

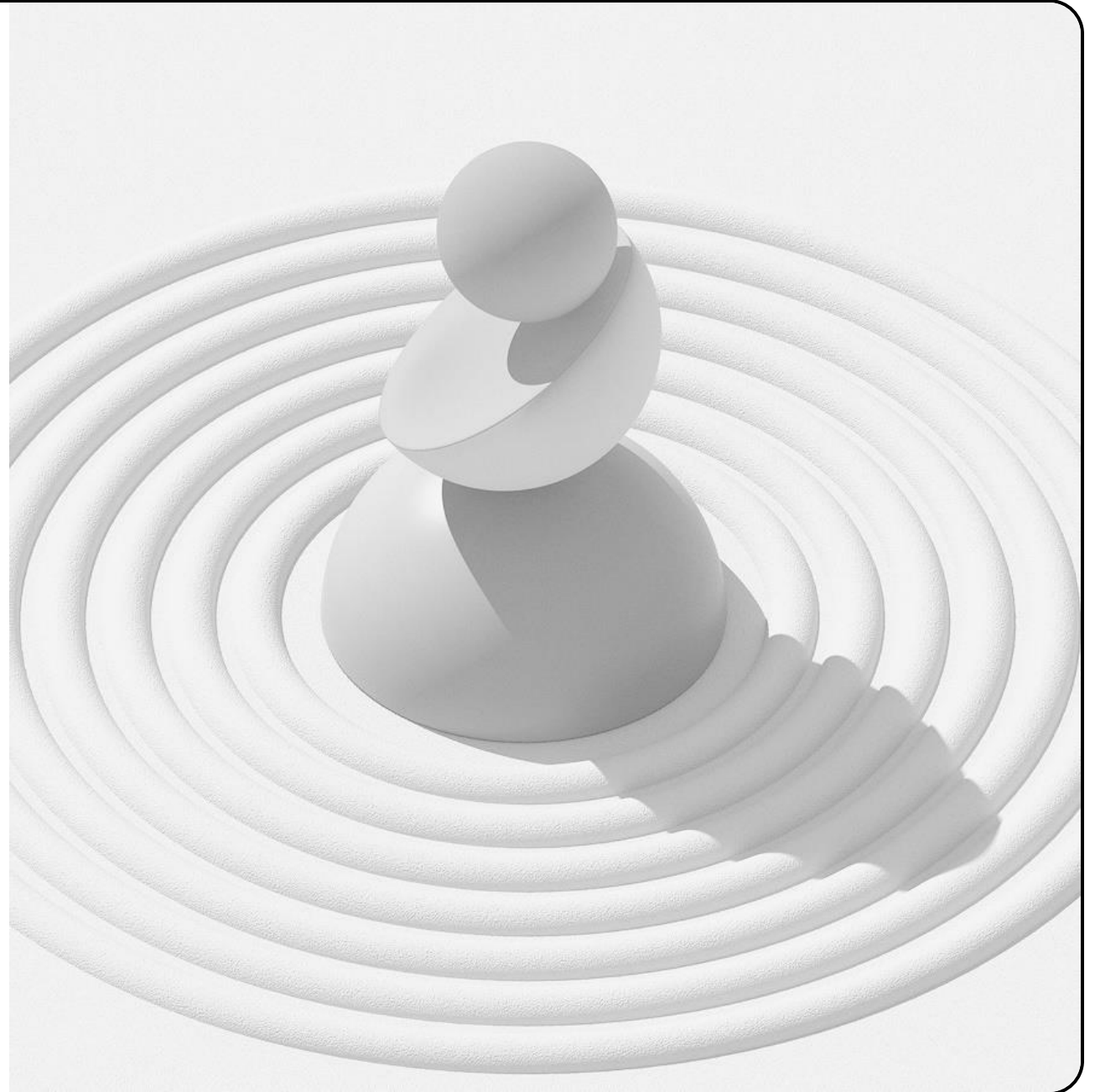
Presented by Alikhan Smagulov

FRUIT QUALITY DETECTION

INTRODUCTION

The agricultural sector faces an ever-increasing demand for stringent quality inspection of produce. Traditionally, this process relies on manual labour, which is both time-consuming and inherently subjective. Our project aims to revolutionise this by automating fruit quality detection using advanced computer vision techniques.

The core objective is to develop an application capable of classifying fruits as either fresh or rotten directly from images, thereby enhancing efficiency and consistency in quality control.



