

toenhace

December 6, 2025

```
[43]: import sys
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import pickle
import torch
import torch.nn as nn
import torch.optim as optim
from torchvision import datasets, transforms, models
from torch.utils.data import DataLoader, random_split, Dataset
import os
from pyclass import TransformSubset, TinyCNN
from PIL import Image
import random
from sklearn.model_selection import train_test_split
from torch.utils.data import Subset
from torch.utils.data import WeightedRandomSampler
from sklearn.metrics import accuracy_score, recall_score,
    balanced_accuracy_score, classification_report, confusion_matrix
import torch
import numpy as np
```

```
[44]: SEED = 42

random.seed(SEED)
np.random.seed(SEED)
torch.manual_seed(SEED)
torch.cuda.manual_seed(SEED)
torch.cuda.manual_seed_all(SEED)
torch.backends.cudnn.benchmark = False
torch.backends.cudnn.deterministic = True
```

```
[45]: class PLKDataset(Dataset):
    def __init__(self, file_path, transform=None):
        with open(file_path, 'rb') as f:
            data = pickle.load(f)
        self.images = data['images']
        self.labels = data['labels'].reshape(-1)
```

```

        self.transform = transform

    def __len__(self):
        return len(self.images)

    def __getitem__(self, idx):
        image = self.images[idx]
        label = int(self.labels[idx])

        image = Image.fromarray(image.astype('uint8'))

        if self.transform:
            image = self.transform(image)

        return image, label

```

```
[46]: if torch.backends.mps.is_available():
    device = torch.device("mps")
    use_mps = True
elif torch.cuda.is_available():
    device = torch.device("cpu")
    use_mps = False

print(device)
```

mps

```
[47]: dataset = PLKDataset('ift-3395-6390-kaggle-2-competition-fall-2025/train_data.
                           ↴pkl')
```

```
[48]: raw_images, raw_labels = dataset.images, dataset.labels
```

```
[49]: loader = DataLoader(dataset, batch_size=32, shuffle=True)
```

```
[50]: labels = dataset.labels
idx = np.arange(len(dataset))
train_idx, valid_idx = train_test_split(idx, test_size=0.2, stratify=labels, ↴
                                         random_state=42)
```

```
[51]: X_train_raw = dataset.images[train_idx]

X_train_float = X_train_raw.astype(np.float32) / 255.0

means = X_train_float.mean(axis=(0,1,2))
std = X_train_float.std(axis=(0,1,2))
```

```

IR_MEAN = means.tolist()
IR_STD = std.tolist()

print("IR_MEAN=", IR_MEAN)
print("IR_STD=", IR_STD)

```

```

IR_MEAN= [0.2111530601978302, 0.005323430523276329, 0.22947929799556732]
IR_STD= [0.18979966640472412, 0.016638699918985367, 0.16980312764644623]

```

```

[52]: train_transform = transforms.Compose([
    # --- Transformations Géométriques ---
    transforms.RandomHorizontalFlip(p=0.5),
    transforms.RandomVerticalFlip(p=0.5),
    transforms.RandomRotation(180),

    transforms.RandomAffine(degrees=0, translate=(0.05, 0.05), scale=(0.9, 1.
    ↵1)),
    transforms.ColorJitter(brightness=0.2, contrast=0.2),

    transforms.ToTensor(),
    transforms.Normalize(mean=IR_MEAN, std=IR_STD)
])

# Le val_transform reste inchangé (correct)
val_transform = transforms.Compose([
    transforms.ToTensor(),
    transforms.Normalize(mean=IR_MEAN, std=IR_STD)
])

```

```

[53]: train_dataset = Subset(dataset, train_idx)
train_data = TransformSubset(train_dataset, train_transform)

val_dataset = Subset(dataset, valid_idx)
val_data = TransformSubset(val_dataset, val_transform)

y_train_idx = train_dataset.indices
all_labels = dataset.labels[y_train_idx]
class_count = np.bincount(all_labels.astype(int))
weights_class = 1. / np.sqrt(class_count)
sample_weights = [weights_class[int(label)] for label in all_labels]
sample_weights = torch.from_numpy(np.array(sample_weights)).double()

