Hydration Essentials: Classifying Water Bottle Images

Team ID	SWTID1720163161
Project Title	Hydration Essentials: Classifying Water Bottle Images

Final Report

1. Introduction

1.1 Project Overview

This project employs computer vision and machine learning bto classify images of water bottles into three categories: half, full, and overflowing. The primary goal is to automate the categorization process for applications in e-commerce, inventory management, and quality control. By leveraging a trained model, the project can quickly and accurately classify new images, providing significant value to manufacturers, retailers, and consumers.

1.2 Objectives

- Streamline the process of categorizing water bottles.
- Improve the accuracy and speed of water bottle classification.
- Provide valuable insights for manufacturers, retailers, and consumers to optimize their processes and product offerings.

---

2. Project Initialization and Planning Phase

2.1 Define Problem Statement

The core problem addressed by this project is the need for efficient and accurate classification of water bottles based on their water content. Traditional manual methods are time-consuming and prone to errors, necessitating an automated solution that can handle diverse bottle images and accurately classify them into predefined categories.

Link: (1.1)

2.2 Project Proposal (Proposed Solution)
The proposed solution involves developing a machine learning model trained on a dataset of water bottle images with varying levels of water content. This model will be capable of identifying key features associated with each water content category and classifying new images accordingly. The project will utilize convolutional neural networks (CNNs) due to their efficacy in image recognition tasks.

Link: (1.3)

2.3 Initial Project Planning
The initial planning phase includes:

- Collecting a diverse dataset of water bottle images to ensure the model can generalize well.
- Preprocessing the images for uniformity, including resizing and normalization.
- Selecting and training an appropriate machine learning model.
- Evaluating the model's performance and making necessary optimizations.

Link: (1.2)

---

3. Data Collection and Preprocessing Phase

3.1 Data Collection Plan and Raw Data Sources Identified
The dataset comprises images of water bottles categorized into three classes: 'Full Water level', 'Half water level', and 'Overflowing'. A total of 486 images were collected from Kaggle, ensuring a diverse representation of each category.

Link: https://www.kaggle.com/datasets/chethuhn/water-bottle-dataset

Link: (2.1)

3.2 Data Quality Report
The dataset includes 3 classes with the following details:

- Number of Classes: 3
- Class Names: ['Full Water level', 'Half water level', 'Overflowing']
- Training Examples: 12
- Validation Examples: 2

- Test Examples: 2

Link: (2.2)

3.3 Data Preprocessing

Preprocessing steps involve resizing images to a uniform size of 256x256 pixels and normalizing pixel values to fall within the range of 0 to 1. This ensures consistency and enhances the model's ability to learn from the images.

Link: (2.3)

---

4. Model Development Phase

4.1 Model Selection Report

A Convolutional Neural Network (CNN) was selected for this project due to its proven effectiveness in image classification tasks. The model was implemented using TensorFlow and Keras, and consists of 16 layers, including convolutional, pooling, and dense layers, designed to extract and learn intricate features from the images.

Link: (3.2)

4.2 Initial Model Training Code, Model Validation, and Evaluation Report

The model was compiled with the Adam optimizer and sparse categorical cross-entropy loss function. It was then trained on the dataset for 10 epochs, with validation at each epoch to monitor performance and avoid overfitting.

Link: (3.1)

---

5. Model Optimization and Tuning Phase

5.1 Tuning Documentation

Hyperparameter tuning involved experimenting with different learning rates, batch sizes, and epoch numbers to optimize the model's performance. Techniques such as early stopping and learning rate scheduling were employed to enhance training efficiency and model accuracy.

Link: (4.1)

5.2 Final Model Selection Justification

The final model was selected based on its performance on the validation dataset.

The selection criteria included metrics such as accuracy, precision, recall, and F1-score. The chosen model demonstrated the best balance between these metrics, ensuring robust performance across different water bottle images.

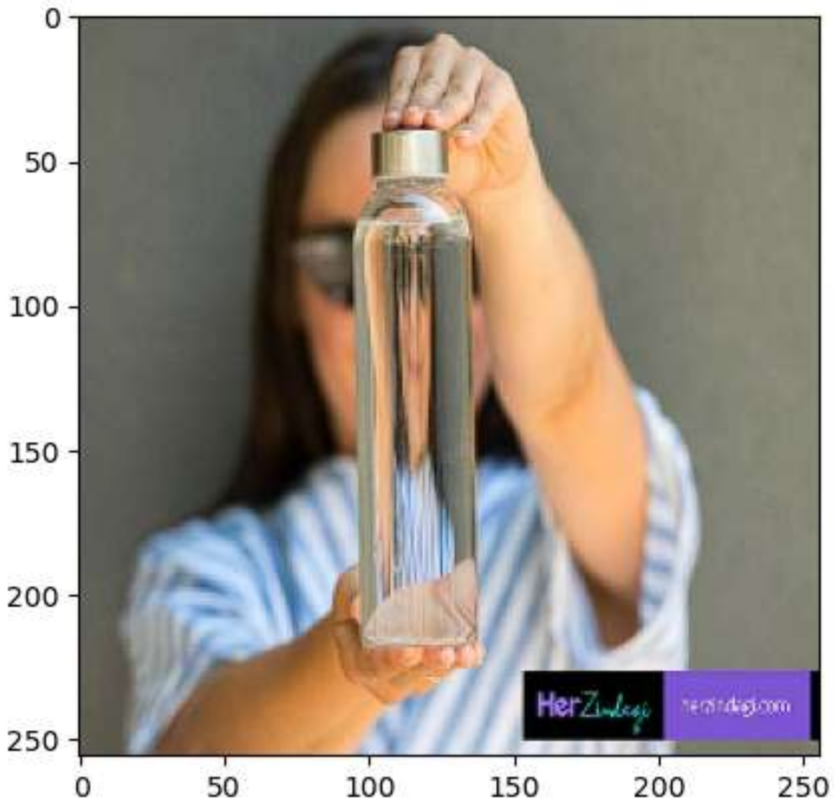
Link: (4.1)

---

6. Results

6.1 Output Screenshots

```
first image to predict
actual label: Full  Water level
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 151ms/step
predicted label: Full  Water level
```



Actual: Full Water level,
Predicted: Full Water level.
Confidence: 88.92%

Actual: Full Water level,
Predicted: Full Water level.
Confidence: 97.58%

Actual: Full Water level,
Predicted: Full Water level.
Confidence: 84.75%

Actual: Full Water level,
Predicted: Half water level.
Confidence: 68.71%

---

7. Advantages & Disadvantages

Advantages

- Automates the classification process, saving time and reducing manual effort.
- Improves accuracy and consistency in categorizing water bottle images.
- Enhances inventory management and quality control for manufacturers and retailers.

Disadvantages

- The model's performance may be affected by variations in image quality and bottle designs.
- Requires a significant amount of labeled data for training and validation.

---

8. Conclusion

The project successfully achieved its objectives by developing a CNN model that accurately classifies water bottle images based on their water content. The automation of this classification process has potential applications in e-commerce, inventory management, and quality control, offering significant benefits to various stakeholders.

---

9. Future Scope

Future enhancements could include expanding the dataset to cover a wider variety of water bottles, improving model accuracy through advanced techniques, and deploying the model in real-world applications to further validate its performance and utility.

---

10. Appendix

10.1 Source Code

The complete source code used for the project, including data preprocessing, model development, training, and evaluation, is available in the project repository.

10.2 GitHub & Project Demo Link

Link to Github:

Link to Demo Video: