

SCREENSHOTS:

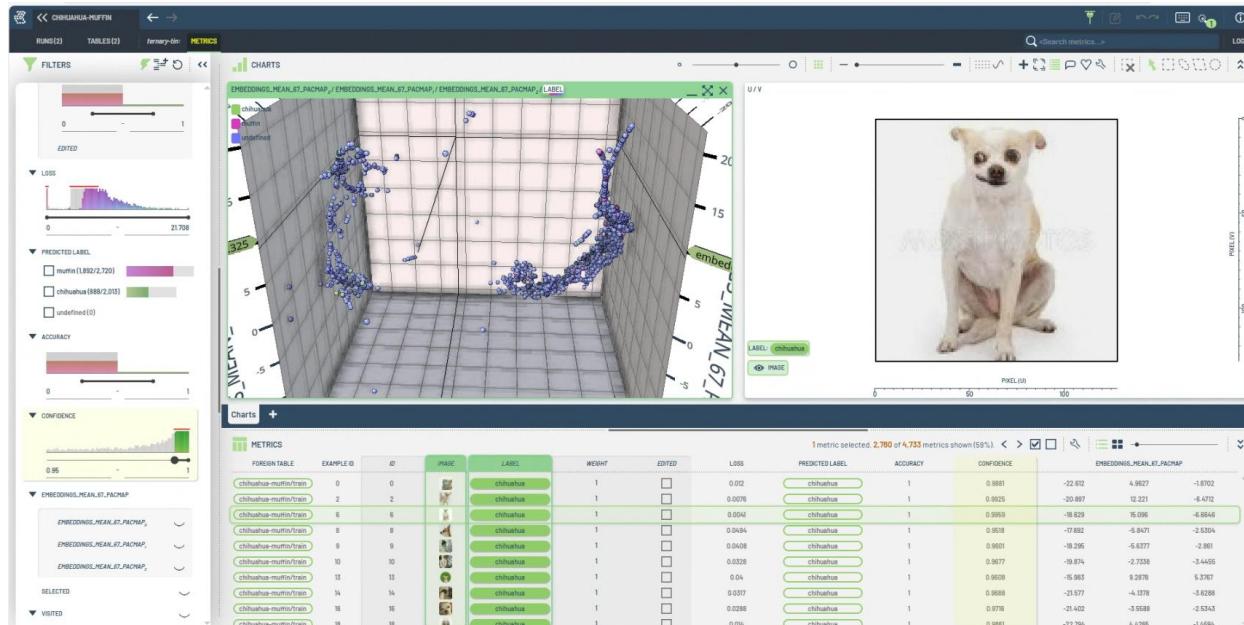
Initial Accuracy:

```

[...]
Epoch 10/10 - 100%
Validation Epoch 0/100: 100%
Epoch 10/10 - Val Loss: 0.4383, Val Accuracy: 83.11%
[OK] New best model Validation accuracy: 83.11%
Epoch 10/10 - 100%
Validation Epoch 1/10: 100%
Epoch 10/10 - Val Loss: 0.5338, Val Accuracy: 82.26%
Epoch 10/10 - 100%
Validation Epoch 2/10: 100%
Epoch 10/10 - Val Loss: 0.4409, Val Accuracy: 83.55%
[OK] New best model Validation accuracy: 83.55%
Epoch 10/10 - 100%
Validation Epoch 3/10: 100%
Epoch 10/10 - Val Loss: 0.3960, Val Accuracy: 84.80%
[OK] New best model Validation accuracy: 84.80%
Epoch 10/10 - 100%
Validation Epoch 4/10: 100%
Epoch 10/10 - Val Loss: 0.3284, Val Accuracy: 83.45%
[OK] Training complete!
Best validation accuracy: 84.80%
[OK] Loaded best model for metrics collection
[OK] Best model saved to 'resnet10_classifier_best.pth'
Collecting metrics on train set with best model...
Collecting SLC metrics for dataset chihuahua-muffin
[OK] Metrics collection complete!
Reducing embeddings using PaCMAP...
WARNING:pacmap.pacmap:Note: 'n_components = 2' have not been thoroughly tested.
Embeddings reduction complete!
[OK] All done! View results at https://dashboard.jlci.ai
PS C:\Users\june\Desktop\hackathon>

```

Version 1 - Confidence level above 0.95



Version 2 - Greater loss 21.708

Project > README.md > starter_notebook.ipynb

Terminal Copilot (4) Local +

```

Validation Epoch 1/10: 100% | 1/14 [00:00<00:00, 1x/10it/s, val_loss=0.404]
Epoch 7/10 - Val Loss: 0.3994, Val Accuracy: 85.14%
Epoch 8/10: 100%
Validation Epoch 8/10: 100% | 1/14 [00:00<00:00, 1x/10it/s, val_loss=0.394]
Epoch 9/10: 100% | 1/14 [00:00<00:00, 1x/10it/s, val_loss=0.5223, Val Accuracy: 79.56%]
Epoch 9/10: 100% | 1/14 [00:00<00:00, 1x/10it/s, val_loss=0.5223, Val Accuracy: 79.56%]
Validation Epoch 9/10: 100% | 1/14 [00:00<00:00, 1x/10it/s, val_loss=0.5223, Val Accuracy: 79.56%]
Epoch 9/10 - Val Loss: 0.4491, Val Accuracy: 85.98%
Epoch 10/10: 100% | 1/14 [00:00<00:00, 1x/10it/s, val_loss=0.5246, Val Accuracy: 83.19%]
Validation Epoch 10/10: 100% | 1/14 [00:00<00:00, 1x/10it/s, val_loss=0.5246, Val Accuracy: 83.19%]
Epoch 10/10 - Val Loss: 0.5246, Val Accuracy: 83.19%
=====
Training complete!
Best validation accuracy: 86.57%
=====

[OK] Loaded best model for metrics collection
[OK] Best model saved to 'resnet18_classifier.best.pth'

Collecting metrics on train set with best model...
Collecting SLC metrics for dataset chihuahua-muffin 100% 0:00:30
[OK] Metrics collection complete!

Reducing embeddings using PaCMAP...
WARNING:pacmap.pacmap:Note: 'n_components != 2' have not been thoroughly tested.
Embeddings reduction complete!
[OK] All done! View results at https://dashboard.jlcc.ai
=====
(3tc-env) PS C:\Users\jheee\Desktop\Hackathon>