MODEL & DATASET

Dataset: Kaggle "Fruits Fresh and Rotten"

- Comprises a diverse collection of images for various fruit types.
- Specifically includes apples, bananas, and oranges.
- Each image is meticulously labelled as either 'fresh' or 'rotten', providing clear categorisation for training.

Model: Transfer Learning with MobileNetV2

- Leverages MobileNetV2, a highly efficient convolutional neural network.
- Initially pre-trained on the vast ImageNet dataset, allowing it to recognise a wide array of features.
- Fine-tuned specifically for our application to excel in fruit classification tasks.

Software: Tools and libraries

My project's robust implementation is built upon a carefully selected stack of powerful technologies, ensuring both efficiency and a user-friendly experience.



Python 3.10+

The primary programming language, chosen for its versatility and extensive libraries.



Tensorflow & Keras

Essential for constructing, training, and optimising our deep learning model.



Pillow

Utilised for efficient image manipulation and processing operations.



NumPy

Provides fundamental support for numerical computation, crucial for data handling.



Tkinter&ttk

Enables the creation of a modern, intuitive desktop Graphical User Interface.



PyInstaller

An optional tool used for packaging the application into a standalone executable.

METHODOLOGY: FROM DATA TO PREDICTION

Data Preparation

- Organising the dataset into distinct training and validation sets.
- Implementing data augmentation techniques (e.g., rotation, zoom, flips) to enhance model generalisation.

Model Training

- Applying transfer learning with the MobileNetV2 architecture.
- Incorporating early stopping criteria to prevent overfitting and ensure optimal performance.
- The trained model is then securely saved as an .h5 file for later deployment.

Prediction & GUI

- Loading the trained model directly into the Graphical User Interface.
- Allowing users to select or drag-and-drop images for analysis.
- Displaying the top-3 predictions along with their confidence percentages for clarity.

CORE CAPABILITIES: WHAT OUR SYSTEM OFFERS

Multi-Fruit Detection

Capable of accurately identifying and classifying various fruit types, including apples, bananas, and oranges.

Top-3 Predictions with Confidence

Provides not just one, but the top three most probable predictions, each accompanied by a confidence percentage, offering comprehensive insights.