```

```
sampler = WeightedRandomSampler(weights=sample_weights,  
    ↪num_samples=len(sample_weights), replacement=True)  
  
train_loader = DataLoader(train_data, batch_size=64, sampler=sampler,  
    ↪shuffle=False, pin_memory=True, drop_last=True)  
val_loader = DataLoader(val_data, batch_size=64, shuffle=False,  
    ↪pin_memory=True, drop_last=True)
```

```
[54]: model = TinyCNN().to(device)
```

```
[55]: from torch.optim import AdamW  
from torch.optim.lr_scheduler import CosineAnnealingLR  
  
criterion = nn.CrossEntropyLoss(label_smoothing=0.1)  
  
optimizer = optim.AdamW(model.parameters(),  
    lr=1e-3,  
    weight_decay=1e-2  
)  
  
scheduler = CosineAnnealingLR(optimizer, T_max = 50, eta_min =1e-6)
```

```
[56]: best_val_loss = float('inf')  
  
for epoch in range(20):  
    model.train()  
    running_train_loss = 0  
  
    for images, labels in train_loader:  
        images = images.to(device)  
        labels = labels.to(device)  
  
        optimizer.zero_grad()  
  
        outputs = model(images)  
        loss = criterion(outputs, labels)  
  
        loss.backward()  
        optimizer.step()  
  
        running_train_loss += loss.item()  
  
    avg_train_loss = running_train_loss / len(train_loader)  
    print(f"Epoch {epoch+1}, Train Loss = {avg_train_loss:.4f}")  
  
    scheduler.step()
```

```

model.eval()
preds, gts = [], []
running_val_loss = 0

with torch.no_grad():
    for images, labels in val_loader:
        images = images.to(device)
        labels = labels.to(device)

        outputs = model(images)

        val_loss = criterion(outputs, labels)
        running_val_loss += val_loss.item()

        _, pred = torch.max(outputs, 1)

        preds.extend(pred.cpu().numpy())
        gts.extend(labels.cpu().numpy())

avg_val_loss = running_val_loss / len(val_loader)

# Calcul des métriques
all_preds = np.array(preds)
all_labels = np.array(gts)

bal_acc = balanced_accuracy_score(all_labels, all_preds)

print(f"Val Loss = {avg_val_loss:.4f}, Bal Acc = {bal_acc:.4f}")

if avg_val_loss < best_val_loss:
    best_val_loss = avg_val_loss
    torch.save(model.state_dict(), 'best_model_cnn_ir.pth')
    patience_counter = 0
else:
    patience_counter += 1
    if patience_counter >= 10:
        print("Arrêt précoce")
        break

```

/Users/yamira.poldosilva/Documents/UDEM/A25/IFT3395/kaggle2/kaggle2/lib/python3.13/site-packages/torch/utils/data/dataloader.py:692: UserWarning: 'pin_memory' argument is set as true but not supported on MPS now, device pinned memory won't be used.

```
    warnings.warn(warn_msg)
```

Epoch 1, Train Loss = 1.5040

```
Val Loss = 1.4986, Bal Acc = 0.2757
/Users/yamira.poldosilva/Documents/UDEM/A25/IFT3395/kaggle2/kaggle2/lib/python3.
13/site-packages/torch/utils/data/dataloader.py:692: UserWarning: 'pin_memory'
argument is set as true but not supported on MPS now, device pinned memory won't
be used.
    warnings.warn(warn_msg)

Epoch 2, Train Loss = 1.5117
Val Loss = 1.5118, Bal Acc = 0.2994
/Users/yamira.poldosilva/Documents/UDEM/A25/IFT3395/kaggle2/kaggle2/lib/python3.
13/site-packages/torch/utils/data/dataloader.py:692: UserWarning: 'pin_memory'
argument is set as true but not supported on MPS now, device pinned memory won't
be used.
    warnings.warn(warn_msg)

Epoch 3, Train Loss = 1.4795
Val Loss = 1.3757, Bal Acc = 0.3140
/Users/yamira.poldosilva/Documents/UDEM/A25/IFT3395/kaggle2/kaggle2/lib/python3.
13/site-packages/torch/utils/data/dataloader.py:692: UserWarning: 'pin_memory'
argument is set as true but not supported on MPS now, device pinned memory won't
be used.
    warnings.warn(warn_msg)

Epoch 4, Train Loss = 1.4909
Val Loss = 1.5092, Bal Acc = 0.3367
/Users/yamira.poldosilva/Documents/UDEM/A25/IFT3395/kaggle2/kaggle2/lib/python3.
13/site-packages/torch/utils/data/dataloader.py:692: UserWarning: 'pin_memory'
argument is set as true but not supported on MPS now, device pinned memory won't
be used.
    warnings.warn(warn_msg)

Epoch 5, Train Loss = 1.4604
Val Loss = 1.3942, Bal Acc = 0.3173
/Users/yamira.poldosilva/Documents/UDEM/A25/IFT3395/kaggle2/kaggle2/lib/python3.
13/site-packages/torch/utils/data/dataloader.py:692: UserWarning: 'pin_memory'
argument is set as true but not supported on MPS now, device pinned memory won't
be used.
    warnings.warn(warn_msg)

Epoch 6, Train Loss = 1.4326
Val Loss = 1.4098, Bal Acc = 0.2844
/Users/yamira.poldosilva/Documents/UDEM/A25/IFT3395/kaggle2/kaggle2/lib/python3.
13/site-packages/torch/utils/data/dataloader.py:692: UserWarning: 'pin_memory'
argument is set as true but not supported on MPS now, device pinned memory won't
be used.
    warnings.warn(warn_msg)
```

```
Epoch 7, Train Loss = 1.4613
Val Loss = 1.4173, Bal Acc = 0.3282

/Users/yamira.poldosilva/Documents/UDEM/A25/IFT3395/kaggle2/kaggle2/lib/python3.
13/site-packages/torch/utils/data/dataloader.py:692: UserWarning: 'pin_memory'
argument is set as true but not supported on MPS now, device pinned memory won't
be used.

    warnings.warn(warn_msg)

Epoch 8, Train Loss = 1.4867
Val Loss = 1.4014, Bal Acc = 0.3252

/Users/yamira.poldosilva/Documents/UDEM/A25/IFT3395/kaggle2/kaggle2/lib/python3.
13/site-packages/torch/utils/data/dataloader.py:692: UserWarning: 'pin_memory'
argument is set as true but not supported on MPS now, device pinned memory won't
be used.

    warnings.warn(warn_msg)

Epoch 9, Train Loss = 1.4686
Val Loss = 1.3949, Bal Acc = 0.3696

/Users/yamira.poldosilva/Documents/UDEM/A25/IFT3395/kaggle2/kaggle2/lib/python3.
13/site-packages/torch/utils/data/dataloader.py:692: UserWarning: 'pin_memory'