```

Hackathon > starter_notebook.ipynb

Version 3 - labelled muffin and chihuahua using ID with confidence more than .94

Project > README.md > starter_notebook.ipynb

Terminal Copilot (4) Local +

```

print("Training complete!")
print(f"Best validation accuracy: {best_val_accuracy:.2f}%")
=====
17.10it/s, loss=0.2401
Validation Epoch 6/10: 100% | 7/74 [00:02<00:00, 25.52it/s, val_loss=0.269]
Epoch 6/10 - Val Loss: 0.6561, Val Accuracy: 78.04%
Epoch 7/10: 100% | 13/13 [00:00<00:00, 17.01it/s, loss=0.521]
Validation Epoch 7/10: 100% | 7/74 [00:02<00:00, 24.69it/s, val_loss=0.553]
=====
Epoch 6/10: 100%
Validation Epoch 6/10: 100% | 1/14 [00:00<00:00, 1x/10it/s, val_loss=0.4873, Val Accuracy: 82.77%]
Epoch 7/10: 100% | 1/14 [00:00<00:00, 1x/10it/s, val_loss=0.3290, Val Accuracy: 86.91%]
[OK] New best model! Validation accuracy: 86.91%
Epoch 8/10: 100%
Validation Epoch 8/10: 100% | 1/14 [00:00<00:00, 1x/10it/s, val_loss=0.4488, Val Accuracy: 84.21%]
Epoch 9/10: 100%
Validation Epoch 9/10: 100% | 1/14 [00:00<00:00, 1x/10it/s, val_loss=0.3519, Val Accuracy: 88.01%]
[OK] New best model! Validation accuracy: 88.01%
Epoch 10/10: 100%
Validation Epoch 10/10: 100% | 1/14 [00:00<00:00, 1x/10it/s, val_loss=0.938, Val Accuracy: 85.47%]
=====
Training complete!
Best validation accuracy: 88.01%
=====

[OK] Loaded best model for metrics collection
[OK] Best model saved to 'resnet18_classifier.best.pth'

Collecting metrics on train set with best model...
Collecting SLC metrics for dataset chihuahua-muffin 88% 0:00:04

```

Hackathon > starter_notebook.ipynb

Version 4 - labelled muffin with confidence more than .94

The screenshot shows a Jupyter Notebook interface with several tabs open. The active tab is 'starter_notebook.ipynb'. The code in the cell includes printing training completion and validation accuracy. The terminal below shows the execution of the code, including training epochs from 0 to 10, validation metrics, and the saving of a best model. A progress bar indicates the completion of metrics collection.

```

55 print("Training complete!")
56 print("Best validation accuracy: %s" % best_val_accuracy)
57 print("*" * 60)
58
59 [17]: 27.7% [Validation Epoch 6/10: 100%|██████████| 74/74 [00:02<00:00, 25.52it/s, val_loss=0.269]
60 Validation Epoch 6/10: 100%|██████████| 74/74 [00:02<00:00, 25.52it/s, val_loss=0.269]
61 Epoch 7/10: 100%|██████████| 69/69 [00:05<00:00, 12.95it/s, val_loss=0.059]
62 Epoch 8/10: 100%|██████████| 69/69 [00:05<00:00, 12.95it/s, val_loss=0.059]
63 Epoch 9/10: 100%|██████████| 69/69 [00:05<00:00, 10.69it/s, loss=2.1]
64 Epoch 10/10: 100%|██████████| 74/74 [00:04<00:00, 11.76it/s, val_loss=0.101]
65 Epoch 10/10: 100%|██████████| 74/74 [00:04<00:00, 11.76it/s, val_loss=0.101]
66 Epoch 10/10 - Val Loss: 0.3766, Val Accuracy: 86.32%
67 Epoch 10/10: 100%|██████████| 74/74 [00:04<00:00, 11.76it/s, val_loss=0.101]
68 Epoch 10/10 - Val Loss: 0.3776, Val Accuracy: 85.64%
69
70 =====
71 Training complete!
72 Best validation accuracy: 88.60%
73
74
75 [OK] Loaded best model for metrics collection
76 [OK] Best model saved to 'resnet18_classifier_best.pth'
77
78 Collecting metrics on train set with best model...
79 Collecting SLC metrics for dataset chihuahua-muffin 100% 0:00:30
80
81 [OK] Metrics collection complete!
82 Reducing embeddings using PaCMAP...
83 WARNING:pacmap.pacmap:Note: 'n_components = 2' have not been thoroughly tested.
84 Embeddings reduction complete!
85
86 [OK] All done! View results at https://embedboard.slc.ml
87
88 [SLC-env] PS C:\Users\juhee\Desktop\Hackathon>

```

Version 5 - CL > 0.95, predicted label - chihuahua, dropping those into labels, same for muffins, and outliers labeled as undefined.