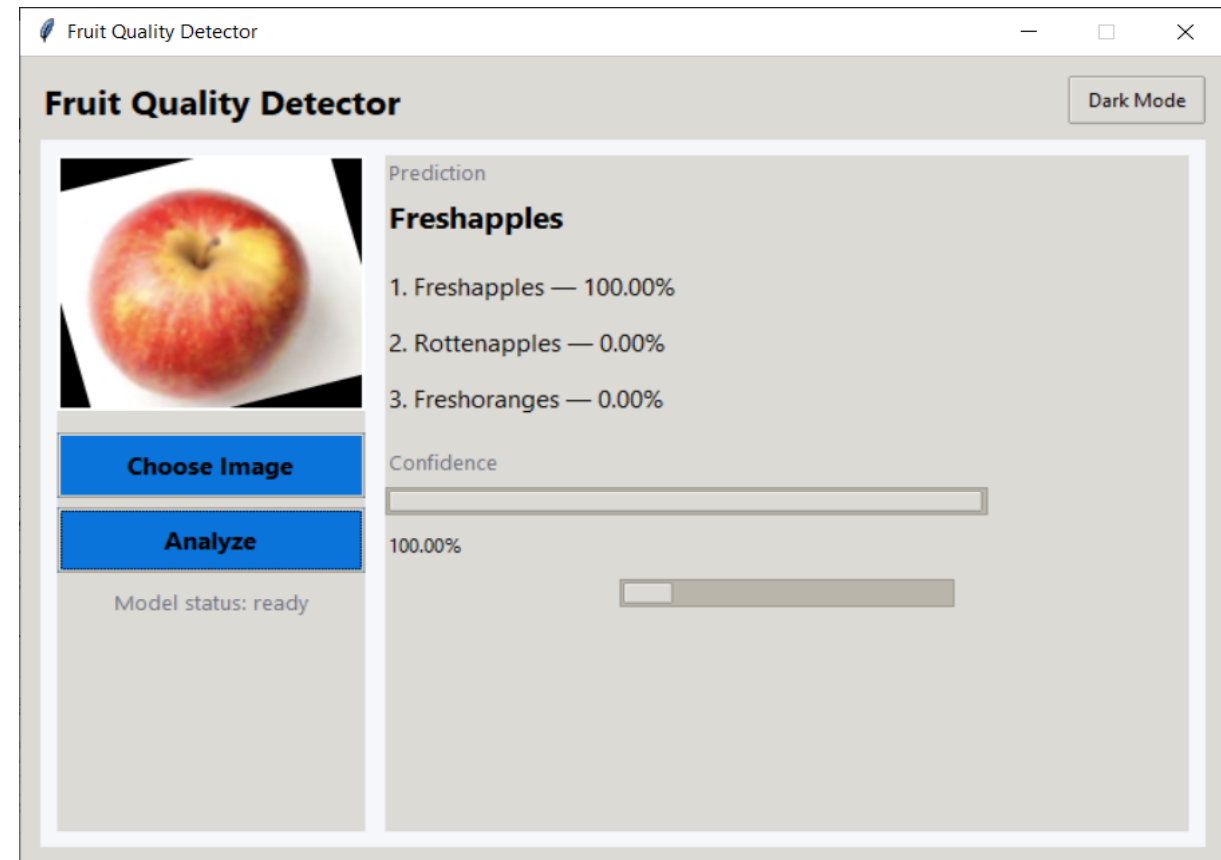
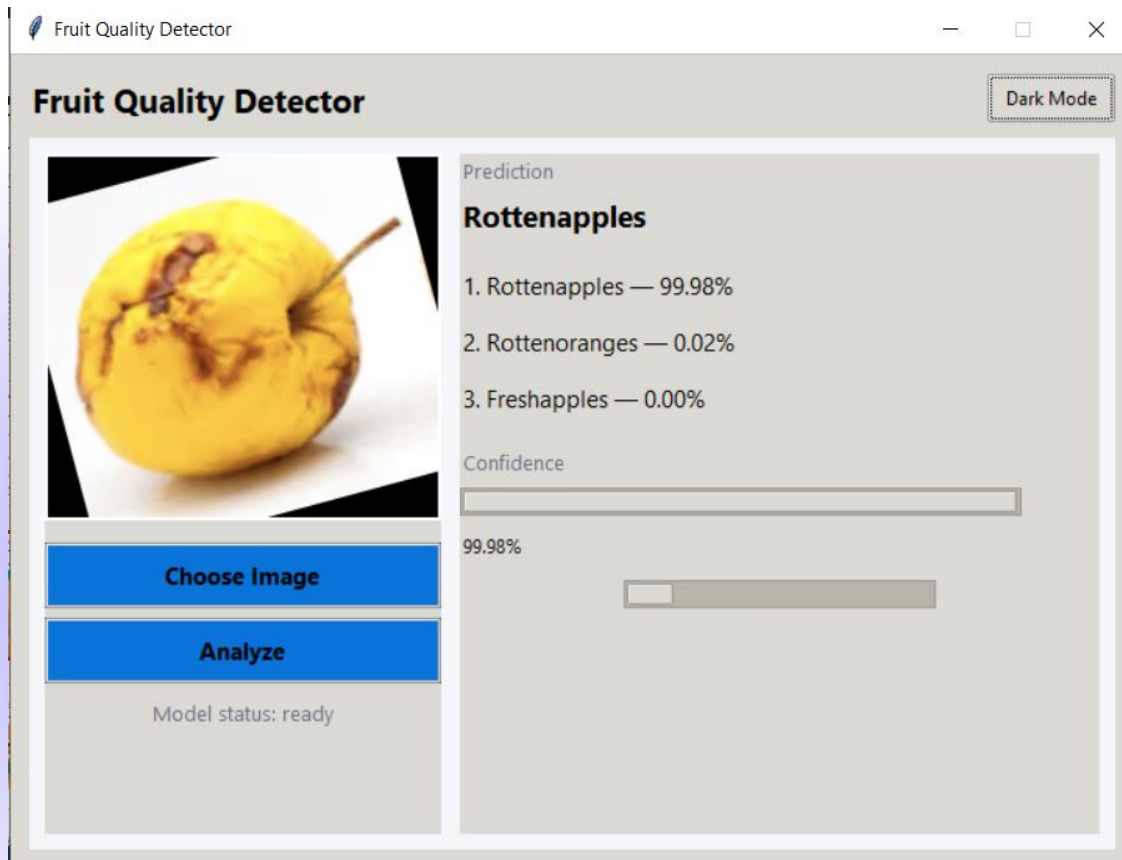
Modern GUI with Image Preview

Features an intuitive Graphical User Interface that allows for easy image selection and a clear preview before prediction.



