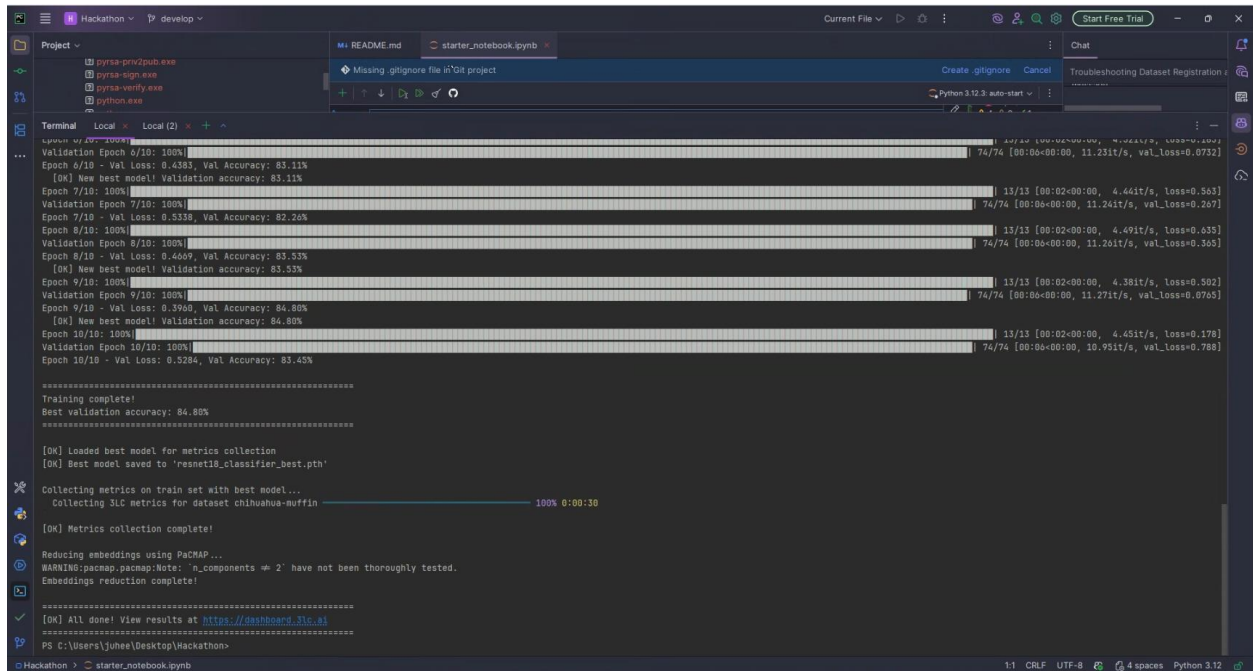


SCREENSHOTS:

Initial Accuracy:



```
Epoch 6/10: 100% | 74/74 [00:06:00:00, 11.231t/s, val_loss=0.0732]
Validation Epoch 6/10: 100% | 15/13 [00:02:00:00, 4.461t/s, loss=0.563]
Epoch 6/10 - Val Loss: 0.4383, Val Accuracy: 83.11%
[OK] New best model! Validation accuracy: 83.11%

Epoch 7/10: 100% | 74/74 [00:06:00:00, 11.241t/s, val_loss=0.267]
Validation Epoch 7/10: 100% | 15/13 [00:02:00:00, 4.491t/s, loss=0.635]
Epoch 7/10 - Val Loss: 0.5338, Val Accuracy: 82.26%

Epoch 8/10: 100% | 74/74 [00:06:00:00, 11.201t/s, val_loss=0.365]
Validation Epoch 8/10: 100% | 15/13 [00:02:00:00, 4.381t/s, loss=0.502]
Epoch 8/10 - Val Loss: 0.4669, Val Accuracy: 83.53%
[OK] New best model! Validation accuracy: 83.53%

Epoch 9/10: 100% | 74/74 [00:06:00:00, 11.271t/s, val_loss=0.0765]
Validation Epoch 9/10: 100% | 15/13 [00:02:00:00, 4.451t/s, loss=0.178]
Epoch 9/10 - Val Loss: 0.3968, Val Accuracy: 84.80%
[OK] New best model! Validation accuracy: 84.80%

Epoch 10/10: 100% | 74/74 [00:06:00:00, 10.951t/s, val_loss=0.788]
Validation Epoch 10/10: 100% | 15/13 [00:02:00:00, 4.451t/s, loss=0.178]
Epoch 10/10 - Val Loss: 0.5284, Val Accuracy: 83.45%

Training complete!
Best validation accuracy: 84.80%

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

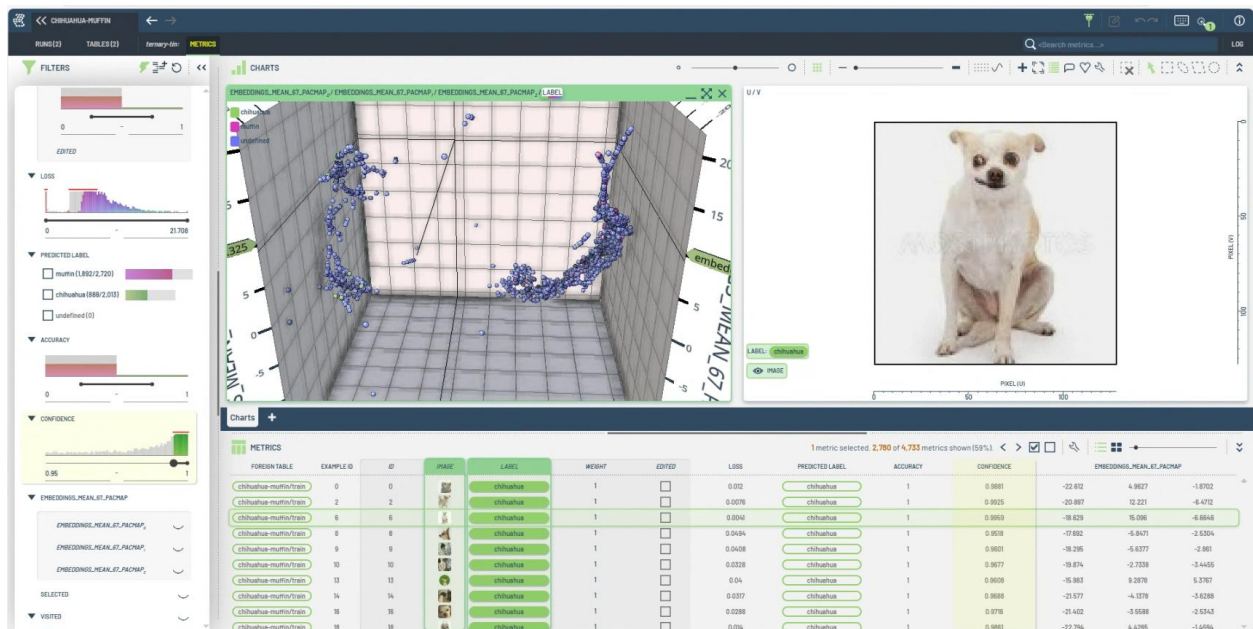
Collecting metrics on train set with best model...
Collecting 31c metrics for dataset chihuahua-muffin 100% 0:00:30

[OK] Metrics collection complete!

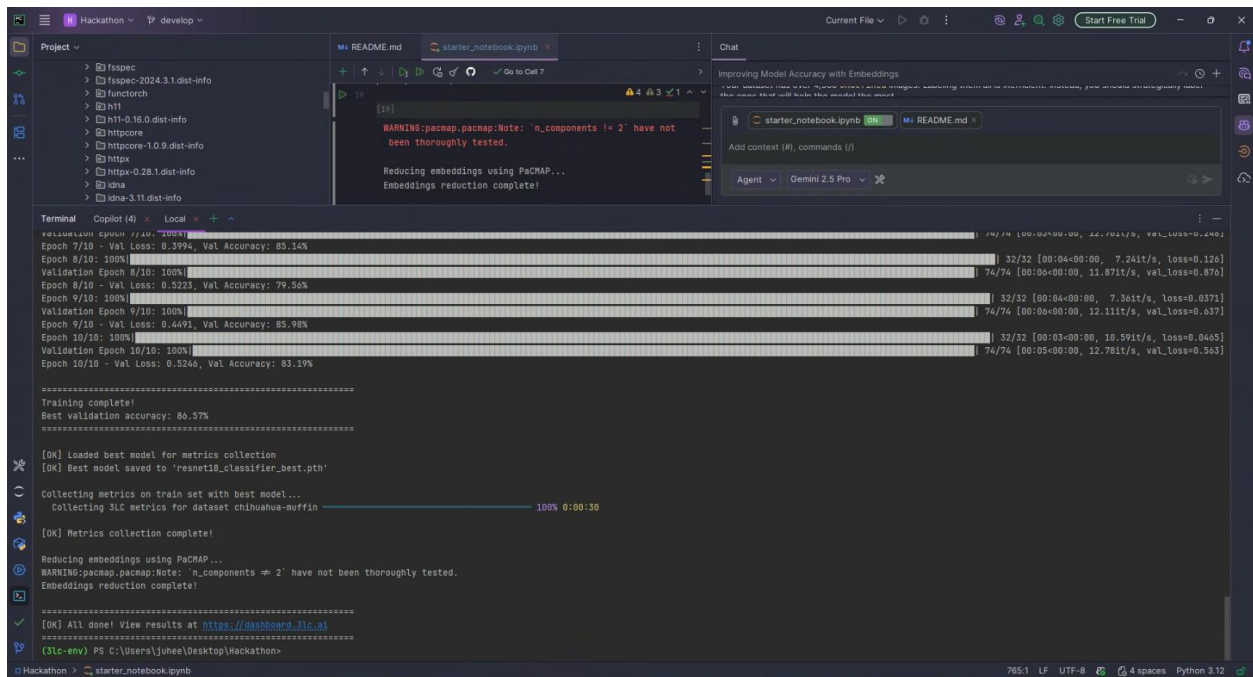
Reducing embeddings using PaCMAP...
WARNING:paomap.paomap:Note: 'n_components' ≠ 2' have not been thoroughly tested.
Embeddings reduction complete!

[OK] All done! View results at https://dashboards.31c.ai
PS C:\Users\j\h\h\Desktop\Hackathon>
```