The screenshot shows a Jupyter Notebook interface with several tabs open. The active tab is 'starter_notebook.ipynb'. The code in the cell includes printing training completion and validation accuracy. The terminal below shows the execution of the code, including training epochs from 4 to 10, validation metrics, and the saving of a best model. A progress bar indicates the completion of metrics collection.

```

55 print("Training complete!")
56 print("Best validation accuracy: %s" % best_val_accuracy)
57 print("*" * 60)
58
59 [17]: 27.7% [Validation Epoch 6/10: 100%|██████████| 74/74 [00:02<00:00, 25.52it/s, val_loss=0.269]
60 Validation Epoch 6/10: 100%|██████████| 74/74 [00:02<00:00, 25.52it/s, val_loss=0.269]
61 Epoch 7/10: 100%|██████████| 69/69 [00:05<00:00, 12.95it/s, val_loss=0.059]
62 Epoch 8/10: 100%|██████████| 69/69 [00:05<00:00, 12.95it/s, val_loss=0.059]
63 Epoch 9/10: 100%|██████████| 69/69 [00:05<00:00, 10.69it/s, loss=2.1]
64 Epoch 10/10: 100%|██████████| 74/74 [00:04<00:00, 11.76it/s, val_loss=0.101]
65 Epoch 10/10 - Val Loss: 0.3297, Val Accuracy: 89.02%
66 [OK] New best model! Validation accuracy: 89.02%
67 Epoch 10/10: 100%|██████████| 74/74 [00:04<00:00, 11.76it/s, val_loss=0.101]
68 Validation Epoch 4/10: 100%|██████████| 74/74 [00:05<00:00, 13.06it/s, val_loss=0.186]
69 Epoch 4/10 - Val Loss: 0.3297, Val Accuracy: 89.02%
70 [OK] New best model! Validation accuracy: 89.02%
71 Epoch 5/10: 100%|██████████| 69/69 [00:05<00:00, 10.41it/s, loss=0.329]
72 Validation Epoch 5/10: 100%|██████████| 74/74 [00:04<00:00, 11.51it/s, val_loss=0.175]
73 Epoch 6/10: 100%|██████████| 69/69 [00:05<00:00, 10.53it/s, loss=0.3455]
74 Epoch 6/10 - Val Loss: 0.3455, Val Accuracy: 88.15%
75 Epoch 7/10: 100%|██████████| 69/69 [00:05<00:00, 10.53it/s, loss=0.3079]
76 Validation Epoch 6/10: 100%|██████████| 74/74 [00:04<00:00, 11.94it/s, val_loss=0.0834]
77 [OK] New best model! Validation accuracy: 89.10%
78 Epoch 7/10: 100%|██████████| 69/69 [00:05<00:00, 10.53it/s, loss=0.3079]
79 Validation Epoch 7/10: 100%|██████████| 74/74 [00:04<00:00, 11.53it/s, val_loss=0.18]
80 Epoch 8/10: 100%|██████████| 69/69 [00:05<00:00, 10.52it/s, loss=0.3092]
81 Epoch 8/10 - Val Loss: 0.3092, Val Accuracy: 88.51%
82 Epoch 9/10: 100%|██████████| 69/69 [00:05<00:00, 10.52it/s, loss=0.3074]
83 Epoch 9/10 - Val Loss: 0.3074, Val Accuracy: 88.43%
84 Epoch 10/10: 100%|██████████| 69/69 [00:05<00:00, 10.53it/s, loss=0.3178]
85 Epoch 10/10 - Val Loss: 0.3178, Val Accuracy: 88.85%
86 [OK] New best model! Validation accuracy: 89.10%
87 Epoch 10/10: 100%|██████████| 74/74 [00:04<00:00, 12.25it/s, val_loss=0.158]
88 Validation Epoch 10/10: 100%|██████████| 74/74 [00:04<00:00, 7.48it/s, loss=0.247]
89 Epoch 10/10 - Val Loss: 0.4202, Val Accuracy: 85.05%
90
91 =====
92 Training complete!
93 Best validation accuracy: 89.10%
94
95 [OK] Loaded best model for metrics collection
96 [OK] Best model saved to 'resnet18_classifier_best.pth'
97
98 Collecting metrics on train set with best model...
99 Collecting SLC metrics for dataset chihuahua-muffin 62% 0:00:12

```

Version 6 - CL > 0.95, predicted label - chihuahua, dropping those into labels, same for muffins, and outliers labeled as undefined.

```

README.md  starter_notebook.ipynb
[1]: print("Training complete!")
[2]: print(f"Best validation accuracy: {best_val_accuracy:.2f}%")
[3]: print("-" * 60)

[4]: LY-JULYIN: U009D-Q-EWU]
Validation Epoch 0/10: 100%|██████████| 74/74
[00:02<00:00, 25.52it/s, val_loss=0.269]

Epoch 7/10 - Val Loss: 0.3155, Val Accuracy: 89.19%
[OK] New best model! Validation accuracy: 89.19%
Epoch 8/10: 100%
Validation Epoch 8/10: 100%
Epoch 8/10 - Val Loss: 0.3988, Val Accuracy: 84.46%
Epoch 9/10: 100%
Validation Epoch 9/10: 100%
Epoch 9/10 - Val Loss: 0.3965, Val Accuracy: 83.45%
Epoch 10/10: 100%
Validation Epoch 10/10: 100%
Epoch 10/10 - Val Loss: 0.3549, Val Accuracy: 88.94%

=====
Training complete!
Best validation accuracy: 89.19%
=====

[OK] Loaded best model for metrics collection
[OK] Best model saved to 'resnet18_classifier_best.pth'

Collecting metrics on train set with best model...
Collecting SLC metrics for dataset chihuahua-muffin
[OK] Metrics collection complete!

Reducing embeddings using PaCMAP...
WARNING: pacmap.pacmap: Note: 'n_components = 2' have not been thoroughly tested.
Embeddings reduction complete!

[OK] All done! View results at https://dashboard.3tc.ai
=====
(3tc-env) PS C:\Users\Junee\Desktop\Hackathon>

