

# Validation

```
In [17]: import torch
import torch.nn as nn
import torch.optim as optim
from torch.optim import lr_scheduler
import numpy as np
import torchvision
from torchvision import datasets, models, transforms
from torch.utils.data import TensorDataset, DataLoader, Dataset
from sklearn.utils import shuffle
```

```
In [2]: device = torch.device("cuda:0" if torch.cuda.is_available() else "cpu")
```

```
In [3]: import numpy as np
import random
import numpy.random as npr
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
import matplotlib.image as img
from matplotlib import offsetbox
from PIL import Image
import shutil
import pandas as pd
import zipfile as zf
from tqdm.auto import tqdm
import yaml
import collections
import json
import random
import matplotlib.patches as patches
import pickle
import cv2
import time
import os
import copy
import math
from time import time
import warnings
%matplotlib inline
plt.style.use('bmh')
warnings.filterwarnings('ignore')
```

```
In [4]: num_of_gpus = torch.cuda.device_count()
print(num_of_gpus)
```

1

```
In [6]: class ResNet101(nn.Module):
def __init__(self, num_classes):
super(ResNet101, self).__init__()
self.resnet = nn.Sequential(*list(torchvision.models.resnet101(pretrained=True).children()[1:]))
self.classifier = nn.Linear(2048, num_classes)
```

```
def forward(self, x):
    x = self.resnet(x)
    x = x.view(x.size(0), -1)
    x = self.classifier(x)
    return x
```

```
In [7]: PATH = './FPLD-ResNet-101-CLS.pth'
        model = torch.load(PATH)
        model = model.to(device)
```

```
In [8]: def load_images_labels(df):
        X, t = [], []
        for i in range(len(df)):
            if df['Image'][i]:
                img = cv2.imread(df['Image'][i])
                X.append(np.array(img).reshape(3,300,300))
                t.append(int(df['Label'][i]))
            else:
                break
        print("Processed all the Images and their corresponding attributes")
        return np.array(X), np.array(t)
```

```
In [9]: val_df = pd.read_csv('Val.csv')
        val_df = val_df.dropna(how='any')
        val_df
```

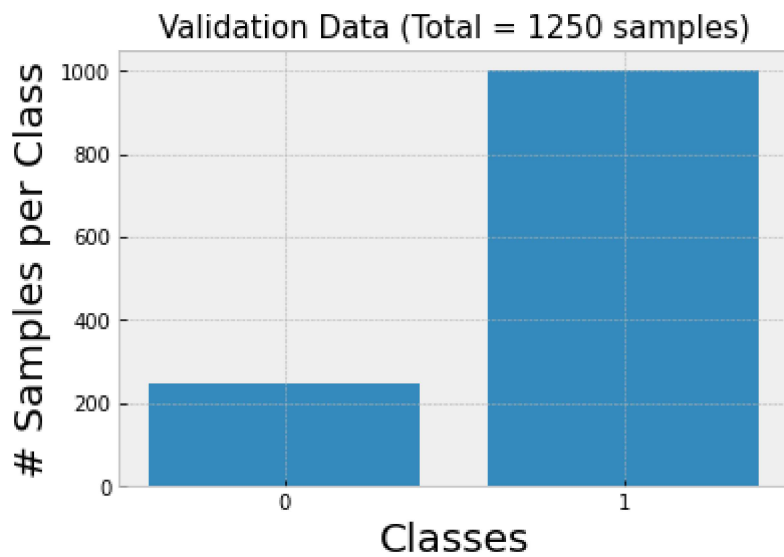
```
Out[9]:
```

	Image	Label
0	/blue/eel6825/ravipatim/2015/preprocessed_data...	1
1	/blue/eel6825/ravipatim/2015/preprocessed_data...	1
2	/blue/eel6825/ravipatim/2015/preprocessed_data...	1
3	/blue/eel6825/ravipatim/2015/preprocessed_data...	1
4	/blue/eel6825/ravipatim/2015/preprocessed_data...	1
...	...	...
1245	/blue/eel6825/ravipatim/2015/preprocessed_data...	0
1246	/blue/eel6825/ravipatim/2015/preprocessed_data...	0
1247	/blue/eel6825/ravipatim/2015/preprocessed_data...	0
1248	/blue/eel6825/ravipatim/2015/preprocessed_data...	0
1249	/blue/eel6825/ravipatim/2015/preprocessed_data...	0