argument is set as true but not supported on MPS now, device pinned memory won't
be used.

    warnings.warn(warn_msg)

Epoch 10, Train Loss = 1.4585
Val Loss = 1.3826, Bal Acc = 0.3146

/Users/yamira.poldosilva/Documents/UDEM/A25/IFT3395/kaggle2/kaggle2/lib/python3.
13/site-packages/torch/utils/data/dataloader.py:692: UserWarning: 'pin_memory'
argument is set as true but not supported on MPS now, device pinned memory won't
be used.

    warnings.warn(warn_msg)

Epoch 11, Train Loss = 1.4316
Val Loss = 1.4450, Bal Acc = 0.3281

/Users/yamira.poldosilva/Documents/UDEM/A25/IFT3395/kaggle2/kaggle2/lib/python3.
13/site-packages/torch/utils/data/dataloader.py:692: UserWarning: 'pin_memory'
argument is set as true but not supported on MPS now, device pinned memory won't
be used.

    warnings.warn(warn_msg)

Epoch 12, Train Loss = 1.4400
Val Loss = 1.4878, Bal Acc = 0.2789

/Users/yamira.poldosilva/Documents/UDEM/A25/IFT3395/kaggle2/kaggle2/lib/python3.
13/site-packages/torch/utils/data/dataloader.py:692: UserWarning: 'pin_memory'
argument is set as true but not supported on MPS now, device pinned memory won't
be used.

    warnings.warn(warn_msg)
```

```
Epoch 13, Train Loss = 1.4490
Val Loss = 1.3521, Bal Acc = 0.3462

/Users/yamira.poldosilva/Documents/UDEM/A25/IFT3395/kaggle2/kaggle2/lib/python3.
13/site-packages/torch/utils/data/dataloader.py:692: UserWarning: 'pin_memory'
argument is set as true but not supported on MPS now, device pinned memory won't
be used.

    warnings.warn(warn_msg)

Epoch 14, Train Loss = 1.4186
Val Loss = 1.3684, Bal Acc = 0.3575

/Users/yamira.poldosilva/Documents/UDEM/A25/IFT3395/kaggle2/kaggle2/lib/python3.
13/site-packages/torch/utils/data/dataloader.py:692: UserWarning: 'pin_memory'
argument is set as true but not supported on MPS now, device pinned memory won't
be used.

    warnings.warn(warn_msg)

Epoch 15, Train Loss = 1.4327
Val Loss = 1.4054, Bal Acc = 0.3551

/Users/yamira.poldosilva/Documents/UDEM/A25/IFT3395/kaggle2/kaggle2/lib/python3.
13/site-packages/torch/utils/data/dataloader.py:692: UserWarning: 'pin_memory'
argument is set as true but not supported on MPS now, device pinned memory won't
be used.

    warnings.warn(warn_msg)

Epoch 16, Train Loss = 1.4379
Val Loss = 1.3522, Bal Acc = 0.3319

/Users/yamira.poldosilva/Documents/UDEM/A25/IFT3395/kaggle2/kaggle2/lib/python3.
13/site-packages/torch/utils/data/dataloader.py:692: UserWarning: 'pin_memory'
argument is set as true but not supported on MPS now, device pinned memory won't
be used.

    warnings.warn(warn_msg)

Epoch 17, Train Loss = 1.4405
Val Loss = 1.3714, Bal Acc = 0.2899

/Users/yamira.poldosilva/Documents/UDEM/A25/IFT3395/kaggle2/kaggle2/lib/python3.
13/site-packages/torch/utils/data/dataloader.py:692: UserWarning: 'pin_memory'
argument is set as true but not supported on MPS now, device pinned memory won't
be used.

    warnings.warn(warn_msg)

Epoch 18, Train Loss = 1.4496
Val Loss = 1.4150, Bal Acc = 0.3356