```

765:1 LF UTF-8 4 spaces Python 3.12

Version 7: CL > 0.95, predicted label - chihuahua, dropping those into labels, and outliers labeled as undefined.

```

README.md  starter_notebook.ipynb
[1]: print("Training complete!")
[2]: print(f"Best validation accuracy: {best_val_accuracy:.2f}%")
[3]: print("-" * 60)

[4]: LY-JULYIN: U009D-Q-EWU]
Validation Epoch 0/10: 100%|██████████| 74/74
[00:02<00:00, 25.52it/s, val_loss=0.269]

Validation Epoch 4/10: 100%|██████████| 74/74
[00:06<00:00, 12.18it/s, val_loss=0.199]
Epoch 4/10 - Val Loss: 0.3583, Val Accuracy: 88.51%
[OK] New best model! Validation accuracy: 88.51%
Epoch 5/10: 100%
Validation Epoch 5/10: 100%
Epoch 5/10 - Val Loss: 0.3635, Val Accuracy: 88.18%
Epoch 6/10: 100%
Validation Epoch 6/10: 100%
Epoch 6/10 - Val Loss: 0.4879%, Val Accuracy: 82.01%
Epoch 7/10: 100%
Validation Epoch 7/10: 100%
Epoch 7/10 - Val Loss: 0.5739, Val Accuracy: 77.53%
Epoch 8/10: 100%
Validation Epoch 8/10: 100%
Epoch 8/10 - Val Loss: 0.3804, Val Accuracy: 85.90%
Epoch 9/10: 100%
Validation Epoch 9/10: 100%
Epoch 9/10 - Val Loss: 0.3273, Val Accuracy: 89.27%
[OK] New best model! Validation accuracy: 89.27%
Epoch 10/10: 100%
Validation Epoch 10/10: 100%
Epoch 10/10 - Val Loss: 0.4579, Val Accuracy: 86.40%

=====
Training complete!
Best validation accuracy: 89.27%
=====

[OK] Loaded best model for metrics collection
[OK] Best model saved to 'resnet18_classifier_best.pth'

Collecting metrics on train set with best model...
Collecting SLC metrics for dataset chihuahua-muffin
[OK] Metrics collection complete!

Reducing embeddings using PaCMAP...
WARNING: pacmap.pacmap: Note: 'n_components = 2' have not been thoroughly tested.
Embeddings reduction complete!

[OK] All done! View results at https://dashboard.3tc.ai
=====
(3tc-env) PS C:\Users\Junee\Desktop\Hackathon>

```

765:1 LF UTF-8 4 spaces Python 3.12

Version 8: CL > 0.99, labeled - undefined selected, predicted label - Muffin, dragging those into labels, and done same for Chihuahua, and outliers labeled as undefined.

```

Epoch 5/15 - Val Loss: 0.3877, Val Accuracy: 86.57%
Epoch 10/15: 100% | 125/125 [00:12:00:00, 9.88it/s, loss=0.098]
Epoch 10/15: 100% | 74/74 [00:00<00:00, 11.19it/s, val_loss=0.118]
Validation Epoch 10/15: 100% | 125/125 [00:16:00:00, 7.50it/s, loss=0.109]
Epoch 10/15 - Val Loss: 0.3794, Val Accuracy: 86.57%
Epoch 11/15: 100% | 125/125 [00:16:00:00, 7.50it/s, loss=0.109]
Validation Epoch 11/15: 100% | 74/74 [00:00<00:00, 11.37it/s, val_loss=0.119]
Epoch 11/15 - Val Loss: 0.4059, Val Accuracy: 84.63%
Epoch 12/15: 100% | 125/125 [00:12:00:00, 9.66it/s, loss=0.111]
Validation Epoch 12/15: 100% | 74/74 [00:00<00:00, 11.43it/s, val_loss=0.136]
Epoch 12/15 - Val Loss: 0.5815, Val Accuracy: 78.12%
Epoch 13/15: 100% | 125/125 [00:16:00:00, 7.43it/s, loss=0.0721]
Validation Epoch 13/15: 100% | 74/74 [00:00<00:00, 11.18it/s, val_loss=0.146]
Epoch 13/15 - Val Loss: 0.3728, Val Accuracy: 86.66%
Epoch 14/15: 100% | 125/125 [00:13:00:00, 9.45it/s, loss=0.139]
Validation Epoch 14/15: 100% | 74/74 [00:00<00:00, 11.25it/s, val_loss=0.144]
Epoch 14/15 - Val Loss: 0.3388, Val Accuracy: 89.19%
Epoch 15/15: 100% | 125/125 [00:12:00:00, 10.07it/s, loss=0.203]
Validation Epoch 15/15: 100% | 74/74 [00:00<00:00, 11.14it/s, val_loss=0.296]
Epoch 15/15 - Val Loss: 0.4406, Val Accuracy: 81.93%
=====
Training complete!
Best validation accuracy: 89.44%
=====
[OK] Loaded best model for metrics collection
[OK] Best model saved to 'resnet18_classifier.best.pth'
Collecting metrics on train set with best model...
  Collecting SLC metrics for dataset chihuahua-muffin

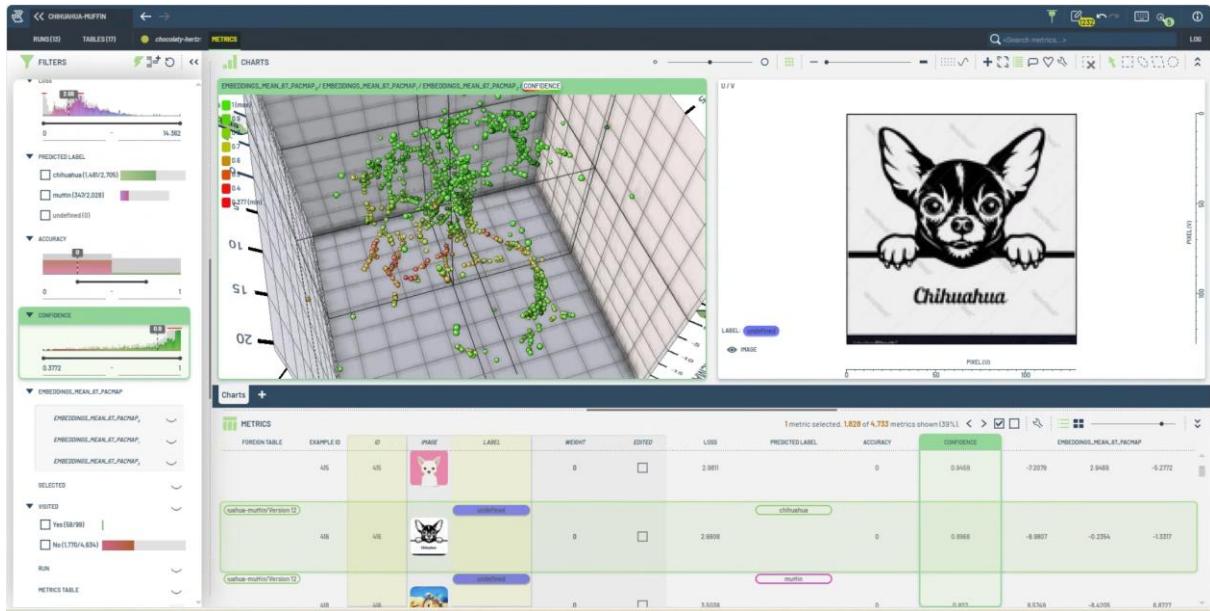
```