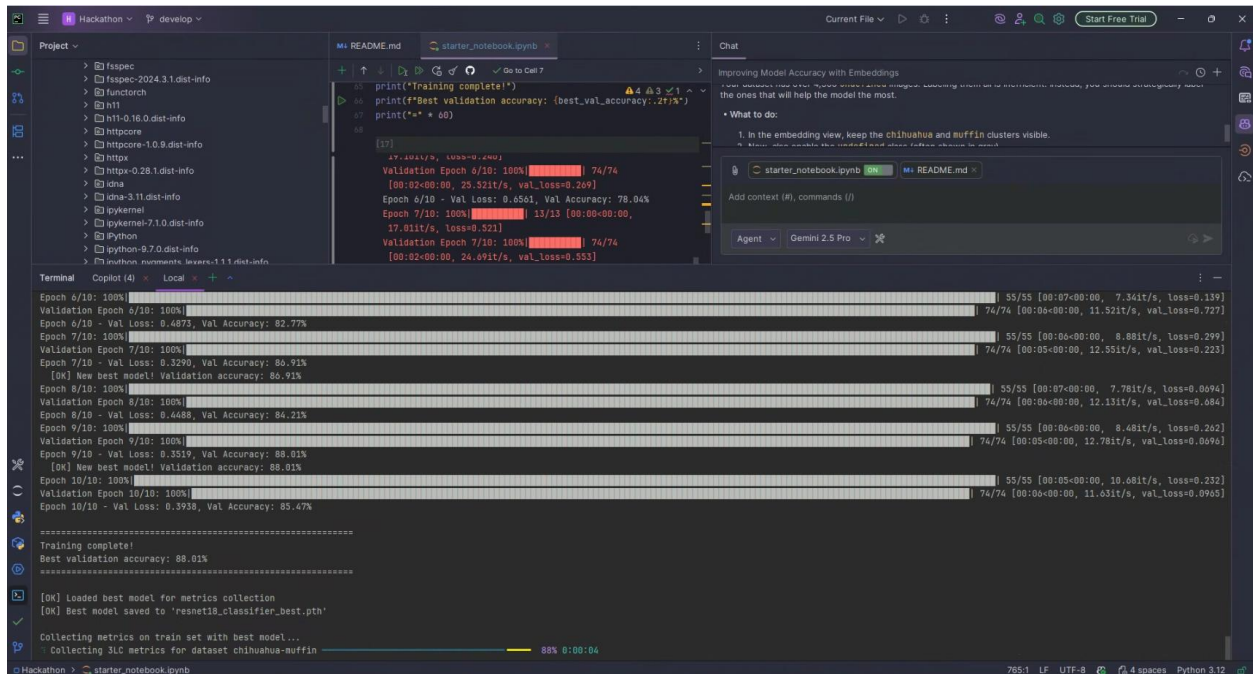
Version 1 - Confidence level above 0.95



Version 2 - Greater loss 21.708



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



Version 4 - labelled muffin with confidence more than .94

```
print("Training complete!")
print(f"Best validation accuracy: (best_val_accuracy:.2f)%")
print("*" * 60)

[17]
47.406179, Loss=0.4061
Validation Epoch 8/10: 100% | 74/74
[00:02:00:00, 25.52it/s, val_loss=0.269]
Epoch 8/10 - Val Loss: 0.4061, Val Accuracy: 86.15%

[Ok] New Best Model! Validation accuracy: 86.15%
Epoch 8/10: 100% | 74/74 [00:06:00:00, 8.79it/s, loss=0.0648]
Validation Epoch 9/10: 100% | 74/74 [00:05:00:00, 12.93it/s, val_loss=0.0389]
Epoch 9/10 - Val Loss: 0.3325, Val Accuracy: 88.68%
[Ok] New Best Model! Validation accuracy: 88.68%
Epoch 9/10: 100% | 74/74 [00:05:00:00, 10.69it/s, loss=2.1]
Validation Epoch 9/10: 100% | 74/74 [00:06:00:00, 11.76it/s, val_loss=0.101]
Epoch 9/10 - Val Loss: 0.3765, Val Accuracy: 86.32%
Epoch 10/10: 100% | 74/74 [00:06:00:00, 7.36it/s, loss=0.0476]
Validation Epoch 10/10: 100% | 74/74 [00:06:00:00, 11.51it/s, val_loss=0.459]
Epoch 10/10 - Val Loss: 0.3776, Val Accuracy: 85.64%

=====
Training complete!
Best validation accuracy: 88.68%
=====

[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 31C 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://dashboards.31c.ai
=====
(31c-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.

```
print("Training complete!")
print(f"Best validation accuracy: (best_val_accuracy:.2f)%")
print("*" * 60)

[17]
47.406179, Loss=0.4061
Validation Epoch 4/10: 100% | 74/74
[00:02:00:00, 25.52it/s, val_loss=0.269]
Epoch 4/10 - Val Loss: 0.4061, Val Accuracy: 89.82%

[Ok] New Best Model! Validation accuracy: 89.82%
Epoch 5/10: 100% | 69/69 [00:06:00:00, 10.43it/s, loss=0.329]
Validation Epoch 5/10: 100% | 74/74 [00:06:00:00, 11.51it/s, val_loss=0.175]
Epoch 5/10 - Val Loss: 0.3455, Val Accuracy: 86.15%
Epoch 6/10: 100% | 69/69 [00:06:00:00, 10.53it/s, loss=0.369]
Validation Epoch 6/10: 100% | 74/74 [00:06:00:00, 11.94it/s, val_loss=0.0834]
Epoch 6/10 - Val Loss: 0.3079, Val Accuracy: 89.10%
[Ok] New Best Model! Validation accuracy: 89.10%
Epoch 7/10: 100% | 69/69 [00:06:00:00, 10.53it/s, loss=0.842]
Validation Epoch 7/10: 100% | 74/74 [00:06:00:00, 10.53it/s, val_loss=0.18]
Epoch 7/10 - Val Loss: 0.3092, Val Accuracy: 88.51%
Epoch 8/10: 100% | 69/69 [00:06:00:00, 10.52it/s, loss=0.397]
Validation Epoch 8/10: 100% | 74/74 [00:06:00:00, 11.36it/s, val_loss=0.0702]
Epoch 8/10 - Val Loss: 0.3074, Val Accuracy: 88.43%
Epoch 9/10: 100% | 69/69 [00:06:00:00, 10.53it/s, loss=0.0624]
Validation Epoch 9/10: 100% | 74/74 [00:06:00:00, 12.25it/s, val_loss=0.158]
Epoch 9/10 - Val Loss: 0.3176, Val Accuracy: 88.85%
Epoch 10/10: 100% | 69/69 [00:06:00:00, 7.48it/s, loss=0.247]
Validation Epoch 10/10: 100% | 74/74 [00:06:00:00, 11.51it/s, val_loss=0.551]
Epoch 10/10 - Val Loss: 0.4202, Val Accuracy: 85.85%

=====
Training complete!
Best validation accuracy: 89.10%
=====

[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 31C 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.


```
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: 0.3986, 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 3LC metrics for dataset chihuahua-muffin 100% 0:00:29

[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://dashboards.3lc.ai
=====
(3LC-env) PS C:\Users\junhee\Desktop\Hackathon>
```

