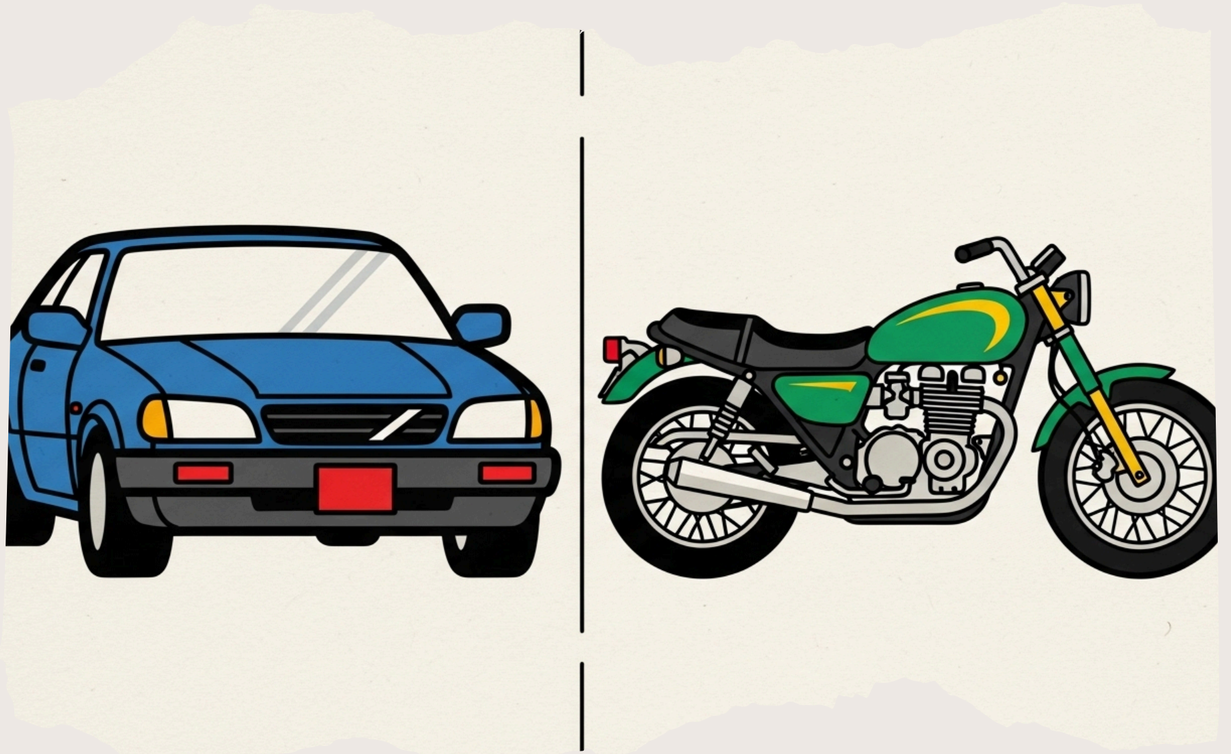
VEHICLE
CLASSIFICATIION

By : Justin Kpana, Isaac Lasso Younes,
and Carl Forman

PROJECT
OBJECTIVE

Our project aimed to build an AI system, leveraging machine learning and deep learning techniques in Python, to accurately classify vehicle images into two categories: cars and motorcycles.



DATASETS

Train and Test

Our dataset consisted of 200 images. Their size was 150 x 150. For different trials, we varied the training and testing split ratios between 60%-40% and 85%-15%. The dataset was obtained from Kaggle.

OUR CODE

```
[ ]  train_dir = "/content/CarMotor_dataset/train"
     test_dir = "/content/CarMotor_dataset/test"

     train_datagen = ImageDataGenerator(
         rescale=1./255,
         rotation_range=20,
         width_shift_range=0.2,
         height_shift_range=0.2,
         shear_range=0.2,
         zoom_range=0.2,
         horizontal_flip=True,
         fill_mode='nearest'
     )

     test_datagen = ImageDataGenerator(rescale=1./255)

     train_generator = train_datagen.flow_from_directory(
         train_dir,
         target_size=(150, 150),
         batch_size=32,
         class_mode='binary'
     )

     test_generator = test_datagen.flow_from_directory(
         test_dir,
         target_size=(150, 150),
         batch_size=32,
         class_mode='binary'
     )
```

