Course-End Project: Automating Port Operations

Project Title: Boat Type Classification for Marina Pier Inc.

---

1. Project Overview

Marina Pier Inc., a port operations company based in San Francisco, aims to modernize its operational pipeline by deploying an automated and intelligent system to classify incoming boats. Human errors, such as misclassification of boat types, can result in faulty reporting, inefficiencies, and even security concerns. To eliminate such risks, the company plans to implement a **deep learning-based image classification system** that can accurately identify the type of boat entering the port.

This project explores two modeling approaches:

- A **Convolutional Neural Network (CNN)** model built from scratch.
- A **transfer learning-based MobileNetV2 model** optimized for mobile deployment.

---

2. Objectives

1. Develop a CNN-based boat classification model.
2. Build a lightweight transfer learning model using MobileNetV2 for mobile deployment.
3. Evaluate and compare the performance of both models.

---

3. Dataset Description

- **Dataset Name**: boat_type_classification_dataset.zip
- **Total Images**: 1162
- **Number of Classes**: 9
    - Buoy
    - Cruise_ship
    - Ferry_boat
    - Freight_boat
    - Gondola
    - Inflatable_boat

    - Kayak
    - Paper_boat
    - Sailboat
- **Data Format**: Directory-structured images by class

---

4. Model 1: CNN from Scratch

4.1 Data Preparation

- Split: 80% training, 20% testing
- Image normalization using image_scale = 1./255
- Batch size: 32
- Loaded using tf.keras.preprocessing.image_dataset_from_directory

4.2 CNN Architecture

text

CopyEdit

```
- Conv2D(32, (3,3), activation='relu') + MaxPooling2D
- Conv2D(32, (3,3), activation='relu') + MaxPooling2D
- GlobalAveragePooling2D
- Dense(128, activation='relu')
- Dense(128, activation='relu')
- Dense(9, activation='softmax')
```

4.3 Training & Evaluation

- Optimizer: Adam
- Loss: Categorical Crossentropy
- Metrics: Accuracy, Precision, Recall
- Epochs: 20
- Plotted training loss and accuracy

4.4 Performance (CNN)

- **Test Accuracy**: 0.34

- **Test Loss**: 1.67
- **Precision**: 0.53
- **Recall**: 0.12
- **Observation**: The CNN model underperformed, possibly due to limited data, lack of regularization, or insufficient depth.

---

5. Model 2: Transfer Learning using MobileNetV2

5.1 Data Preparation

- Split: 70% training, 30% testing
- Image normalization using image_scale = 1./255
- Batch size: 32
- Used same image loader method as Model 1

5.2 Transfer Learning Architecture

text

CopyEdit

- MobileNetV2 (include_top=False, weights='imagenet', input_shape=(224, 224, 3))

- GlobalAveragePooling2D

- Dropout(0.2)

- Dense(256, activation='relu') + BatchNormalization + Dropout(0.1)

- Dense(128, activation='relu') + BatchNormalization + Dropout(0.1)

- Dense(9, activation='softmax')

5.3 Training & Evaluation

- Optimizer: Adam
- Loss: Categorical Crossentropy
- Metrics: Accuracy, Precision, Recall
- Epochs: 50 with EarlyStopping (monitoring validation loss)

5.4 Performance (MobileNetV2)

- **Test Accuracy**: 0.88

- **Test Loss**: 0.48
- **Precision**: 0.92
- **Recall**: 0.83
- **Observation**: The model generalized well, showed strong classification ability, and is lightweight—ideal for mobile deployment.

---

6. Model Comparison

Metric	CNN Model	MobileNetV2
Accuracy	0.34	0.88
Loss	1.67	0.48
Precision	0.53	0.92
Recall	0.12	0.83

Visualization: A bar chart comparing Accuracy, Loss, Precision, and Recall for both models highlights the clear superiority of MobileNetV2 in all aspects.

---

7. Conclusion

- **CNN model** trained from scratch showed limited performance due to lack of depth and capacity.
- **MobileNetV2 with transfer learning** leveraged pretrained features effectively, offering excellent performance with low latency—ideal for real-time port operations on mobile devices.
- This project demonstrates the **practical benefits of transfer learning**, especially when data is limited and deployment constraints (like mobile usage) are in place.