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

```
Validation Epoch 4/10: 100%
Epoch 4/10 - Val Loss: 0.3383, 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: 0.3035, 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: 0.5730, 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: 86.46%

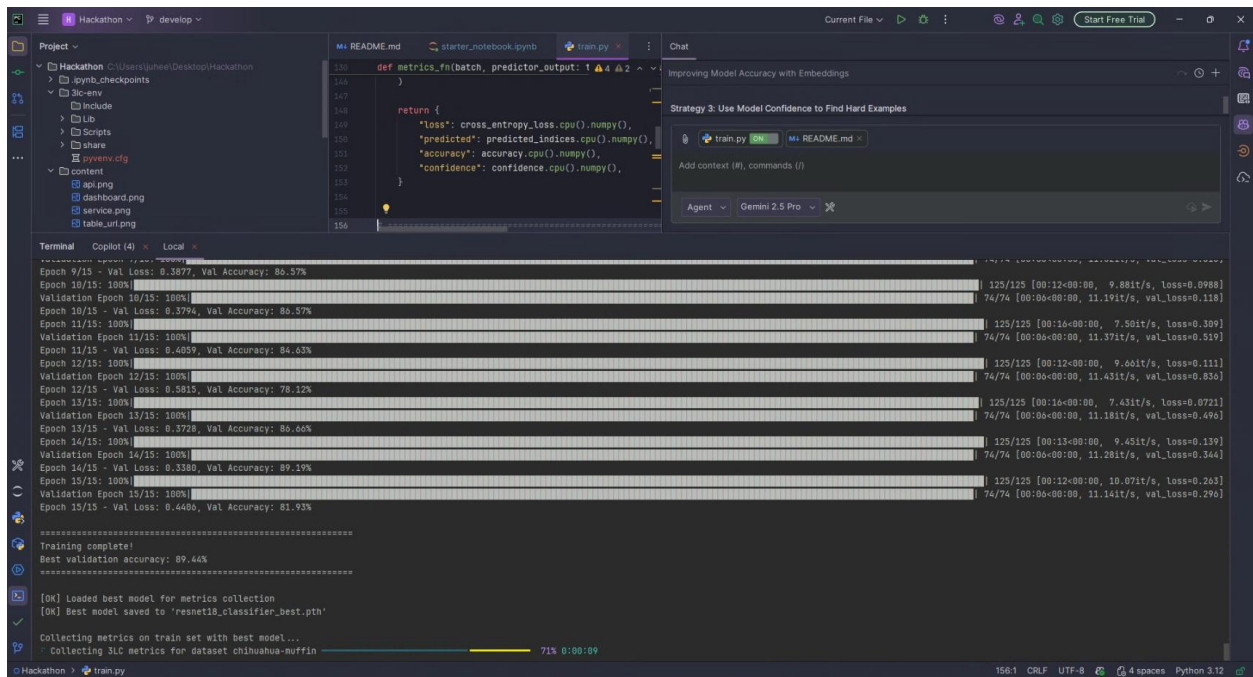
=====
Training complete!
Best validation accuracy: 89.27%
=====

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

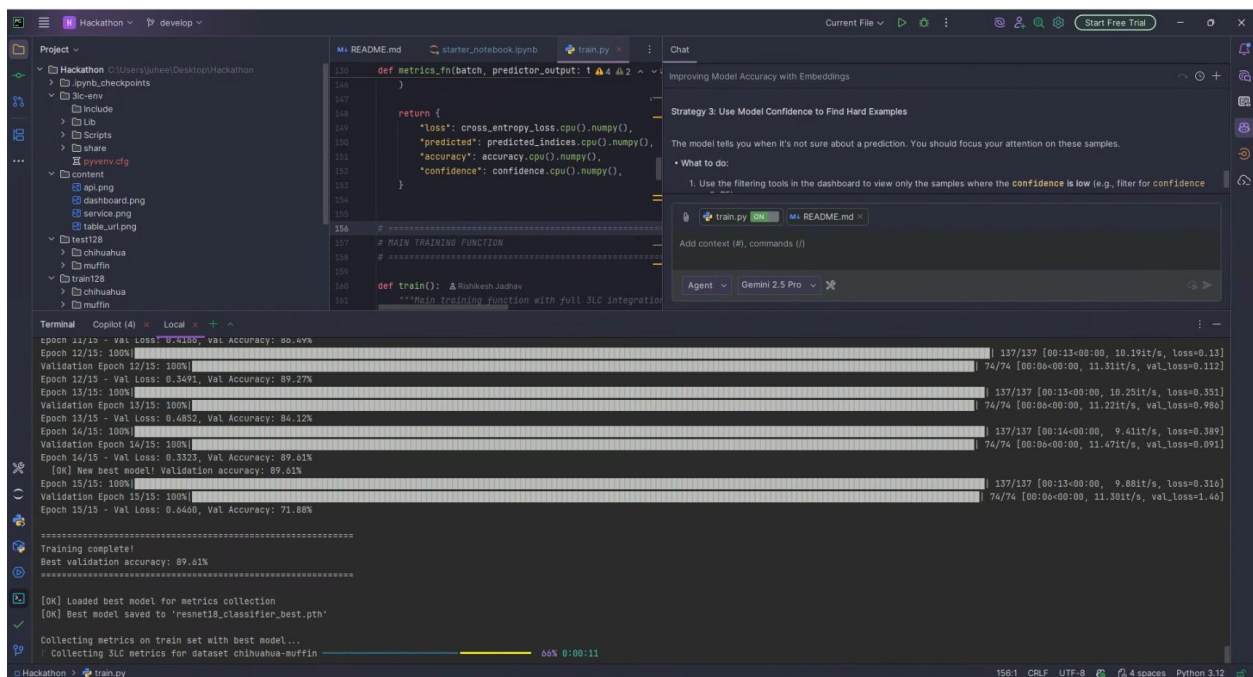
Collecting metrics on train set with best model...
Collecting 3LC metrics for dataset chihuahua-muffin 59% 0:00:13

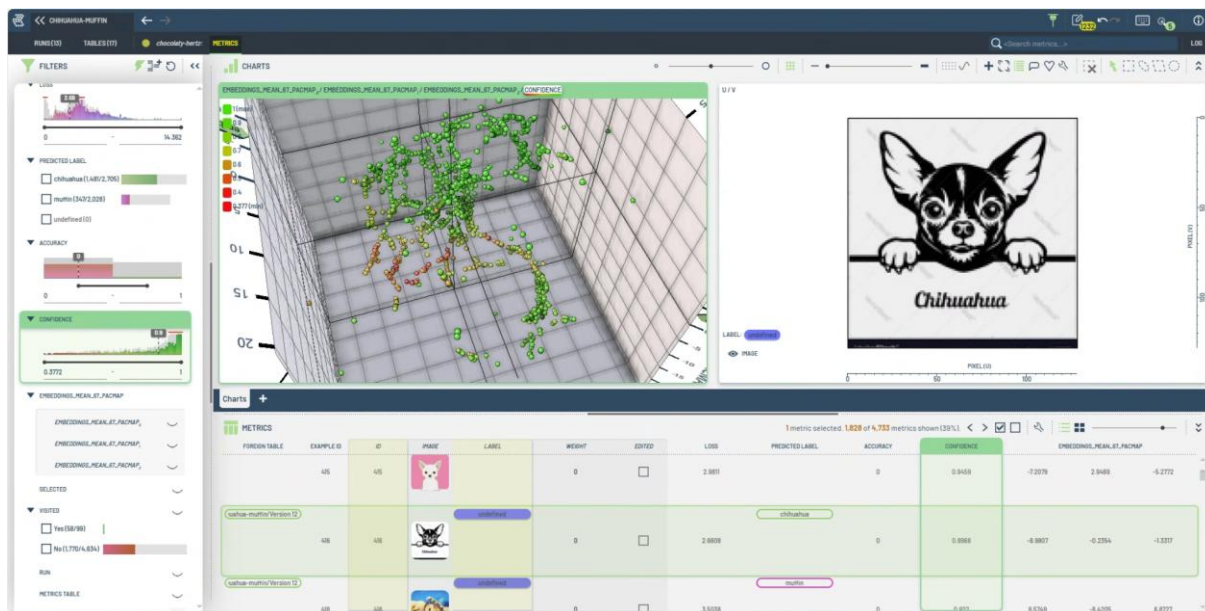
Start
```