/Users/yamira.poldosilva/Documents/UDEM/A25/IFT3395/kaggle2/kaggle2/lib/python3.
13/site-packages/torch/utils/data/dataloader.py:692: UserWarning: 'pin_memory'
argument is set as true but not supported on MPS now, device pinned memory won't
be used.

    warnings.warn(warn_msg)
```

```
Epoch 19, Train Loss = 1.4357
Val Loss = 1.3940, Bal Acc = 0.3375

/Users/yamira.poldosilva/Documents/UDEM/A25/IFT3395/kaggle2/kaggle2/lib/python3.
13/site-packages/torch/utils/data/dataloader.py:692: UserWarning: 'pin_memory'
argument is set as true but not supported on MPS now, device pinned memory won't
be used.

    warnings.warn(warn_msg)
```

```
Epoch 20, Train Loss = 1.4416
Val Loss = 1.3801, Bal Acc = 0.3389
```

```
[57]: bal_acc = balanced_accuracy_score(all_labels, all_preds)
recall = recall_score(all_labels, all_preds, average='macro')
acc = accuracy_score(all_labels, all_preds)

print(f"Validation Balanced Accuracy: {bal_acc:.4f}")
print(f"Validation Recall: {recall:.4f}")
print(f"Validation Accuracy: {acc:.4f}")
print(classification_report(all_labels, all_preds, digits=4))
```

```
Validation Balanced Accuracy: 0.3389
Validation Recall: 0.3389
Validation Accuracy: 0.4635
      precision    recall   f1-score   support
          0       0.6344    0.6941    0.6629      85
          1       0.2439    0.4167    0.3077      24
          2       0.2000    0.0541    0.0851      37
          3       0.3750    0.5294    0.4390      34
          4       0.0000    0.0000    0.0000      12
accuracy           0.4635      192
macro avg       0.2907    0.3389    0.2989      192
weighted avg     0.4163    0.4635    0.4261      192
```

```
/Users/yamira.poldosilva/Documents/UDEM/A25/IFT3395/kaggle2/kaggle2/lib/python3.
13/site-packages/sklearn/metrics/_classification.py:1731:
UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in labels
with no predicted samples. Use `zero_division` parameter to control this
behavior.

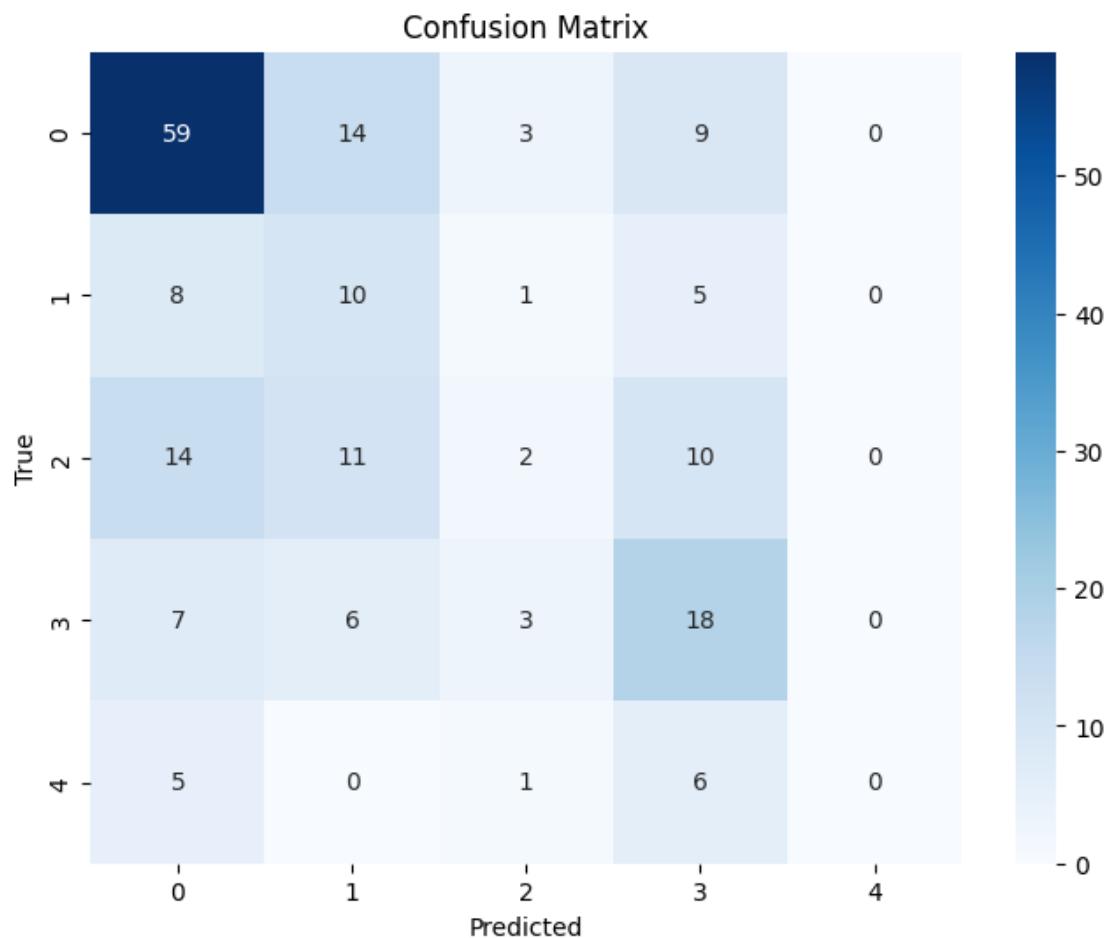
    _warn_prf(average, modifier, f"{metric.capitalize()} is", result.shape[0])
/Users/yamira.poldosilva/Documents/UDEM/A25/IFT3395/kaggle2/kaggle2/lib/python3.
13/site-packages/sklearn/metrics/_classification.py:1731:
UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in labels
with no predicted samples. Use `zero_division` parameter to control this
behavior.

    _warn_prf(average, modifier, f"{metric.capitalize()} is", result.shape[0])
/Users/yamira.poldosilva/Documents/UDEM/A25/IFT3395/kaggle2/kaggle2/lib/python3.
```

```
13/site-packages/sklearn/metrics/_classification.py:1731:  
UndefinedMetricWarning: Precision is ill-defined and being set to 0.0 in labels  
with no predicted samples. Use `zero_division` parameter to control this  
behavior.  
    _warn_prf(average, modifier, f"{metric.capitalize()} is", result.shape[0])
```

```
[58]: import seaborn as sns
```

```
cm = confusion_matrix(all_labels, all_preds)  
plt.figure(figsize=(8, 6))  
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues')  