Version 9: CL > 0.99, labeled - undefined selected, predicted label - chihuahua, dragging those into labels, and done same for muffins, and outliers labeled as undefined.

```

Epoch 11/15 - Val Loss: 0.4163, Val Accuracy: 80.49%
Epoch 12/15: 100% | 137/137 [00:13:00:00, 10.19it/s, loss=0.13]
Validation Epoch 12/15: 100% | 74/74 [00:00<00:00, 11.31it/s, val_loss=0.112]
Epoch 12/15 - Val Loss: 0.3491, Val Accuracy: 89.27%
Epoch 13/15: 100% | 137/137 [00:13:00:00, 10.25it/s, loss=0.351]
Validation Epoch 13/15: 100% | 74/74 [00:00<00:00, 11.22it/s, val_loss=0.386]
Epoch 13/15 - Val Loss: 0.4852, Val Accuracy: 84.12%
Epoch 14/15: 100% | 137/137 [00:14:00:00, 9.41it/s, loss=0.389]
Validation Epoch 14/15: 100% | 74/74 [00:00<00:00, 11.47it/s, val_loss=0.391]
Epoch 14/15 - Val Loss: 0.3323, Val Accuracy: 89.61%
[OK] New best model! Validation accuracy: 89.61%
Epoch 15/15: 100% | 137/137 [00:13:00:00, 9.88it/s, loss=0.316]
Validation Epoch 15/15: 100% | 74/74 [00:00<00:00, 11.30it/s, val_loss=0.146]
Epoch 15/15 - Val Loss: 0.4406, Val Accuracy: 71.88%
=====
Training complete!
Best validation accuracy: 89.61%
=====
[OK] Loaded best model for metrics collection
[OK] Best model saved to 'resnet18_classifier.best.pth'
Collecting metrics on train set with best model...
  Collecting SLC metrics for dataset chihuahua-muffin

```



Version 10: Pulled confidence column up to the chart's label (title), make the blob coloured on the basis of their confidence instead of the label. Selected CL < 0.68, predicted label - chihuahua, separated label by using filter ID, dragging the predicted label into labels, for muffins, labeled them by selecting all, and outliers labeled as undefined. predicted label - Muffin, did same for this.

The figure shows a code editor interface with a project structure on the left. The 'train.py' file contains the following code:

```

26     python train.py
27     ...
28
29     > import ...
30
31     # Training hyperparameters
32     EPOCHS = 10
33     BATCH_SIZE = 16
34     LEARNING_RATE = 0.0001
35
36     # Project configuration
37     PROJECT_NAME = "chihuahua-Muffin"
38     DATASET_NAME = "chihuahua-muffin"
39
40
41
42
43
44
45
46
47
48
49