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.



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.





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.

```

1 #!/usr/bin/env python
2 import sys
3 import os
4 import numpy as np
5 import tensorflow as tf
6 import keras
7
8 # Training hyperparameters
9 EPOCHS = 10
10 BATCH_SIZE = 16
11 LEARNING_RATE = 0.0001
12
13 # Project configuration
14 PROJECT_NAME = "Chihuahua-Muffin"
15 DATASET_NAME = "chihuahua-muffin"
16
17 # Training script
18 def train():
19     # Load data
20     # Train model
21     # Evaluate model
22
23 if __name__ == '__main__':
24     train()
  
```

Validation Epoch 4/10: 100% | 74/74 [00:06<00:00, 10.87it/s, val_loss=0.105]
 Epoch 6/10 - Val Loss: 0.3175, Val Accuracy: 90.62%
 [OK] New best model! Validation accuracy: 90.62%
 Epoch 7/10: 100% | 143/143 [00:14<00:00, 9.98it/s, loss=0.195]
 Validation Epoch 7/10: 100% | 74/74 [00:06<00:00, 11.41it/s, val_loss=0.338]
 Epoch 7/10 - Val Loss: 0.3174, Val Accuracy: 91.39%
 [OK] New best model! Validation accuracy: 91.39%
 Epoch 8/10: 100% | 143/143 [00:15<00:00, 9.34it/s, loss=1.66]
 Validation Epoch 8/10: 100% | 74/74 [00:06<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 [00:17<00:00, 8.12it/s, loss=0.343]
 Validation Epoch 9/10: 100% | 74/74 [00:05<00:00, 12.38it/s, val_loss=0.294]
 Epoch 9/10 - Val Loss: 0.3225, Val Accuracy: 90.83%
 Epoch 10/10: 100% | 143/143 [00:14<00:00, 9.79it/s, loss=0.0852]
 Validation Epoch 10/10: 100% | 74/74 [00:06<00:00, 10.64it/s, val_loss=0.23]
 Epoch 10/10 - Val Loss: 0.3132, Val Accuracy: 90.62%

Training complete!
 Best validation accuracy: 91.39%

[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 31C metrics for dataset chihuahua-muffin 71% 0:00:09

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):
        # Load ResNet18 without pretrained weights
        self.resnet = models.resnet18(weights=None)

        # Set 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(out_features=256, out_features=128),
            nn.ReLU(),
            nn.Dropout(0.3),
            nn.Linear(out_features=128, num_classes)
        )

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

```

```

Epoch 9/10: 100% | 148/148 [00:15:00:00, 9.50it/s, loss=1.53]
Validation Epoch 9/10: 100% | 74/74 [00:00:00:00, 11.89it/s, val_loss=0.0801]
Epoch 9/10 - Val Loss: 0.2884, Val Accuracy: 90.71%
[OK] New best model! Validation accuracy: 90.71%
Epoch 10/10: 100% | 148/148 [00:14:00:00, 10.20it/s, loss=1.53]
Validation Epoch 10/10: 100% | 74/74 [00:00:00:00, 11.41it/s, val_loss=0.0618]
Epoch 10/10 - Val Loss: 0.3604, Val Accuracy: 88.68%

Training complete!
Best validation accuracy: 90.71%

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

Collecting metrics on train set with best model...

```

EPOCHS = 20 and LEARNING_RATE = 0.001

```

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")
print(f"Using device: {device}")

```

```

Validation Epoch 20/20: 100% | 74/74 [00:00:00:00, 11.28it/s, val_loss=0.195]
Epoch 20/20 - Val Loss: 0.2876, 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 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.3lc.ai

(3lc-env) PS C:\Users\jshel\Desktop\Hackathon>

```

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.

