**CropVision Desktop App Installation & User Manual (Linux/HPC)**

**1. Project Identified**

**1.1 Purpose**

Evaluate and use three complementary vision models (YOLOv11, Xception-TFLite, ResNet-50/Keras) to classify crop type and plant health state from leaf photos. Runs offline with optional GPU acceleration (A100/L40S).

**1.2 Repository:** <https://github.com/DataWithAnish/Plant_detection_3-model>

**2. System Requirements (Linux/HPC)**

**2.1 Supported Platform**

* **OS:** Linux (cluster node or workstation)

**2.2 Recommended Hardware**

* **GPU:** NVIDIA A100 40GB (SM\_80) or L40S 40GB (SM\_89) (1× GPU is sufficient)
* **Driver:** ≥ 535 (validated on 580.82.07)
* **Disk:** 10–20 GB free (models + caches)

**2.3 Software**

* **Python:** 3.10 (Conda environment recommended)
* **TensorFlow:** 2.16+ (bundled CUDA 12.x + cuDNN 9.x; no separate Keras install)
* **PyTorch:** 2.3.0 + torchvision 0.18.0 (cu121 wheels)
* **Other:** numpy, pillow, matplotlib 3.8–3.10, opencv-python 4.11.0.86, pandas, seaborn, ultralytics, h5py

### 2.4 Shared-GPU Note

If another job holds most VRAM, TensorFlow/cuDNN may fail to initialize. Use a free GPU or set:

|  |
| --- |
| export TF\_FORCE\_GPU\_ALLOW\_GROWTH=true |

## 3. Repository Layout

|  |
| --- |
| COSC591-Plant/ |
| ├─ COSC591-plant-classifier.py # main Tkinter app |
| ├─ setup\_and\_run.sh # optional one-click bootstrap (Linux) |
| ├─ requirements.lock.txt # pinned deps |
| └─ models/ |
| ├─ yolo/ |
| │ └─ best.pt |
| ├─ xception/ |
| │ ├─ xception\_pv1.tflite |
| │ ├─ labels\_crop.txt |
| │ └─ labels\_state.txt |
| └─ resnet50/ |
| ├─ resnet50\_stage2\_conv5.h5 # this is not included in github |
| ├─ class\_index.csv # preferred |
| └─ class\_names.json # fallback |

**Required assets:** Xception/ResNet files must exist and be **> 1 KB** or those modules will not start.  
**Path note:** Some repos use inference\_code/models/... — same files apply.

**Source repo:** <https://github.com/DataWithAnish/Plant_detection_3-model>

## 4. Installation (Linux)

### 4.1 Manual Install (no script)

**(a) Get a GPU shell (example; adapt to your cluster)**

|  |
| --- |
| salloc --partition=moore --gres=gpu:1 --time=02:00:00 --pty bash  hostname && nvidia-smi |

**(b) Create & activate Conda env**

|  |
| --- |
| source ~/miniconda3/etc/profile.d/conda.sh 2>/dev/null || true  conda create -y -n COSC591-Plant python=3.10  conda activate COSC591-Plant  python -m pip install -U pip |

**(c) Install packages**

|  |
| --- |
| # TensorFlow (bundled CUDA 12 + cuDNN 9)  pip install "tensorflow[and-cuda]==2.16.\*"  # PyTorch (CUDA 12.1 wheels)  pip install --index-url https://download.pytorch.org/whl/cu121 \  "torch==2.3.0" "torchvision==0.18.0"  # Core + tools  pip install "numpy>=1.26,<2.2" pillow "matplotlib>=3.8,<3.11" \  opencv-python==4.11.0.86 pandas seaborn ultralytics h5py |

**Do not** install a separate keras on Linux use tf.keras from TF 2.16+.

**(d) Place model files**

**(e) Launch**

|  |
| --- |
| export TF\_FORCE\_GPU\_ALLOW\_GROWTH=true # recommended on shared GPUs  # export CUDA\_VISIBLE\_DEVICES=0 # pin to a specific GPU if needed  python COSC591-plant-classifier.py |

### 4.2 Scripted Install (optional)

|  |
| --- |
| chmod +x setup\_and\_run.sh  ./setup\_and\_run.sh  # later runs:  conda activate COSC591-Plant  python COSC591-plant-classifier.py |

### 4.3 Uninstallation

|  |
| --- |
| conda remove --name COSC591-Plant --all |

## 5. Packaging & Large Files (GitHub & Submission)

**GitHub:** <https://github.com/DataWithAnish/Plant_detection_3-model>  
(Models >100 MB are excluded from Git; they’re included in the submission ZIP.)