plt.xlabel('Predicted')  
plt.ylabel('True')  
plt.title('Confusion Matrix')  
plt.show()
```



```
[59]: torch.save({
    "model_state_dict": model.state_dict(),
    "num_classes": 5,
}, "cnnnet.pth")

[60]: checkpoint = torch.load("cnnnet.pth", map_location=device, weights_only=False)

model = TinyCNN()

model.load_state_dict(checkpoint["model_state_dict"])
model.to(device)
model.eval()

[60]: TinyCNN(
    (conv1): Conv2d(3, 32, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (bn1): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    (conv2): Conv2d(32, 32, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (bn2): BatchNorm2d(32, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    (conv3): Conv2d(32, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (bn3): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    (conv4): Conv2d(64, 64, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (bn4): BatchNorm2d(64, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    (conv5): Conv2d(64, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (bn5): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    (conv6): Conv2d(128, 128, kernel_size=(3, 3), stride=(1, 1), padding=(1, 1))
    (bn6): BatchNorm2d(128, eps=1e-05, momentum=0.1, affine=True,
track_running_stats=True)
    (global_pool): AdaptiveAvgPool2d(output_size=(1, 1))
    (dropout): Dropout(p=0.2, inplace=False)
    (fc): Linear(in_features=128, out_features=5, bias=True)
)

[61]: test_dataset = pickle.load(open('ift-3395-6390-kaggle-2-competition-fall-2025/
˓→test_data.pkl', 'rb'))
test_images = test_dataset['images']

test_transform = transforms.Compose([
    transforms.Resize((64, 64)),
    transforms.ToTensor(),
```

```
    transforms.Normalize(mean=IR_MEAN, std=IR_STD)
])
```

```
[62]: class TestPKLDataset(Dataset):
    def __init__(self, images, transform=None):
        self.images = images
        self.transform = transform

    def __len__(self):
        return len(self.images)

    def __getitem__(self, idx):
        image = self.images[idx]

        image = Image.fromarray(image.astype('uint8'))

        if self.transform:
            image = self.transform(image)
        return image
```

```
[63]: test_ds = TestPKLDataset(test_images, transform=test_transform)
test_loader = DataLoader(test_ds, batch_size=64, shuffle=False, pin_memory=True)
preds = []

with torch.no_grad():
    for images in test_loader:
        images = images.to(device)
        outputs = model(images)
        _, predicted = torch.max(outputs, 1)
        preds.extend(predicted.cpu().numpy())

df = pd.DataFrame({
    "ID": np.arange(1, len(preds) + 1),
    "Label": preds
})

df.to_csv("IFT3395_YAPS_MCSV53.csv", index=False)
```

```
/Users/yamira.poldosilva/Documents/UDEM/A25/IFT3395/kaggle2/kaggle2/lib/python3.13/site-packages/torch/utils/data/dataloader.py:692: UserWarning: 'pin_memory' argument is set as true but not supported on MPS now, device pinned memory won't be used.
  warnings.warn(warn_msg)
```