```

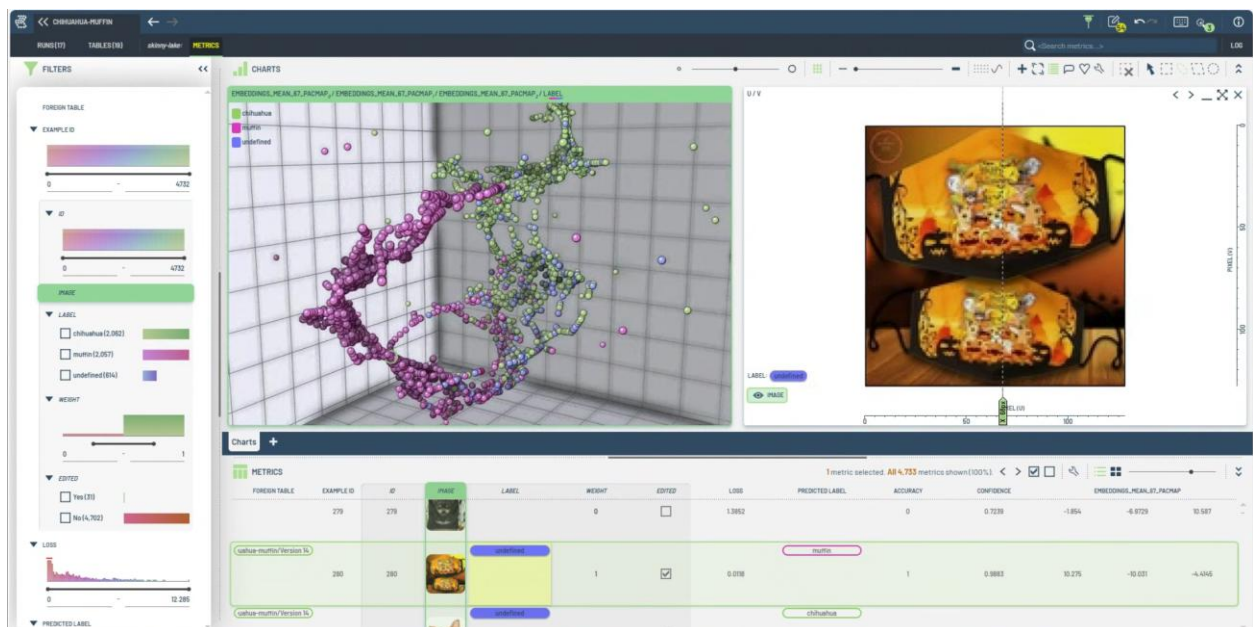
25 #!python
26 python train.py
27 ...
28
29 # Training hyperparameters
30 EPOCHS = 20
31 BATCH_SIZE = 16
32 LEARNING_RATE = 0.001
33
34 # Project configuration
35 PROJECT_NAME = "Chihuahua-Muffin"
36 DATASET_NAME = "chihuahua-muffin"

```

```

Epoch 16/20: 100% | 156/156 [00:21:00:00, 7.42it/s, loss=0.388]
Validation Epoch 16/20: 100% | 74/74 [00:05:00:00, 12.46it/s, val_loss=0.127]
Epoch 16/20 - Val Loss: 0.3383, Val Accuracy: 90.90%
Epoch 17/20: 100% | 156/156 [00:23:00:00, 6.55it/s, loss=0.593]
Validation Epoch 17/20: 100% | 74/74 [00:08:00:00, 9.12it/s, val_loss=0.328]
Epoch 17/20 - Val Loss: 0.4653, Val Accuracy: 85.50%
Epoch 18/20: 100% | 156/156 [00:22:00:00, 7.00it/s, loss=0.344]
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, 7.03it/s, loss=2.11]
Validation Epoch 19/20: 100% | 74/74 [00:08:00:00, 9.23it/s, val_loss=0.498]
Epoch 19/20 - Val Loss: 0.3390, Val Accuracy: 90.88%
Epoch 20/20: 100% | 156/156 [00:23:00:00, 6.59it/s, loss=0.656]
Validation Epoch 20/20: 100% | 74/74 [00:08:00:00, 8.78it/s, val_loss=0.145]
Epoch 20/20 - Val Loss: 0.3517, 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 316 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


```
25 Usage:
26 python train.py
27 ***
28
29 > import ...
30
31 # Training hyperparameters
32 EPOCHS = 20
33 BATCH_SIZE = 16
34 LEARNING_RATE = 0.001
35
36 # Project configuration
37 PROJECT_NAME = "Chihuahua-Muffin"
38 DATASET_NAME = "chihuahua-muffin"
39
40 device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
```

Epoch 18/20: 100%
Validation Epoch 18/20: 100%
Epoch 18/20 - Val Loss: 0.2965, Val Accuracy: 92.31%
[OK] New best model! Validation accuracy: 92.31%

Epoch 19/20: 100%
Validation Epoch 19/20: 100%
Epoch 19/20 - Val Loss: 0.2471, Val Accuracy: 92.31%

Epoch 20/20: 100%
Validation Epoch 20/20: 100%
Epoch 20/20 - Val Loss: 0.3176, 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.

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

```
60 return self.classifier(resnet_features)
61
62 # DATA TRANSFORMS
63 # =====
64
65 train_transform = transforms.Compose([
66     transforms.Resize(128),
67     transforms.RandomCrop(128),
68     transforms.RandomHorizontalFlip(),
69     transforms.RandomRotation(15),
70     transforms.ColorJitter(brightness=0.2, contrast=0.2, saturation=0.2, hue=0.1),
71     transforms.RandomAffine(degrees=0, shear=10, scale=(0.8, 1.2)),
72     transforms.ToTensor(),
73     transforms.Normalize(mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225]),
74 ])
75
76 val_transform = transforms.Compose([
77     transforms.Resize(128),
78     transforms.RandomCrop(128),
79     transforms.RandomHorizontalFlip(),
80     transforms.RandomRotation(15),
81     transforms.ColorJitter(brightness=0.2, contrast=0.2, saturation=0.2, hue=0.1),
82     transforms.RandomAffine(degrees=0, shear=10, scale=(0.8, 1.2)),
83     transforms.ToTensor(),
84     transforms.Normalize(mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225]),
85 ])
```

Epoch 17/20 - Val Loss: 0.2988, Val Accuracy: 91.05%
[OK] New best model! Validation accuracy: 91.05%

Epoch 18/20: 100%
Validation Epoch 18/20: 100%
Epoch 18/20 - Val Loss: 0.3979, Val Accuracy: 89.53%