```

The terminal window shows training logs:

```

Validation Epoch 0/10: 100% | 143/143 [00:05<00:00, 10.87it/s, val_loss=0.105]
Epoch 6/10 - Val Loss: 0.3175, Val Accuracy: 98.62%
[OK] New best model! Validation accuracy: 98.62%
Epoch 7/10: 100% | 143/143 [00:05<00:00, 9.98it/s, loss=0.195]
Validation Epoch 7/10: 100% | 143/143 [00:05<00:00, 11.41it/s, val_loss=0.138]
Epoch 7/10 - Val Loss: 0.3174, Val Accuracy: 91.59%
[OK] New best model! Validation accuracy: 91.59%
Epoch 8/10: 100% | 143/143 [00:05<00:00, 9.34it/s, loss=0.144]
Validation Epoch 8/10: 100% | 143/143 [00:05<00:00, 11.32it/s, val_loss=0.125]
Epoch 8/10 - Val Loss: 0.3151, Val Accuracy: 89.61%
Epoch 9/10: 100% | 143/143 [00:05<00:00, 9.12it/s, loss=0.143]
Validation Epoch 9/10: 100% | 143/143 [00:05<00:00, 12.38it/s, val_loss=0.294]
Epoch 9/10 - Val Loss: 0.3152, Val Accuracy: 90.03%
Epoch 10/10: 100% | 143/143 [00:05<00:00, 9.79it/s, loss=0.0852]
Validation Epoch 10/10: 100% | 143/143 [00:05<00:00, 10.64it/s, val_loss=0.23]
Epoch 10/10 - Val Loss: 0.3152, Val Accuracy: 90.62%
=====
Training complete!
Best validation accuracy: 91.39%
=====

[OK] Loaded best model for metrics collection
[OK] Best model saved to 'resnet18_classifier_best.pth'
Collecting metrics on train set with best model...
  Collecting SLC metrics for dataset chihuahua-muffin

```

The Chat pane shows a message about improving model accuracy with embeddings.

Version 11: Pulled confidence column up to the chart's label (title), make the blob coloured on the basis of their confidence instead of the label. Selected CL < 0.70, predicted label - chihuahua, separated label by using filter ID, dragging the predicted label into labels, for muffins, labeled them by selecting all, and outliers labeled as undefined. predicted label - Muffin, did same for this.

```

class ResNet18Classifier(nn.Module):
    def __init__(self, num_classes=2):
        super().__init__()
        self.resnet = models.resnet18(weights=None)

        # Get the number of features from ResNet's final layer
        resnet_features = self.resnet.fc.in_features

        # Remove the original final layer
        self.resnet.fc = nn.Identity()

        # Create new classification head
        self.classifier = nn.Sequential(
            nn.Linear(resnet_features, out_features=256),
            nn.ReLU(),
            nn.Dropout(0.3),
            nn.Linear(in_features=256, out_features=128),
            nn.ReLU(),
            nn.Dropout(0.3),
            nn.Linear(in_features=128, num_classes)
        )

    def forward(self, x):
        # Get ResNet features (without final classification layer)

```

EPOCHS = 20 and LEARNING_RATE = 0.001

```

EPOCHS = 20
BATCH_SIZE = 32
LEARNING_RATE = 0.001

# Project configuration
PROJECT_NAME = "Chihuahua-Muffin"
DATASET_NAME = "chihuahua-muffin"

device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
print(f"Using device: {device}")

Validation Epoch 20/20: 100%
Epoch 20/20 - Val Loss: 0.2870, Val Accuracy: 91.98%
[OK] New best model! Validation accuracy: 91.98%

Training complete!
Best validation accuracy: 91.98%

[OK] Loaded best model for metrics collection
[OK] Best model saved to 'resnet18_classifier_best.pth'
Collecting metrics on train set with best model...
Collecting 3LC metrics for dataset chihuahua-muffin
[OK] Metrics collection complete!

Reducing embeddings using PaCMAP...
WARNING:pacmap:Note: n_components != 2 have not been thoroughly tested.
Embeddings reduction complete!

[OK] All done! View results at https://dashboard.jlc.ai