SHOWCASE: PREDICTING FRUIT QUALITY

Witness my application in action, demonstrating its seamless functionality and accurate prediction capabilities.





CONCLUSION:

MobileNetV2 Effectiveness

Demonstrated the efficiency of MobileNetV2 with transfer learning for accurate image classification in this domain.

Automated Quality Detection Achieved

I have successfully developed an automated system for fruit quality detection, addressing the limitations of manual inspection.

User-Friendly Tkinter GUI

Tkinter-based Graphical User Interface ensures an intuitive and accessible experience for users.

FUTURE HORIZONS: ENHANCEMENTS AND INTEGRATIONS

While current system is robust, several avenues exist for further improvement and expansion, pushing the boundaries of automated fruit quality inspection.

→ Expand Fruit Type Database

Incorporating a wider variety of fruits into the dataset to broaden the application's utility and versatility.

→ Real-Time Camera Integration

Developing functionality to allow for real-time detection and classification directly from a live camera feed, enabling immediate feedback.

→ Continuous Model Refinement

Improving model accuracy by utilising larger, more diverse datasets and exploring advanced deep learning architectures.

REFERENCES

Title & Authors

Computer Vision-Based Agriculture Engineering —
Han Zhongzhi (2020)

Computer Vision and Recognition Systems Using
Machine and Deep Learning Approaches:
Fundamentals, Technologies and Applications —
Chowdhary, Alazab, Chaudhary (eds., 2021)

W3Schools

<https://www.w3schools.com>

Kaggle

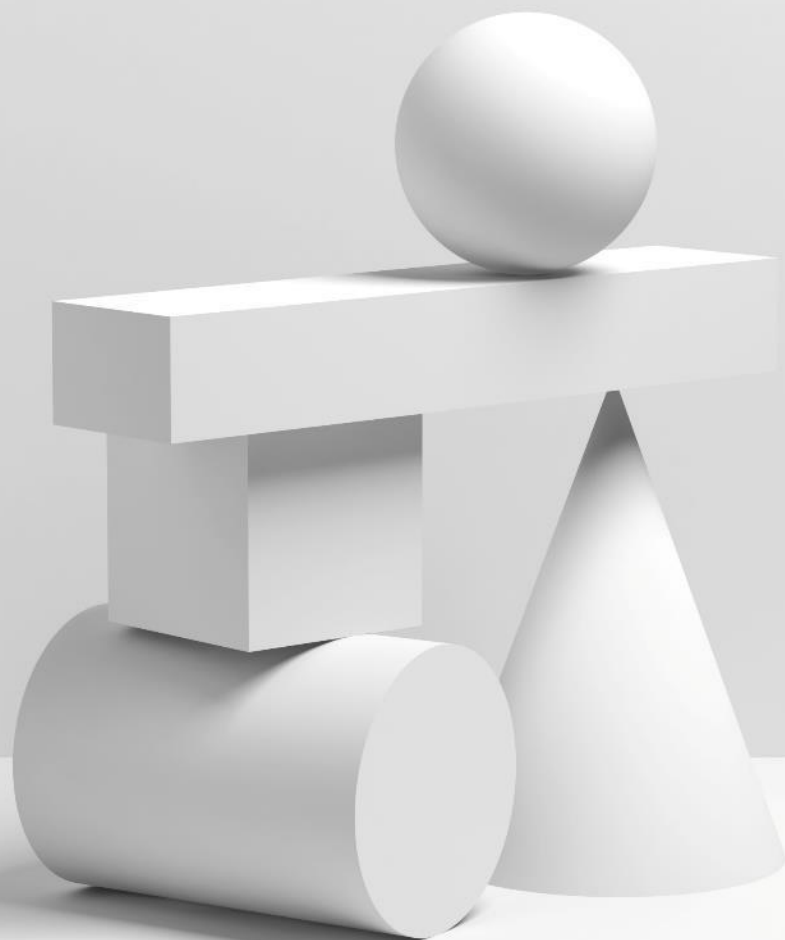
<https://www.kaggle.com>

StackOverflow

<https://stackoverflow.com>

TensorFlow Official Documentation

<https://www.tensorflow.org>



THANK YOU
FOR
ATTENTION