**PRACTICAL 9**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Name:** | Harsh Shah | **Semester:** | VI | **Division:** | 6 |
| **Roll No.:** | 21BCP359 | **Date:** |  | **Batch:** | G11 |
| **Aim:** | For your project definition demonstrate applicable task out of prediction and classification. Explain the entire work flow of your project through a single diagram. | | | | |

**Plant Disease Classification using CNN**

1. **Data Collection:**

* Gather high-quality images of plant leaves.
* Include images of healthy leaves and leaves with various diseases you aim to classify.
* Ensure the image collection is representative of the target diseases.

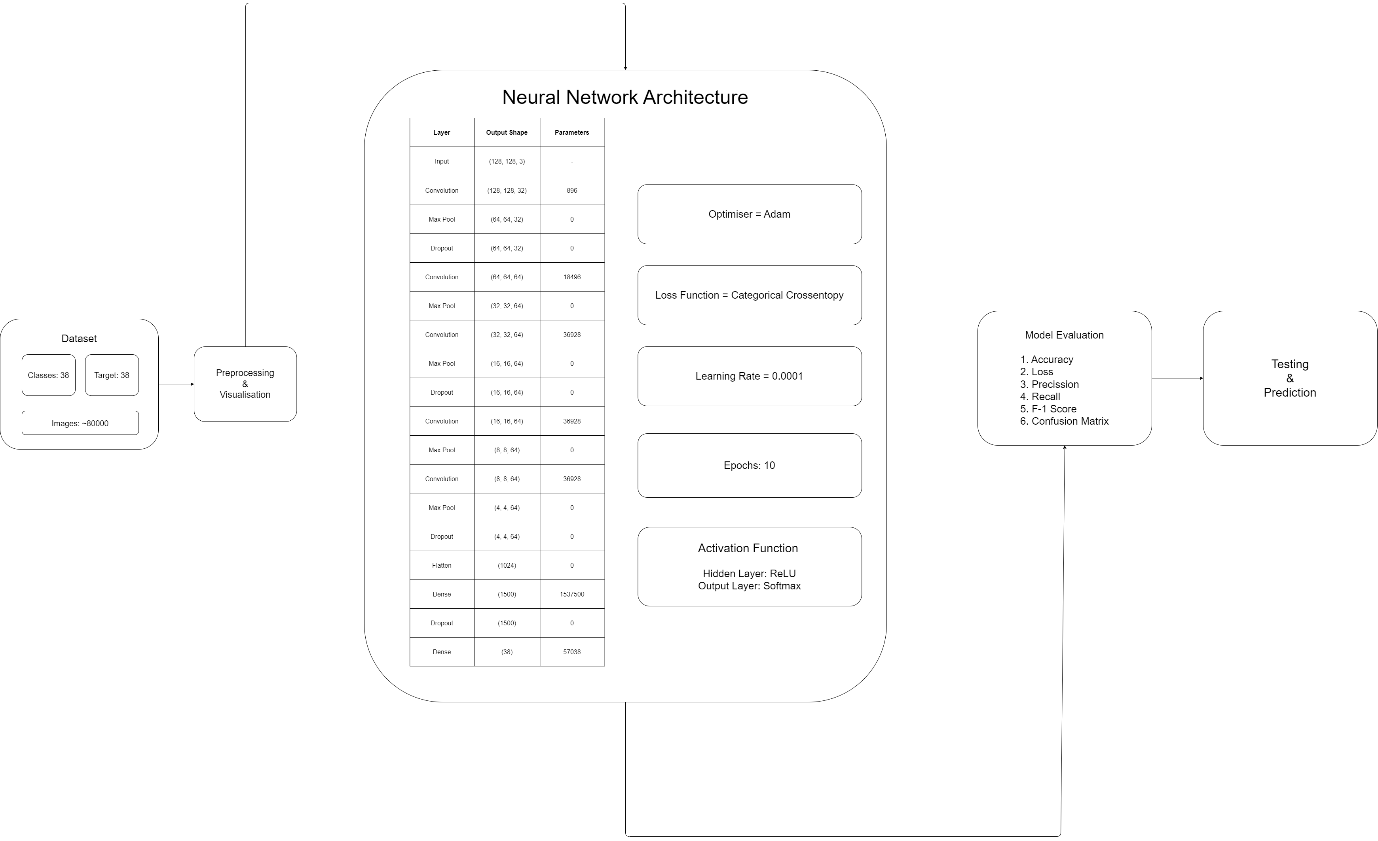
1. **Data Pre-processing:**

* Prepare the images for training the model.
  + 1. **Model Development**
* **Model Architecture:**
  + This model utilizes a Convolutional Neural Network (CNN).
  + CNNs excel at extracting spatial features from images, making them ideal for image classification tasks.
* **Building Blocks:**
  + **Convolutional Layers:** These layers apply filters to the image, identifying patterns and extracting features.
  + **Pooling Layers:** Reduce image dimensionality by summarizing features from previous layers (e.g., Max Pooling).
  + **Activation Functions:** Introduce non-linearity to the network, allowing it to learn complex relationships – ReLU and Softmax.
  + **Flatten Layer:** Transform the extracted features into a one-dimensional vector for feeding into fully-connected layers.
  + **Fully-Connected Layers:** Dense layers that process the flattened features and make the final classification decision.
    1. **Training Process:**
  + Divide the pre-processed data into training, validation, and test sets.
  + The training set is used to train the model, the validation set monitors training progress, and the test set evaluates final model performance on unseen data.
  + The model learns by iteratively adjusting its internal weights based on the training data and the chosen loss function - Categorical Cross entropy for multi-class classification.
  + The optimizer - Adam guides these adjustments to minimize the loss function and improve classification accuracy.

**Evaluation and Deployment**

1. **Model Evaluation:**
   * After training, assess the model's performance on the held-out test set.
   * Common metrics include **accuracy**, **precision**, **recall**, and **F1-score**.
2. **Deployment:**
   * Once satisfied with the model's performance, deploy it for real-world use.
   * This could involve integrating it into a mobile application or web service for on-demand plant disease prediction.

**Flow Chart**

****