```

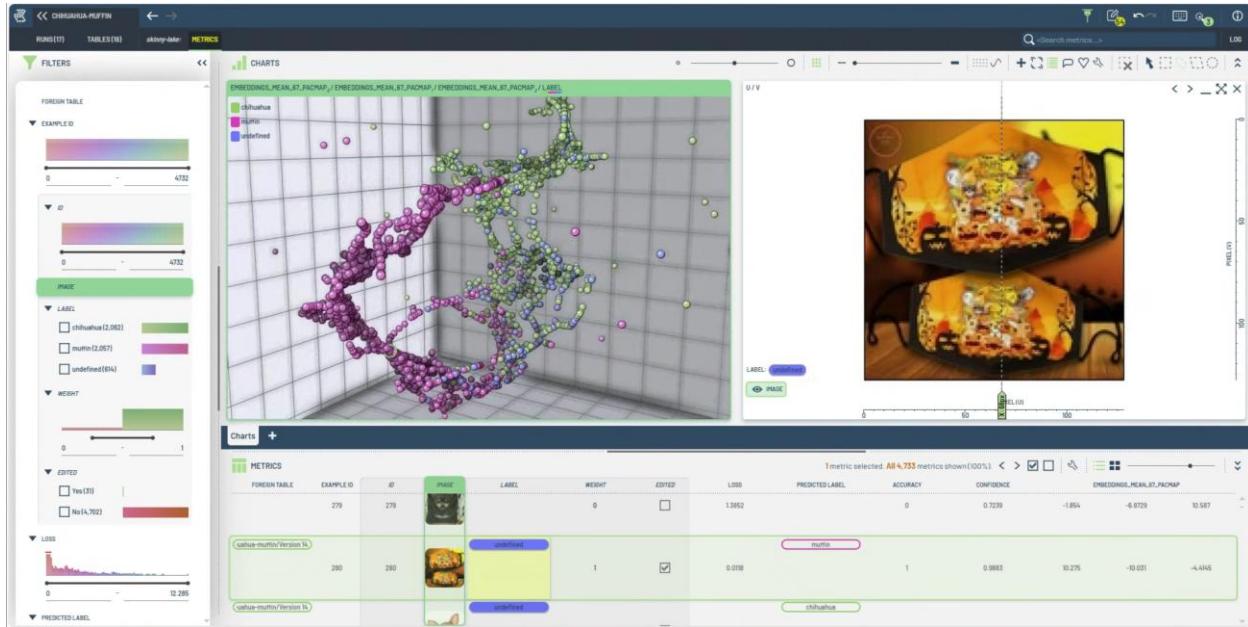
Version 12: Pulled confidence column up to the chart's label (title), make the blob coloured on the basis of their confidence instead of the label. cluster selection done on low confidence labels on chart.

```

Project
├── Hackathon
│   ├── ipynb_checkpoints
│   ├── 3lc-env
│   │   ├── include
│   │   ├── Lib
│   │   ├── Scripts
│   │   └── share
│   └── pyenv.cfg
│       ├── api.png
│       ├── dashboard.png
│       ├── service.png
│       └── test.png
└── train128
    ├── test128
    │   ├── chihuahua
    │   ├── muffin
    └── train128

Terminal Copilot (4) Local +
Epoch 16/20: 100% | 150/150 [00:21<00:00, 7.42it/s, loss=0.308]
Validation Epoch 16/20: 100% | 74/74 [00:05<00:00, 12.46it/s, val_loss=0.127]
Epoch 17/20: 100% | 150/150 [00:21<00:00, 6.65it/s, loss=0.299]
Validation Epoch 17/20: 100% | 74/74 [00:08<00:00, 9.12it/s, val_loss=0.328]
Epoch 17/20 - Val Loss: 0.4655, Val Accuracy: 85.50%
Epoch 18/20: 100% | 150/150 [00:21<00:00, 7.00it/s, loss=0.144]
Validation Epoch 18/20: 100% | 74/74 [00:07<00:00, 9.49it/s, val_loss=0.34]
Epoch 18/20 - Val Loss: 0.3941, Val Accuracy: 88.20%
Epoch 19/20: 100% | 156/156 [00:22<00:00, 7.05it/s, loss=0.211]
Validation Epoch 19/20: 100% | 74/74 [00:08<00:00, 9.23it/s, val_loss=0.498]
Epoch 19/20 - Val Loss: 0.3596, Val Accuracy: 90.88%
Epoch 20/20: 100% | 156/156 [00:22<00:00, 7.05it/s, loss=0.056]
Validation Epoch 20/20: 100% | 74/74 [00:08<00:00, 8.78it/s, val_loss=0.145]
[OK] New best model! Validation accuracy: 91.98%
=====
Training complete!
Best validation accuracy: 91.98%
=====
[OK] Loaded best model for metrics collection
[OK] Best model saved to 'resnet18_classifier.best.pth'
Collecting metrics on train set with best model...
Collecting SLC metrics for dataset chihuahua-muffin

```



Version 13: undefined selected, predicted - muffin, dragged the predicted label to labels and also checked the mislabelled. Same steps done for predicted - chihuahua, also considering confidence level

```
Usage:  
python train.py  
  
import ...  
  
# Training hyperparameters  
EPOCHS = 20  
BATCH_SIZE = 16  
LEARNING_RATE = 0.001  
  
# Project configuration  
PROJECT_NAME = 'Chihuahua-Muffin'  
DATASET_NAME = 'chihuahua-muffin'  
  
device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')  
  
Epoch 18/20: 100% [215/215 [00:23<00:00, 9.28it/s, loss=0.582]  
Validation Epoch 18/20: 100% [76/76 [00:08<00:00, 8.93it/s, val_loss=0.226]  
Epoch 18/20 - Val Loss: 0.2565, Val Accuracy: 92.31%  
[OK] New best model! Validation accuracy: 92.31%  
Epoch 19/20: 100% [215/215 [00:23<00:00, 9.28it/s, loss=0.235]  
Validation Epoch 19/20: 100% [74/74 [00:08<00:00, 8.87it/s, val_loss=0.182]  
Epoch 19/20 - Val Loss: 0.2471, Val Accuracy: 92.31%  
Epoch 20/20: 100% [215/215 [00:23<00:00, 9.28it/s, loss=0.185]  
Validation Epoch 20/20: 100% [74/74 [00:08<00:00, 8.85it/s, val_loss=0.0627]  
Epoch 20/20 - Val Loss: 0.3175, Val Accuracy: 91.22%  
  
*****  
Training complete!  
Best validation accuracy: 92.31%  
*****  
  
[OK] Loaded best model for metrics collection  
[OK] Best model saved to 'resnet18_classifier_best.pth'  
  
Collecting metrics on train set with best model...  
Collecting 3LC metrics for dataset chihuahua-muffin 100% 0:00:31  
  
[OK] Metrics collection complete!  
  
Reducing embeddings using PaCMAP...  
WARNING:pacmap.pacmap:Note: n_components != 2 have not been thoroughly tested.  
Hackathon > train.py
```

`transforms.RandomRotation(15), transforms.ColorJitter(brightness=0.2, contrast=0.2, saturation=0.2, hue=0.1),`

The screenshot shows a Jupyter Notebook interface with the following components:

- Project View:** On the left, a tree view of the project structure under "Hackathon".
- Code Editor:** The main area displays a Python script named "train.py". The code defines data transforms for training and validation datasets.
- Terminal Output:** Below the code editor, the terminal window shows the training process. It includes validation accuracy, epoch numbers, and a message indicating a new best model at epoch 19.
- Status Bar:** At the bottom right, it shows "43% 0:00:19".

