

Technische Hochschule Deggendorf
Faculty Applied Information Technology
Studiengang Bachelor of Cyber-Security

Projekt "WeedDetector"

Projektübergabe

Vorgelegt von:

Christof Renner (22301943)
Manuel Friedl (1236626)

Prüfungsleitung:

Prof. Dr. Holger Jehle

Datum: 05. Juli 2025

Contents

1	Einführung und Projektübersicht	3
1.1	Motivation und Zielsetzung	3
1.2	Technologiestack	3
2	Installation	4
2.1	Installation mittels Docker (macOS und Linux)	4
2.2	Manuelles Setup unter Windows	4
3	Benutzerhandbuch	5
3.1	Benutzeroberfläche	5
3.1.1	Hauptfunktionen	6
3.2	Grundlegende Bedienung	6
3.2.1	Einzelbildanalyse	6
3.2.2	Live-Kameraerkennung	6
3.2.3	Robotersteuerung	7
3.3	Erweiterte Funktionen	7
3.3.1	Anpassung der Erkennungsparameter	7
3.3.2	Interpretation der Erkennungsergebnisse	7

1 Einführung und Projektübersicht

1.1 Motivation und Zielsetzung

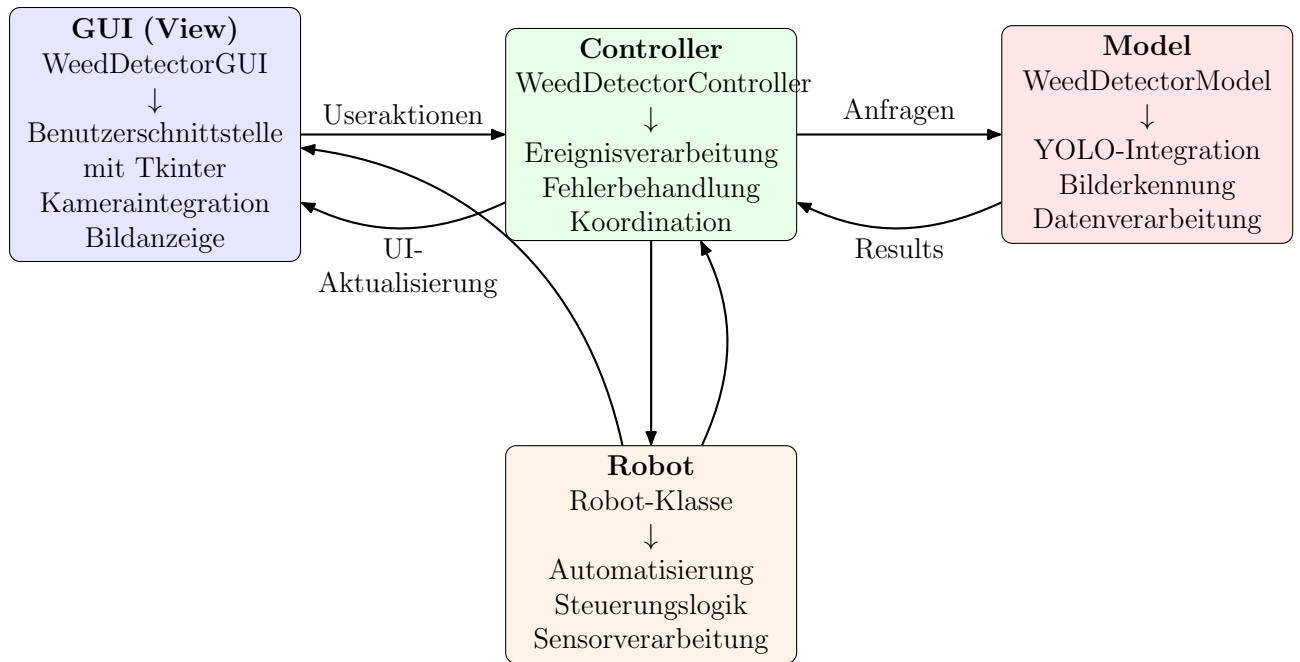


Figure 1.1: Detaillierte Architektur des WeedDetector-Systems

1.2 Technologiestack

Der WeedDetector nutzt einen modernen Technologiestack, der für Bildverarbeitung und maschinelles Lernen optimiert ist:

- **Programmiersprache:** Python 3.13 als Hauptsprache für die gesamte Anwendung
- **Computer Vision:**
 - OpenCV für Bildverarbeitung und Kameraintegration
 - Ultralytics YOLO für Objekterkennung und -klassifikation
- **Benutzeroberfläche:**
 - Tkinter für die grafische Benutzeroberfläche
- **Containerisierung:** Docker für einheitliche Umgebungen

2 Installation

Die Installation des WeedDetector-Systems erfolgt primär über Docker, um eine konsistente Umgebung zu gewährleisten und Abhängigkeitsprobleme zu vermeiden. Die Nutzung unter Linux und macOS ist dabei am besten unterstützt. Für Windows-Nutzer wird das direkte Ausführen der Anwendung empfohlen, da Windows Einschränkungen bei der Kameranutzung und dem Darstellen der GUI haben kann.