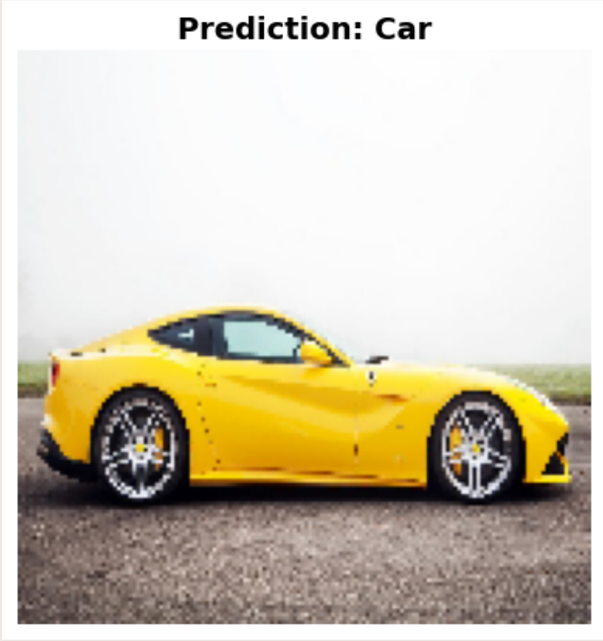
```
model = tf.keras.models.Sequential([
    tf.keras.layers.Conv2D(32, (3,3), activation='relu', input_shape=(150, 150, 3)),
    tf.keras.layers.MaxPooling2D(2,2),

    tf.keras.layers.Conv2D(64, (3,3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2,2),

    tf.keras.layers.Conv2D(128, (3,3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2,2),

    tf.keras.layers.Conv2D(256, (3,3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2,2),

    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(512, activation='relu'),
    tf.keras.layers.Dense(1, activation='sigmoid')  # Binary classification (Cat vs Dog
])

# Compile the model
model.compile(loss='binary_crossentropy',
              optimizer='adam',
              metrics=['accuracy'])
```

```
[ ]  import os, random, shutil

     # Set up paths
     source_path = "CarMotor_dataset/Dataset"
     train_path = "CarMotor_dataset/train"
     test_path = "CarMotor_dataset/test"
     classes = ["Car", "motorcycle"]

     # Make train and test folders
     for cls in classes:
         os.makedirs(os.path.join(train_path, cls), exist_ok=True)
         os.makedirs(os.path.join(test_path, cls), exist_ok=True)

     # Move 80% of images to train, 20% to test
     for cls in classes:
         files = os.listdir(os.path.join(source_path, cls))
         random.shuffle(files)

         split = int(0.8 * len(files))
         train_files = files[:split]
         test_files = files[split:]

         for f in train_files:
             shutil.move(os.path.join(source_path, cls, f), os.path.join(train_path, cls, f))
         for f in test_files:
             shutil.move(os.path.join(source_path, cls, f), os.path.join(test_path, cls, f))
```

RESULTS

EXAMPLES

These are some of the tests we did with our AI classifier



OUR DATA

- Most accurate: 20 Epochs 80% Training and 20% Test
- Least Accurate: 3 epochs with 80%/70% training
- Main factor seems to be epochs because it has less training time

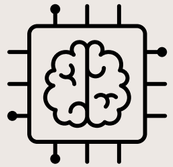
Trial	Epoch	Layers	Batch Size	Split	Accuracy
1	3	11	32	80% - train, 20% - test	50%
2	10	11	32	80% - train, 20% - test	82%
3	20	11	32	80% - train, 20% - test	95%
4	3	9	16	80% - test, 20% train	69%
5	10	9	16	60% - train, 40% - test	75%
6	20	9	16	60% - train, 40% - test	88%
7	3	15	64	70% - train, 30% - test	50%
8	10	15	64	70% - train, 30% - test	53%
9	20	15	64	70% - train, 30% - test	78%
10	30	9	32	50% - train, 50% test	94%
11	10	13	32	85% - train, 15% - test	73%
12	30	13	32	85% - train, 15% - test	91%
13	50	9	32	85% - train, 15% - test	94%

LEARNING EXPERIENCE

We learned how to build an image classifier AI. Additionally, we improved our teamwork skills. One of the challenges we faced was understanding how it worked, which required patience and continuous learning.

Teamwork

Learning



AI

THANK YOU!

GitHub link:
https://github.com/Isaac-tech06/Team-3-AIIE2025