CORNputer Vision Repository 🌽

Welcome to the **CORNputer Vision** repository! This project forms the backbone of my master's thesis, where I developed a camera- and computer vision-powered system for maize seed analysis.

The Python Version used during the development of this repository is **Python 3.11**.

🚀 Ubuntu Installation Guide

Follow these steps to set up the repository on **Ubuntu 20.04**.

🔧 Realsense Library

To capture images using a Realsense camera, install the Realsense library following the Realsense Installation Guide.

📦 Install Pre-requisites

Run the following commands in the root directory of this repository to install necessary pre-requisites:

```
sudo apt-get update
sudo apt-get -y upgrade
sudo apt-get install -y python3-pip
sudo apt-get install build-essential libssl-dev libffi-dev python-dev
sudo apt-get install -y python3-venv
```

🐍 Python Virtual Environment Setup

Execute the following commands in the root directory of this repository to set up the Python virtual environment:

```
python3 -m venv ".venv"
source .venv/bin/activate
pip install --upgrade pip
```

To deactivate the environment, simply run:

```
deactivate
```

📥 Install required python packages

Execute the following command in the root directory of this repository to install the required Python packages:

```
pip install -r requirements_ubuntu.txt
```

🚀 Windows Installation Guide

Follow these steps to set up the repository on **Windows 10/11**.

🔧 Realsense Library

To use a Realsense camera on Windows, install the **Intel RealSense SDK** from the official site:
👉 Intel Realsense SDK Releases

1. Download the latest `.exe` installer under **Assets**
2. Install the SDK and plug in your Realsense camera

🐍 Python Virtual Environment Setup

Open PowerShell or Command Prompt in the root directory of the repository and run:

```
python -m venv .venv
.\.venv\Scripts\activate
python -m pip install --upgrade pip
```

To deactivate the environment:

```
deactivate
```

📥 Install required Python packages

```
pip install torch==2.6.0+cpu torchvision==0.21.0+cpu --index-url https://download.pytorch.
pip install -r requirements_windows.txt
```

Please note: in Windows CUDA is not going to be installed, i.e. GPU training / inference is not possible.

🛠️ Usage

▶ Main Entry Point: `whatrun.py`

Run the main launcher:

```
python whatrun.py
```

You will be prompted to choose:

```
1. Data Capture
2. Training
3. Prediction
```

---

1. 📷 Data Capture (RealSense)

You will be asked to:

- Provide the path to the camera config file
- Choose one of the following modes:
    - `1` – Display camera stream (RGB, Depth, Background Removed)

    - `2` – Save frame on key press
    - `3` – Continuously capture all frames

> Example config: `data_capture/configs/example.yaml`

All images and camera settings are saved automatically to the defined path.

If you want to change Camera Settings manually, do so in the camera configuration.

> Example camera configuration `data_capture/configs/example.json`

---

2. 🏋♂ Model Training

You will be prompted for:

- Path to the training config
- Whether to train and/or validate

```
python whatrun.py
→ 2
→ Enter config path
→ Train? (y/n)
→ Validate? (y/n)
```

> Example training configuration `model_development/param_configs/example.yaml`

---

3. 🧪 Prediction & Post-Processing

You will be prompted for:

- Config file
- Image input folder
- Output folder name
- Task to execute:
    - `1` – Tracking

- `2` – Classification (broken/intact)
- `3` – Classification (tip/no tip)
- `4` – Size estimation
- `5` – Run all steps

🔁 Full Non-Interactive Example (Prediction)

```
python predict/run.py \
  --config predict/configs/example.yaml \
  --data datasets/.../... \
  --name test_run \
  --whatrun 5
```

This command performs tracking, both classification tasks, and size estimation in one go.