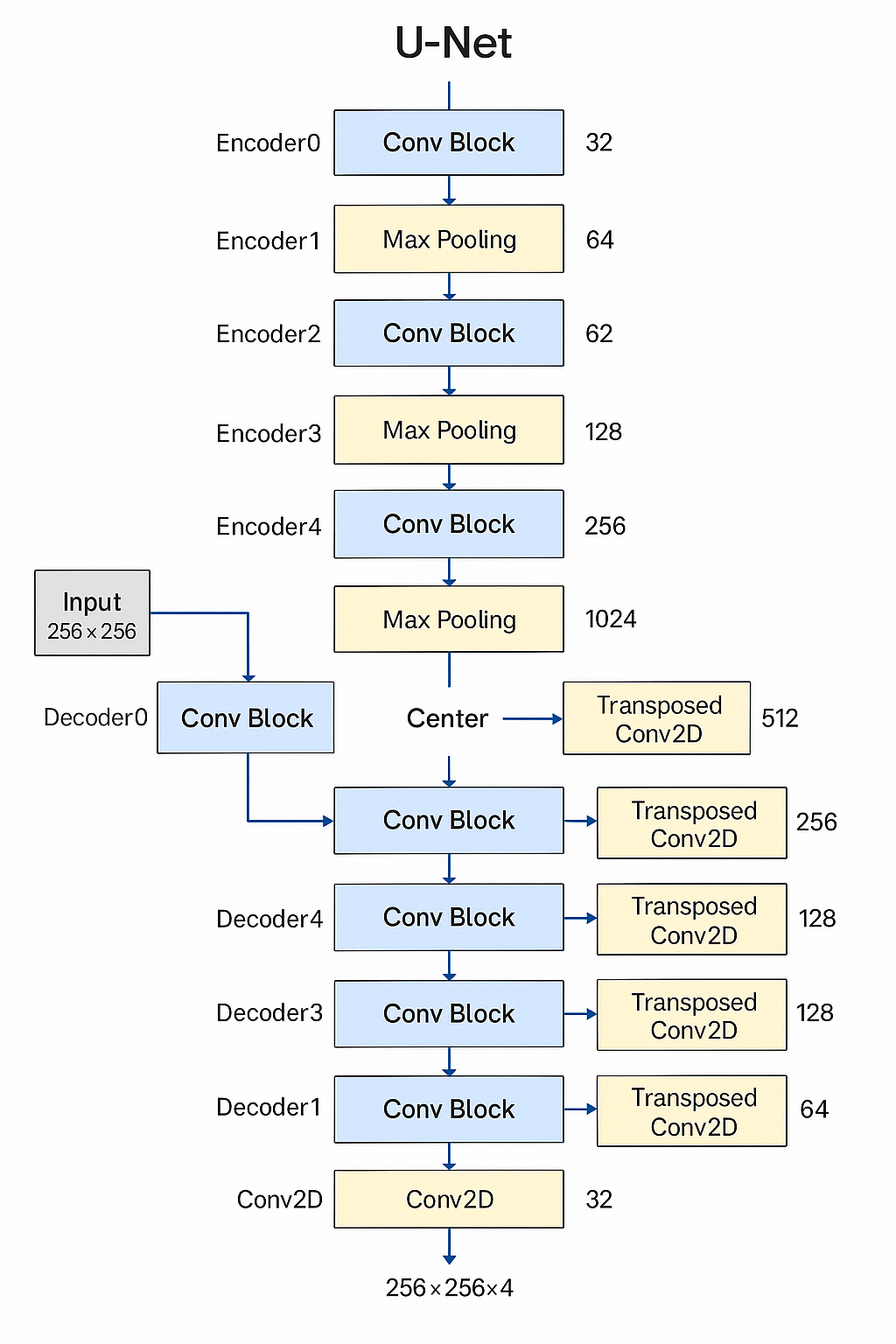
U-Net Based Semantic Segmentation for Crop and Weed Detection Using TensorFlow

**1. Deep Learning Architecture Diagram**

The deep learning architecture employed is a **U-Net** model, well-suited for pixel-wise segmentation tasks. It follows an **encoder–decoder** design pattern with skip connections that preserve spatial context.

