IMNM training pseudocode.

|  |
| --- |
| **Algorithms IMNM training pseudocode** |
| 1: **INPUT:** *X* is the training images with labels; *Y* is the validating images with labels. |
| 2: **OUTPUT:** Optimal classifier and pepper leaf disease identification. |
| 3: Collect the training and validating images *X* and *Y*; load into the deep learning model. |
| 4: Use the improved ResNet， DCN， and PPN to create IMNM\_model. |
| 5: Set *N* as the number of epochs |
| 6: **for** epoch = 1, 2, …, *N* **do** |
| 7: **for** batch size of 32 **do** |
| 8: **for** learning rate of 0.01 **do** |
| 9: Save\_loss.append(train*X*.loss(), val*Y*.loss()) |
| 10: Save\_accuracy.append(train*X*.acc(), val*Y*.acc()) |
| 11: Optimizer = load\_optimizer() |
| 12: Loss = load\_loss(train*X*.loss(), val*Y*.loss()) |
| 13: Model = load\_IMNM\_model (input\_shape, n\_classes) |
| 14: Train*X* = load\_dataset\_for\_train() |
| 15: For epoch in range(0, max\_epoch) |
| 16: train() |
| 17: end for |
| 18: end for |
| 19: end for |
| 20: Test\_dataset = load\_dataset\_for\_test() |
| 21: Compute the accuracy and loss of the test dataset |
| 22: Output pepper leaf disease categories. |

\*Note: DCN refers to the Dynamic Convolutional Network; PPN refers to the Progressive Prototype Network