**Vision and Scope Document**

**for**

**Specification of the PlantVision software system**

**Version 1.0**

**approved**

**Prepared by**

**Koshilem D.V.**

**NURE**

## 19.06.2025

## 1. Постановка мети.

Метою розробки є створення експериментальної системи PlantVision, що реалізує автоматизоване виявлення, класифікацію та трекінг листя й захворювань рослин у відеопотоці з використанням моделей YOLOv8 та YOLOv11. Завданнями системи є:

* інтеграція двох дата-сетів (PlantVillage і PlantDoc) у стандартизований формат YOLO;
* налаштування й запуск пайплайнів навчання для YOLOv8 і YOLOv11;
* обробка відеопотоку, детекція об’єктів, збирання метрик (FPS, час детекції, кількість кадрів із об’єктами, тривалість перебування в кадрі);
* порівняльний аналіз точності й продуктивності моделей.

## 2. Загальний опис системи.

PlantVision побудовано за клієнт-серверною архітектурою:

Клієнтська частина (PyTorch + OpenCV) забезпечує захоплення та відображення відео, відправлення кадрів на сервер детекції, візуалізацію результатів (bounding boxes, сегментація, мітки).

Серверна частина (Python + Ultralytics YOLO API) виконує інференс на моделях YOLO, формує метрики й повертає їх клієнту.

Субсистема збору метрик фіксує час початку й завершення детекції кожного об’єкта для розрахунку середнього часу детекції та тривалості трекінгу.

Система призначена для дослідницьких експериментів із порівняння продуктивності моделей у контрольованих (PlantVillage) та реальних (PlantDoc) умовах.

## 3. Основний функціонал системи

У рамках підготовчого етапу було імпортовано анотації PlantVillage і PlantDoc та конвертовано їх у формат Darknet-TXT, після чого дані були розбиті на навчальний, валідаційний і тестовий набори у співвідношенні 70 %/15 %/15 %. На наступному кроці виконано навчання двох архітектур — YOLOv8 та YOLOv11 — з підбором гіперпараметрів та застосуванням аугментацій, одночасно відстежуючи основні метрики якості (mAP), динаміку функцій втрат і показники FPS під час тренування.

Після завершення тренування реалізовано модуль інференсу, що забезпечує стрімову подачу відеопотоку (з дрону або з файлу), реальнозорієнтовану детекцію та сегментацію об’єктів із накладенням bounding-boxes і сегментних масок на кадр. Для подальшого аналізу побудовано підсистему збору метрик, яка рахує загальну кількість кадрів і кадрів із виявленими об’єктами, обчислює середній час детекції на кадр і трекінгу одного об’єкта, а також відсоток часу, витраченого на інференс від загальної тривалості обробки.

Нарешті, результати дослідження були автоматично зведені у табличний звіт, а також представлені на побудованих графіках, що дозволило безпосередньо порівняти продуктивність моделей YOLOv8 та YOLOv11 на обох датасетах.

## 4. Загальні обмеження

## Система вимоглива до ресурсів GPU: для реального часу рекомендовано мінімум 4 ГБ відеопам’яті.

## Підтримуються лише формати анотацій Darknet-TXT.

## Вхідне відео має бути до 1920×1080 пікселів; більші розміри призводять до зниження FPS.

## Інтеграція з іншими фреймворками (TensorFlow, Detectron2) не передбачена.

## 5. Припущення та залежності

Для коректного запуску системи необхідно мати встановлене середовище Python ≥ 3.8 із бібліотеками PyTorch, OpenCV та Ultralytics, а також бажано доступ до GPU з драйвером CUDA (підтримка роботи на CPU передбачена, але з помітним зниженням продуктивності). Датасети PlantVillage і PlantDoc мають бути локально розпаковані за структурою.

А чекпоінти моделей у форматі .pt розміщуються у папці weights/ і позначаються шаблоном yolov8\_\*.pt або yolov11\_\*.pt. Система підтримує запуск як на Windows, так і на Linux під управлінням інтерпретатора Python.