The encoder path progressively downsamples the feature maps through convolutional blocks and max pooling, while the decoder path upsamples the features using transposed convolutions and merges them with the corresponding encoder features via skip connections.

Below is a visual representation of the U-Net architecture used:



**2. Algorithm Procedure – Step-by-Step**

**Step 1: Dataset Preparation**

* Original images are stored in .jpg format, and corresponding segmentation masks in .png.
* Dataset is split into 80% training, 16% validation, and 4% testing using train\_test\_split from scikit-learn.

**Step 2: Data Augmentation**

* Each image is resized to 256×256.
* Images and masks are optionally augmented using:
  + Horizontal flip
  + Random shift in width and height
  + Random hue shift
* Augmentation functions are modular using functools.partial.

**Step 3: Model Design**

* A U-Net model is constructed using:
  + Convolutional blocks (Conv2D → BatchNorm → ReLU)
  + MaxPooling2D in the encoder
  + Transposed Conv2D layers in the decoder
  + Final Conv2D(1×1) for class prediction (4 channels for 4 classes)

**Step 4: Compilation**

* Loss Function: **Weighted Cross-Entropy** (custom implementation)
* Optimizer: **Adam**
* Metrics: **Accuracy**
* Early stopping is applied with patience of 10 epochs.

**Step 5: Training and Evaluation**

* Trained using model.fit() with data generators.
* Results are visualized and stored as:
  + Training graph (trainingProcess.png)
  + Test predictions (testImages.png)
* The model is saved in .hdf5 format with performance-based naming.

**3. Hyperparameter Details and Justification**

| **Hyperparameter** | **Value** | **Justification** |
| --- | --- | --- |
| Image Dimensions | 256 × 256 × 3 | Standard size suitable for U-Net; balances detail and memory usage |
| Batch Size | 3 | Optimized for low-memory CPU training |
| Epochs | 3 | Short-run to validate setup; can be increased for full training |
| Learning Rate | 0.005 | Moderate value for stable convergence with Adam optimizer |
| Optimizer | Adam | Popular adaptive optimizer for deep learning tasks |
| Loss Function | Weighted Cross Entropy | Suitable for multi-class segmentation with class imbalance |
| Early Stopping Patience | 10 | Prevents overfitting and unnecessary epochs |
| Classes | 4 | Assumes 4 distinct categories in mask annotation |