2.1 Installation mittels Docker (macOS und Linux)

1. **Docker installieren:** Falls noch nicht geschehen, installieren Sie Docker gemäß der offiziellen Anleitung für Ihr Betriebssystem (<https://docs.docker.com/get-docker/>).

2. **Repository klonen:**

```
1 git clone https://mygit.th-deg.de/mf13626/ss25_weed_detector
2 cd ss25_weed_detector
3
4 # oder alternativ:
5 docker pull crnr/weeddetector:latest
```

3. **Docker-Image bauen:**

```
1 xhost +local:root # Erlaubt Docker Zugriff auf X11
2 docker build -t weed-detector .
```

4. **Anwendung starten:**

```
1 docker run --device /dev/video0:/dev/video0 --net=host -e
  DISPLAY=$DISPLAY -v /tmp/.X11-unix:/tmp/.X11-unix weed-
  detector
```

2.2 Manuelles Setup unter Windows

Unter Windows muss die Anwendung direkt ohne Docker ausgeführt werden.

1. **Python-Umgebung einrichten:**

```
1 git clone https://mygit.th-deg.de/mf13626/ss25_weed_detector
2 cd ss25_weed_detector
3 python -m venv venv
4 venv\Scripts\activate
```

2. **Abhängigkeiten installieren:**

```
1 pip install -r requirements.txt
```

3. **Anwendung starten:**

```
1 python app/main.py
```

3 Benutzerhandbuch

3.1 Benutzeroberfläche

Die Benutzeroberfläche des WeedDetector-Systems ist in mehrere funktionale Bereiche unterteilt:

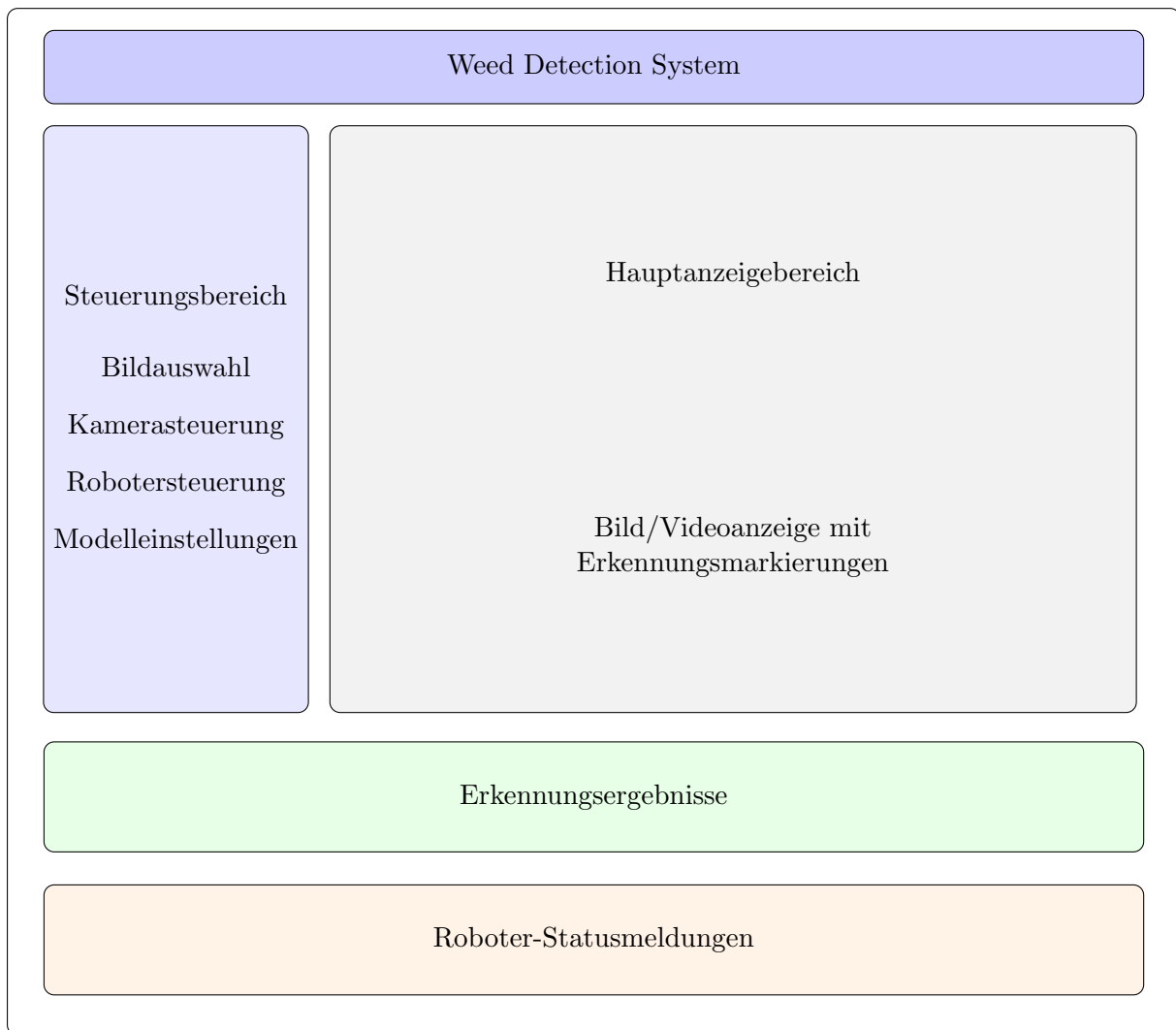


Figure 3.1: Benutzeroberfläche des WeedDetector-Systems

3.1.1 Hauptfunktionen

Die Benutzeroberfläche bietet folgende Hauptfunktionen:

1. **Bildauswahl:** Ermöglicht das Laden eines Bildes aus dem Dateisystem zur Analyse.
2. **Kamerasteuerung:** Aktiviert die Echtzeit-Bilderfassung über die angeschlossene Kamera.
3. **Robotersteuerung:** Startet und stoppt die automatisierte Unkrauterkennung und -behandlung.
4. **Empfindlichkeitseinstellung:** Ermöglicht die Anpassung des Konfidenzwertes für die Erkennung (0,05-0,95).
5. **Erkennungsanzeige:** Zeigt erkannte Unkräuter mit Begrenzungsrahmen, Mittelpunktmarkierungen und Beschriftungen an.
6. **Ergebnisprotokoll:** Listet detaillierte Informationen zu erkannten Objekten, einschließlich Koordinaten und Konfidenzwerten.
7. **Roboter-Aktionsprotokoll:** Dokumentiert alle Aktionen und Statusmeldungen des Robotersystems.

3.2 Grundlegende Bedienung

3.2.1 Einzelbildanalyse

Um ein einzelnes Bild auf Unkraut zu analysieren:

1. Klicken Sie auf die Schaltfläche **”Select Image”**.
2. Wählen Sie eine Bilddatei aus dem Dateiauswahldialog.
3. Das System führt automatisch eine Erkennung durch und zeigt die Ergebnisse an.
4. Passen Sie bei Bedarf den Konfidenzwert an, um die Erkennungsempfindlichkeit zu ändern.

3.2.2 Live-Kameraerkennung

Für die Echtzeit-Unkrauterkennung mit der Kamera:

1. Klicken Sie auf die Schaltfläche **”Start Camera”**.
2. Die Kameraansicht wird im Hauptanzeigebereich angezeigt.
3. Erkannte Unkräuter werden in Echtzeit markiert und im Ergebnisbereich aufgelistet.
4. Zum Beenden der Kameraerkennung klicken Sie auf **”Stop Camera”**.

3.2.3 Robotersteuerung

Zur Aktivierung des automatisierten Robotermodus:

1. Stellen Sie sicher, dass die Kamera aktiviert ist oder aktivieren Sie sie.
2. Klicken Sie auf die Schaltfläche **”Start Robot”**.
3. Der Roboter beginnt mit der automatischen Unkrauterkenkung und -behandlung.
4. Alle Roboterhandlungen werden im Roboter-Statusbereich protokolliert.
5. Bei Erkennung von Unkraut stoppt der Roboter automatisch für eine genauere Analyse.
6. Zum manuellen Stoppen des Roboters klicken Sie auf **”Stop Robot”**.

3.3 Erweiterte Funktionen

3.3.1 Anpassung der Erkennungsparameter

Die Erkennungsgenauigkeit kann durch Anpassung des Konfidenzwertes optimiert werden:

- **Niedriger Konfidenzwert (0,05-0,25):** Erkennt mehr potenzielle Unkräuter, kann aber zu falschen Erkennungen führen.
- **Mittlerer Konfidenzwert (0,25-0,50):** Bietet eine ausgewogene Erkennungsrate für die meisten Anwendungen.
- **Hoher Konfidenzwert (0,50-0,95):** Reduziert Falscherkennungen, kann aber auch tatsächliche Unkräuter übersehen.

3.3.2 Interpretation der Erkennungsergebnisse

Die Erkennungsergebnisse werden in folgenden Formaten dargestellt:

- **Visuelle Markierungen:**
 - Rote Begrenzungsrahmen um erkannte Unkräuter
 - Grüne Kreuze an den Mittelpunkten der erkannten Objekte
 - Textbeschriftungen mit Klasse, Koordinaten und Konfidenzwert
- **Textbasierte Ausgabe:**
 - Nummerierte Liste erkannter Objekte
 - Klassenname jedes erkannten Objekts
 - X/Y-Koordinaten des Mittelpunkts
 - Konfidenzwert der Erkennung