1250 rows × 2 columns

```
In [12]: vals, counts = np.unique(np.array(val_df['Label']), return_counts=True)

        plt.bar(vals, counts)
        plt.xticks(range(2), range(2))
        plt.xlabel('Classes', size=20)
        plt.ylabel('# Samples per Class', size=20)
        plt.title('Validation Data (Total = '+str(len(np.array(val_df['Image'])))+' samples)',
```



```
In [11]: X,y = load_images_labels(val_df)
```

Processed all the Images and their corresponding attributes

```
In [13]: transform_train = transforms.Compose([
    transforms.ToPILImage(),
    transforms.ToTensor(),
    transforms.Normalize((0.485, 0.456, 0.406), (0.229, 0.224, 0.225))
])

transform_test = transforms.Compose([
    transforms.ToPILImage(),
    transforms.ToTensor(),
    transforms.Normalize((0.485, 0.456, 0.406), (0.229, 0.224, 0.225))
])
```

```
In [18]: val_data = TensorDataset(torch.Tensor(X), torch.LongTensor(y))
```

```
In [19]: val_dataloader = DataLoader(val_data, batch_size=32, shuffle=False)
```

```
In [20]: avg=[]
model.eval()    # Set model to evaluate mode
iter=0
for inputs, labels in val_dataloader:
    inputs = inputs.to(device)
    labels = labels.to(device)
    outputs = model(inputs)
    _, predicted = torch.max(outputs.data, 1)

    labels =labels.to(device)
    pred = predicted.to(device)
    labels =labels.cpu().numpy()
    pred =pred.cpu().numpy()
    acc = 100.0 * accuracy_score(labels,pred)
    avg.append(acc)
    print('Validation Accuracy of the network for iteration {0}: {1} %'.format(iter,
    iter+=1
print("Average Validation Accuracy of the network",sum(avg)//iter)
```

Validation Accuracy of the network for iteration 0: 90.625 %  
Validation Accuracy of the network for iteration 1: 96.875 %  
Validation Accuracy of the network for iteration 2: 93.75 %  
Validation Accuracy of the network for iteration 3: 93.75 %  
Validation Accuracy of the network for iteration 4: 81.25 %  
Validation Accuracy of the network for iteration 5: 93.75 %  
Validation Accuracy of the network for iteration 6: 93.75 %  
Validation Accuracy of the network for iteration 7: 100.0 %  
Validation Accuracy of the network for iteration 8: 90.625 %  
Validation Accuracy of the network for iteration 9: 96.875 %  
Validation Accuracy of the network for iteration 10: 93.75 %  
Validation Accuracy of the network for iteration 11: 90.625 %  
Validation Accuracy of the network for iteration 12: 93.75 %  
Validation Accuracy of the network for iteration 13: 96.875 %  
Validation Accuracy of the network for iteration 14: 93.75 %  
Validation Accuracy of the network for iteration 15: 96.875 %  
Validation Accuracy of the network for iteration 16: 90.625 %  
Validation Accuracy of the network for iteration 17: 93.75 %  
Validation Accuracy of the network for iteration 18: 93.75 %  
Validation Accuracy of the network for iteration 19: 93.75 %  
Validation Accuracy of the network for iteration 20: 100.0 %  
Validation Accuracy of the network for iteration 21: 93.75 %  
Validation Accuracy of the network for iteration 22: 90.625 %  
Validation Accuracy of the network for iteration 23: 87.5 %  
Validation Accuracy of the network for iteration 24: 96.875 %  
Validation Accuracy of the network for iteration 25: 96.875 %  
Validation Accuracy of the network for iteration 26: 90.625 %  
Validation Accuracy of the network for iteration 27: 93.75 %  
Validation Accuracy of the network for iteration 28: 96.875 %  
Validation Accuracy of the network for iteration 29: 90.625 %  
Validation Accuracy of the network for iteration 30: 96.875 %  
Validation Accuracy of the network for iteration 31: 84.375 %  
Validation Accuracy of the network for iteration 32: 90.625 %  
Validation Accuracy of the network for iteration 33: 96.875 %  
Validation Accuracy of the network for iteration 34: 90.625 %  
Validation Accuracy of the network for iteration 35: 87.5 %  
Validation Accuracy of the network for iteration 36: 84.375 %  
Validation Accuracy of the network for iteration 37: 75.0 %  
Validation Accuracy of the network for iteration 38: 96.875 %  
Validation Accuracy of the network for iteration 39: 50.0 %  
Average Validation Accuracy of the network 91.0