Epoch 19/20: 100%
Validation Epoch 19/20: 100%
Epoch 19/20 - Val Loss: 0.2735, Val Accuracy: 92.06%
[OK] New best model! Validation accuracy: 92.06%

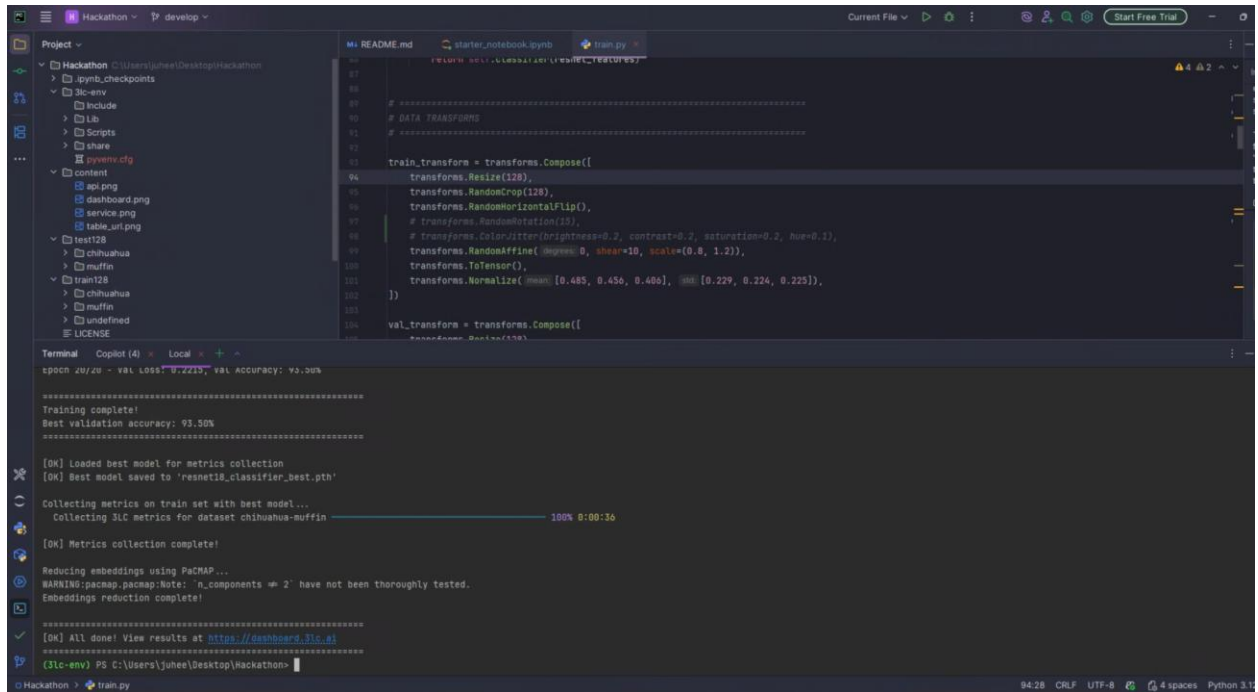
Epoch 20/20: 100%
Validation Epoch 20/20: 100%
Epoch 20/20 - Val Loss: 0.2705, Val Accuracy: 91.47%

Training complete!
Best validation accuracy: 92.06%

[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 43% 0:00:19

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



The screenshot shows a VS Code editor with a project named 'Hackathon'. The file explorer on the left shows a directory structure with files like 'api.png', 'dashboard.png', 'service.png', 'task_01.png', 'test128', 'chihuahua', 'muffin', 'train128', 'chihuahua', 'muffin', 'undefined', and 'LICENSE'. The main editor window shows a Python script 'train.py' with the following code:

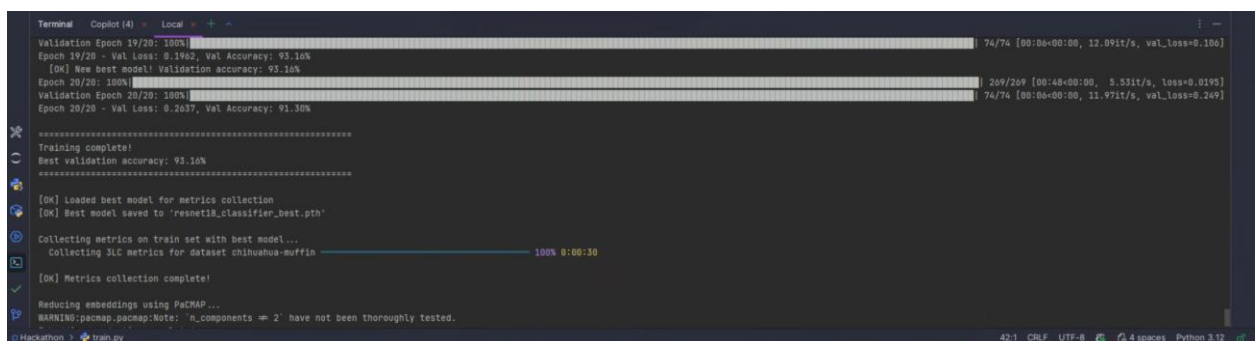
```
def return_def_labels(selected_labels, features):  
    # .....  
    # DATA TRANSFORMS  
    # .....  
    train_transform = transforms.Compose([  
        transforms.Resize(328),  
        transforms.RandomCrop(328),  
        transforms.RandomHorizontalFlip(),  
        # transforms.RandomRotation(15),  
        # 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(328),  
        transforms.CenterCrop(328),  
    ])
```

The terminal output shows the training progress:

```
epoch 20/20 - val_loss: 0.2215, val Accuracy: 93.30%  
.....  
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 31C 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://anphoars.bio.si  
.....  
(31c-env) PS C:\Users\june\Desktop\Hackathon>
```