Version 14: undefined labels selected - Embedding clusters - similar labelled with predicted status and removed weird/noisy images.

```

# DATA TRANSFORMS
# -----
# train_transform = transforms.Compose([
#     transforms.Resize(128),
#     transforms.RandomCrop(128),
#     transforms.RandomHorizontalFlip(),
#     # transforms.RandomRotation(15),
#     # transforms.ColorJitter(brightness=0.2, contrast=0.2, saturation=0.2, hue=0.1),
#     transforms.RandomAffine(degrees=0, shear=10, scale=(0.8, 1.2)),
#     transforms.ToTensor(),
#     transforms.Normalize([mean [0.485, 0.456, 0.406], std [0.229, 0.224, 0.225]]))
# ])
#
# val_transform = transforms.Compose([
#     transforms.Resize(128),
# ])

```

```

Epoch 20/20 - Val Loss: 0.2037, Val Accuracy: 91.30%
=====
Training complete!
Best validation accuracy: 93.50%
=====

[OK] Loaded best model for metrics collection
[OK] Best model saved to 'resnet18_classifier_best.pth'

Collecting metrics on train set with best model...
Collecting SLC metrics for dataset chihuahua-muffin 100% 0:06:36
[OK] Metrics collection complete!

Reducing embeddings using PaCMAP...
WARNING:pacmap.pacmap:Note: 'n_components != 2' have not been thoroughly tested.
Embeddings reduction complete!
=====

[OK] All done! View results at http://dashboard.Slc.a
[!] (Slc-env) PS C:\Users\juhee\Desktop\Hackathon>

```

Strong Data Augmentation: RandomResizedCrop, RandomHorizontalFlip, RandomRotation, ColorJitter. This is for preventing overfitting and helping the model generalize.

Dynamic Learning Rate: ReduceLROnPlateau scheduler. This will help the model train more effectively.

```

Validation Epoch 19/20: 100%
Epoch 19/20 - Val Loss: 0.1962, Val Accuracy: 93.10%
[OK] New best model! Validation accuracy: 93.10%
Epoch 20/20: 100%
Validation Epoch 20/20: 100%
Epoch 20/20 - Val Loss: 0.2037, Val Accuracy: 91.30%
=====
Training complete!
Best validation accuracy: 93.10%
=====

[OK] Loaded best model for metrics collection
[OK] Best model saved to 'resnet18_classifier_best.pth'

Collecting metrics on train set with best model...
Collecting SLC metrics for dataset chihuahua-muffin 100% 0:06:36
[OK] Metrics collection complete!

Reducing embeddings using PaCMAP...
WARNING:pacmap.pacmap:Note: 'n_components != 2' have not been thoroughly tested.

```

Version 15: Effective labelling through clustering and correcting the mislabelled with epochs = 50

```

Project Hackathon
├── Hackathon
│   ├── README.md
│   ├── starter_notebook.ipynb
│   └── train.py
├── content
└── test128
    ├── chihuahua
    ├── muffin
    └── train128
        └── chihuahua-muffin

train.py
EPOCHS = 50
BATCH_SIZE = 16
LEARNING_RATE = 0.001
PROJECT_NAME = 'Chihuahua-Muffin'
DATASET_NAME = 'chihuahua-muffin'

device = torch.device('cuda' if torch.cuda.is_available() else 'cpu')
print(f'Using device: {device}')

# MODEL DEFINITION
class ResNet18Classifier(nn.Module):
    def forward(self, x):
        ...

```

Epoch 45/50 - Val Loss: 0.1800, Val Accuracy: 93.24%
Epoch 46/50: 100%|██████████| 280/280 [00:56<00:00, 4.82it/s, loss=0.0232] | 74/74 [00:08<00:00, 8.78it/s, val_loss=0.00736]
Validation Epoch 46/50: 100%|██████████| 280/280 [00:56<00:00, 4.98it/s, loss=0.0344] | 74/74 [00:07<00:00, 9.59it/s, val_loss=0.00755]
Epoch 47/50 - Val Loss: 0.2024, Val Accuracy: 91.72%
Epoch 47/50: 100%|██████████| 280/280 [00:56<00:00, 4.98it/s, loss=0.0344] | 74/74 [00:07<00:00, 9.59it/s, val_loss=0.00755]
Epoch 48/50 - Val Loss: 0.1844, Val Accuracy: 92.91%
Epoch 48/50: 100%|██████████| 280/280 [00:56<00:00, 4.98it/s, loss=0.0161] | 74/74 [00:07<00:00, 9.77it/s, val_loss=0.0141]
Epoch 49/50 - Val Loss: 0.1844, Val Accuracy: 92.91%
Epoch 49/50: 100%|██████████| 280/280 [00:55<00:00, 5.00it/s, loss=0.0239] | 74/74 [00:08<00:00, 8.93it/s, val_loss=0.00895]
Epoch 49/50 - Val Loss: 0.1921, Val Accuracy: 92.48%
Epoch 50/50: 100%|██████████| 280/280 [00:56<00:00, 4.97it/s, loss=0.0295] | 74/74 [00:08<00:00, 8.71it/s, val_loss=0.00596]
Epoch 50/50 - Val Loss: 0.1898, Val Accuracy: 92.57%

Training complete!
Best validation accuracy: 95.01%

[OK] Loaded best model for metrics collection
[OK] Best model saved to 'resnet18_classifier_best.pth'
Collecting metrics on train set with best model...

