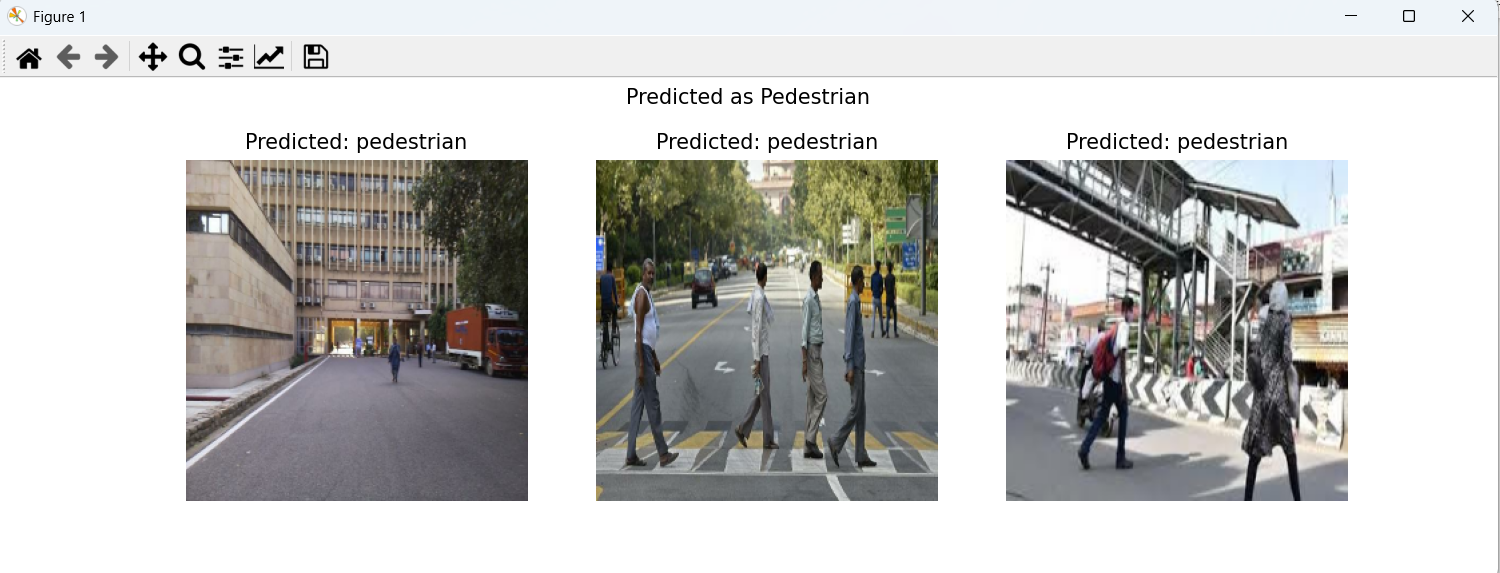
Test Results:





OUTPUT

****

****

Implementation in Pytorch