### 5.1 GitHub Size Limit

GitHub enforces **a 100 MB** per-file limit. Example exceeding file:  
inference\_code/models/resnet50/resnet50\_stage2\_conv5.h5 (~136.8 MB).

### 5.2 Submission Policy

The **assessment ZIP submission will include all model files** (.h5, .tflite, label files).  
The **public GitHub repo will exclude** >100 MB artifacts.

## 6. User Manual (Usage Guide)

### 6.1 Overview of Features

* Multi-model inference: YOLOv11 (PyTorch), Xception (TFLite), ResNet-50 (Keras/TF)
* Side-by-side outputs: top-1/top-5 labels + confidence per model
* Live charts: inference time, confidence trend, load-time breakdown
* Target filters: optional **Target Crop** / **Target State** dropdowns
* Offline operation once models are present

### 6.2 Supported Crops

Apple, Blueberry, Cherry, Grape, Maize, Orange, Peach, Pepper\_bell, Potato, Raspberry, Soybean, Squash, Strawberry, Tomato, **Unknown**

### 6.3 Supported States / Diseases

Apple\_scab, Bacterial\_spot, Black\_rot, Cedar\_apple\_rust, Cercospora\_leaf\_spot, Common\_rust, Early\_blight, Esca\_(Black\_Measles), Haunglongbing\_(Citrus\_greening), Late\_blight, Leaf\_Mold, Leaf\_blight\_(Isariopsis\_Leaf\_Spot), Leaf\_scorch, Northern\_Leaf\_Blight, Powdery\_mildew, Septoria\_leaf\_spot, Spider\_mites, Target\_Spot, Tomato\_Yellow\_Leaf\_Curl\_Virus, Tomato\_mosaic\_virus, **Unknown, healthy**

### 6.4 Using the App

1. **Launch**: python COSC591-plant-classifier.py
2. **Open image**: File, Open (supports .jpg, .jpeg, .png, .bmp, .webp, .tif)
3. **(Optional)** set **Target Crop/State** to narrow expected outputs
4. **Predict (All Models)**: runs Xception, ResNet-50, and YOLO
5. **Review**: top-1/top-5 predictions + confidence; charts update live
6. **Exit**: Esc to leave fullscreen, File Exit or **Ctrl/Cmd-Q**

A screenshot of a computer

AI-generated content may be incorrect.

Figure 1:Home screen

## 7. Troubleshooting

### 7.1 “DNN library initialization failed” (TensorFlow)

Likely VRAM is nearly full.

|  |
| --- |
| nvidia-smi  export TF\_FORCE\_GPU\_ALLOW\_GROWTH=true  export CUDA\_VISIBLE\_DEVICES=0 # choose a less-busy GPU  # also reduce batch size / image size |

### 7.2 No GPU Detected by TF/Torch

Not in a GPU allocation or the node has no free GPU.

|  |
| --- |
| python - <<'PY'  import tensorflow as tf, torch  print("TF:", tf.\_\_version\_\_, tf.config.list\_physical\_devices("GPU"))  print("Torch:", torch.\_\_version\_\_, torch.cuda.is\_available())  PY |

### 7.3 Missing / Tiny Model Files

Ensure required files exist and are **> 1 KB**:

|  |
| --- |
| models/yolo/best.pt  models/xception/xception\_pv1.tflite  models/xception/labels\_{crop,state}.txt  models/resnet50/resnet50\_stage2\_conv5.h5  models/resnet50/class\_index.csv or class\_names.json |

### 7.4 Common Python Deps

|  |
| --- |
| pip install pillow "matplotlib>=3.8,<3.11" opencv-python==4.11.0.86 |

## 8. Known-Good Versions (Linux GPU)

* **Python:** 3.10
* **TensorFlow:** **2.16.\*** (bundled CUDA 12.x, cuDNN 9.x) — use **tf.keras**
* **PyTorch:** **2.3.0** + **torchvision 0.18.0** (cu121 wheels)
* **Ultralytics:** 8.3.x
* **OpenCV-Python:** 4.11.0.86
* **numpy:** 1.26.x, **pillow:** 10.x, **matplotlib:** 3.8–3.10, **pandas:** 2.3.x, **h5py:** 3.14+