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

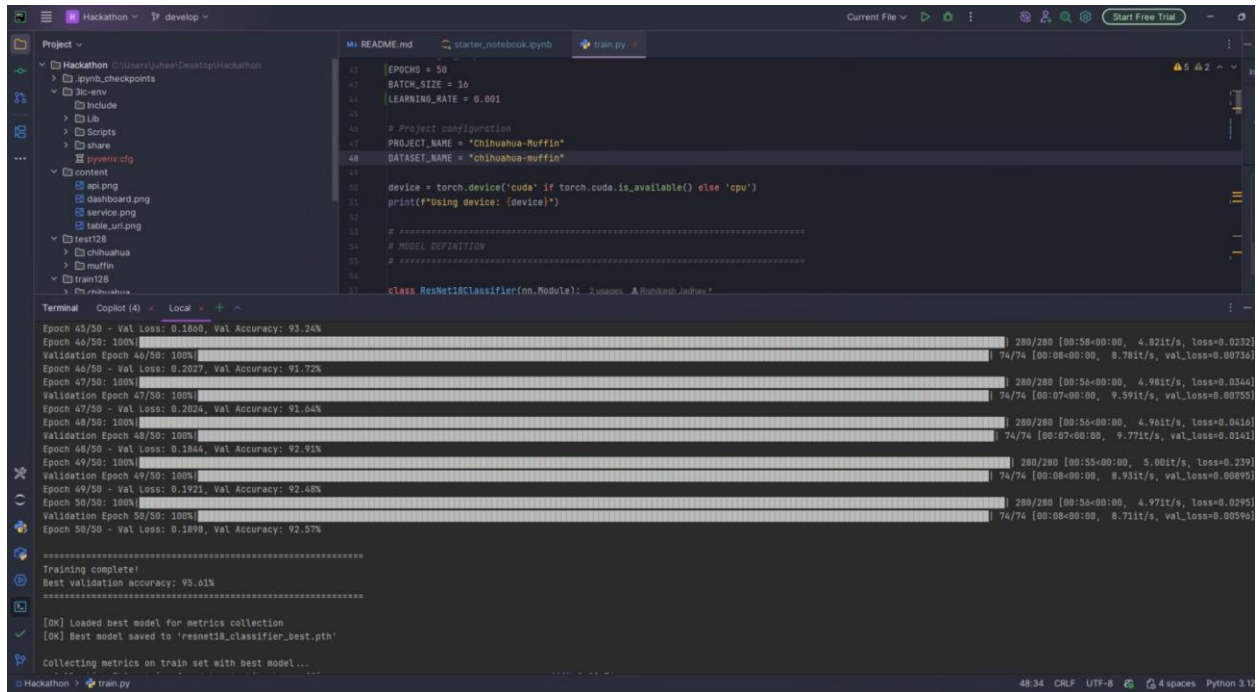
Dynamic Learning Rate: ReduceLROnPlateau scheduler. This will help the model train more effectively.



The screenshot shows a VS Code terminal with the following output:

```
Validation Epoch 19/20: 100% 74/74 [00:06:00:00, 12.091t/s, val_loss=0.186]  
Epoch 19/20 - Val Loss: 0.1962, Val Accuracy: 93.16%  
[OK] New best model! Validation accuracy: 93.16%  
Epoch 20/20: 100% 209/209 [00:48:00:00, 5.531t/s, loss=0.0195]  
Validation Epoch 20/20: 100% 74/74 [00:06:00:00, 11.971t/s, val_loss=0.269]  
Epoch 20/20 - Val Loss: 0.2637, Val Accuracy: 91.30%  
.....  
Training complete!  
Best validation accuracy: 93.16%  
.....  
[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 31C 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.
```

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



```
42 EPOCHS = 50
43 BATCH_SIZE = 16
44 LEARNING_RATE = 0.001
45
46 # Project configuration
47 PROJECT_NAME = "Chihuahua-Muffin"
48 DATASET_NAME = "chihuahua-muffin"
49
50 device = torch.device("cuda" if torch.cuda.is_available() else "cpu")
51 print(f"Using device: {device}")
52
53 # =====
54 # MODEL DEFINITION
55 # =====
56
57 class ResNet18Classifier(nn.Module):
58     """ResNet18 Classifier"""
```

Epoch 45/50 - Val Loss: 0.1800, Val Accuracy: 93.24%
Epoch 46/50: 100% | 280/280 [00:58<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 [00:58<00:00, 4.82it/s, loss=0.0232] | 74/74 [00:08<00:00, 8.78it/s, val_loss=0.00736]
Epoch 47/50 - Val Loss: 0.2027, Val Accuracy: 91.72%
Epoch 47/50: 100% | 280/280 [00:58<00:00, 4.82it/s, loss=0.0344] | 74/74 [00:07<00:00, 9.59it/s, val_loss=0.00755]
Validation Epoch 47/50: 100% | 280/280 [00:58<00:00, 4.82it/s, loss=0.0344] | 74/74 [00:07<00:00, 9.59it/s, val_loss=0.00755]
Epoch 48/50 - Val Loss: 0.2024, Val Accuracy: 91.64%
Epoch 48/50: 100% | 280/280 [00:58<00:00, 4.82it/s, loss=0.0410] | 74/74 [00:07<00:00, 9.77it/s, val_loss=0.0141]
Validation Epoch 48/50: 100% | 280/280 [00:58<00:00, 4.82it/s, loss=0.0410] | 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, 5.80it/s, loss=0.239] | 74/74 [00:08<00:00, 8.93it/s, val_loss=0.00895]
Validation Epoch 49/50: 100% | 280/280 [00:55<00:00, 5.80it/s, loss=0.239] | 74/74 [00:08<00:00, 8.93it/s, val_loss=0.00895]
Epoch 50/50: 100% | 280/280 [00:58<00:00, 4.97it/s, loss=0.0295] | 74/74 [00:08<00:00, 8.71it/s, val_loss=0.00595]
Validation Epoch 50/50: 100% | 280/280 [00:58<00:00, 4.97it/s, loss=0.0295] | 74/74 [00:08<00:00, 8.71it/s, val_loss=0.00595]
Epoch 50/50 - Val Loss: 0.1898, Val Accuracy: 92.37%

Training complete!
Best validation accuracy: 95.61%

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