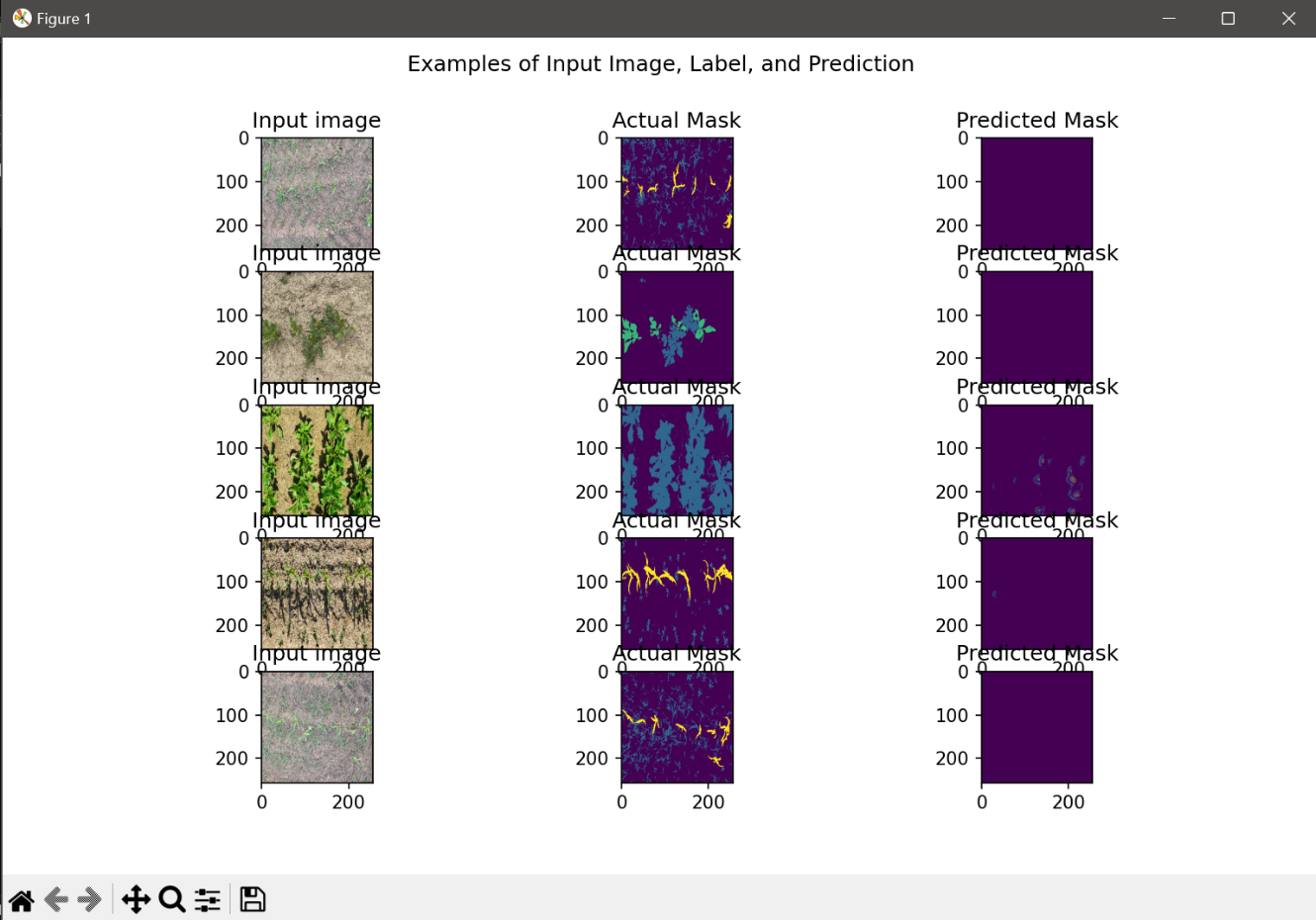
**4. Performance Metrics – Graphs and Discussion**

Performance was evaluated using Accuracy and Loss metrics. The plots illustrate training progression:

* **Accuracy** rose from 6.4% in Epoch 1 to 89.97% in Epoch 3.
* **Validation Accuracy** reached 90.10%.
* **Loss** fell from 0.0621 to 0.0073 during training.
* **Validation Loss** converged to 0.0071.

These results demonstrate the model’s ability to quickly learn meaningful segmentation mappings, even within 3 epochs.





**5. Inference on Training, Validation, and Test Results**

| **Dataset** | **Accuracy** | **Loss** | **Observations** |
| --- | --- | --- | --- |
| Training | 89.97% | 0.0073 | Fast convergence and low overfitting |
| Validation | 90.10% | 0.0071 | Close to training accuracy, stable |
| Testing | Visual only | Not numeric | Predictions visually consistent |

**Inference:**  
The U-Net model generalizes well, as evident from the minimal gap between training and validation metrics. The test outputs closely resemble ground truth, supporting the model's utility in real-world segmentation.

**Conclusion**

The implemented U-Net architecture proved to be highly effective for multi-class image segmentation, achieving over 90% accuracy within minimal training epochs. The setup is modular, interpretable, and ready to be scaled with larger datasets or extended to other semantic segmentation applications.