FLAG CLASSIFIER

SAAHIL RACHH, SHREY PAREKH, DEVANG SHAH, PRIYAMBI HIRAN

ABOUT OUR IDEA

Image classification system that involves identification of a SAARC country, given its flag.

USES

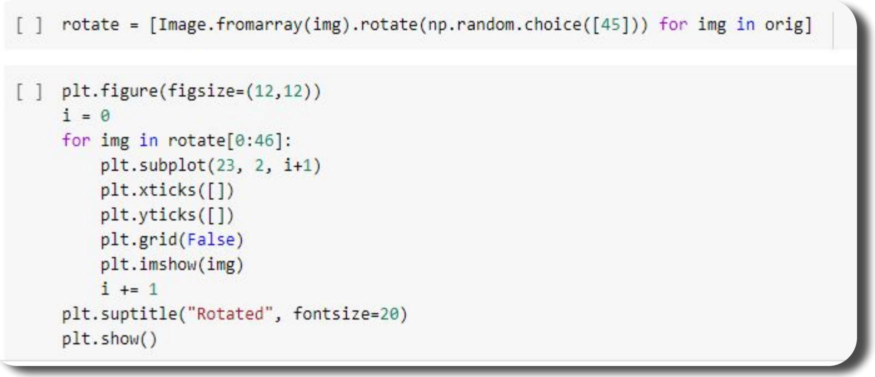
1. Overcrowded places
2. Sport Events
3. International Conferences
4. Miscellaneous applications

Libraries used:

- ALBUMENTATIONS- Image augmentation,
- TORCHVISION(Resnet pretrained model)- Image detection and classification,
- OPENCV- Live images and videos,
- OTHER LIBRARIES- Pytorch, PIL, numpy, etc.

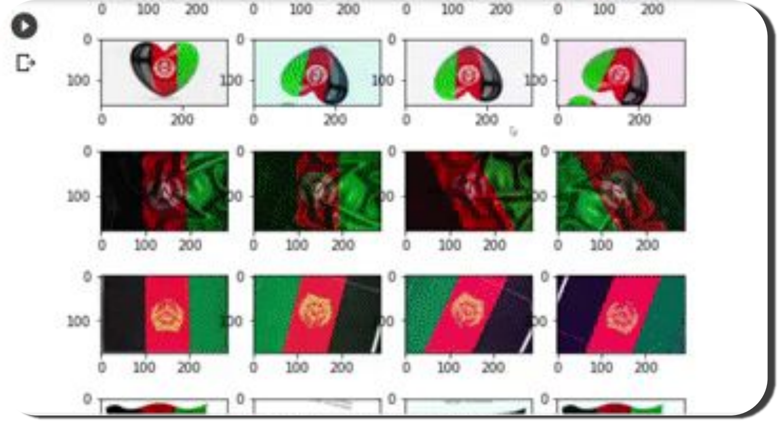
PLATFORMS USED: Google Colab, Google drive, VS code

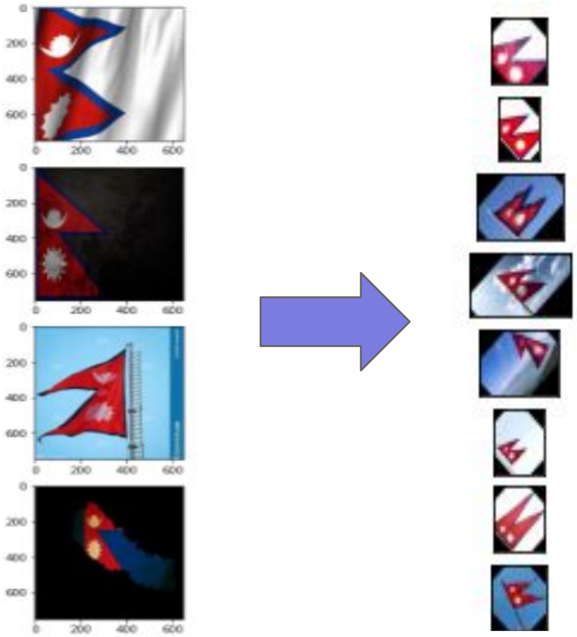
Augmentations

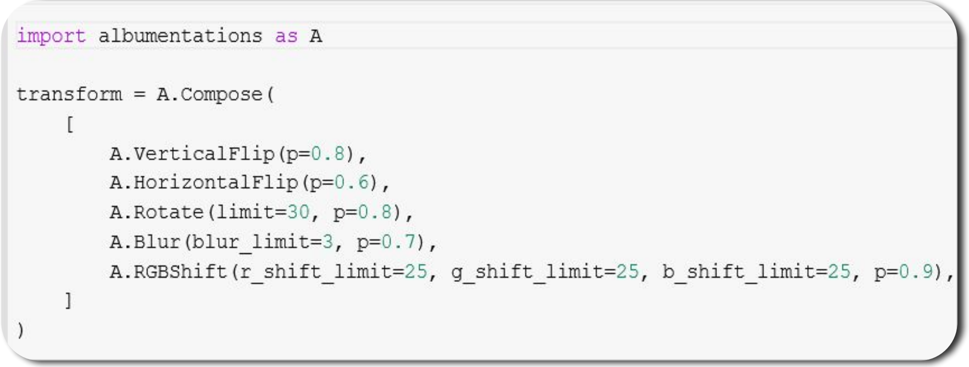


```
[ ] rotate = [Image.fromarray(img).rotate(np.random.choice([45])) for img in orig]
```

```
[ ] plt.figure(figsize=(12,12))
    i = 0
    for img in rotate[0:46]:
        plt.subplot(23, 2, i+1)
        plt.xticks([])
        plt.yticks([])
        plt.grid(False)
        plt.imshow(img)
        i += 1
    plt.suptitle("Rotated", fontsize=20)
    plt.show()
```

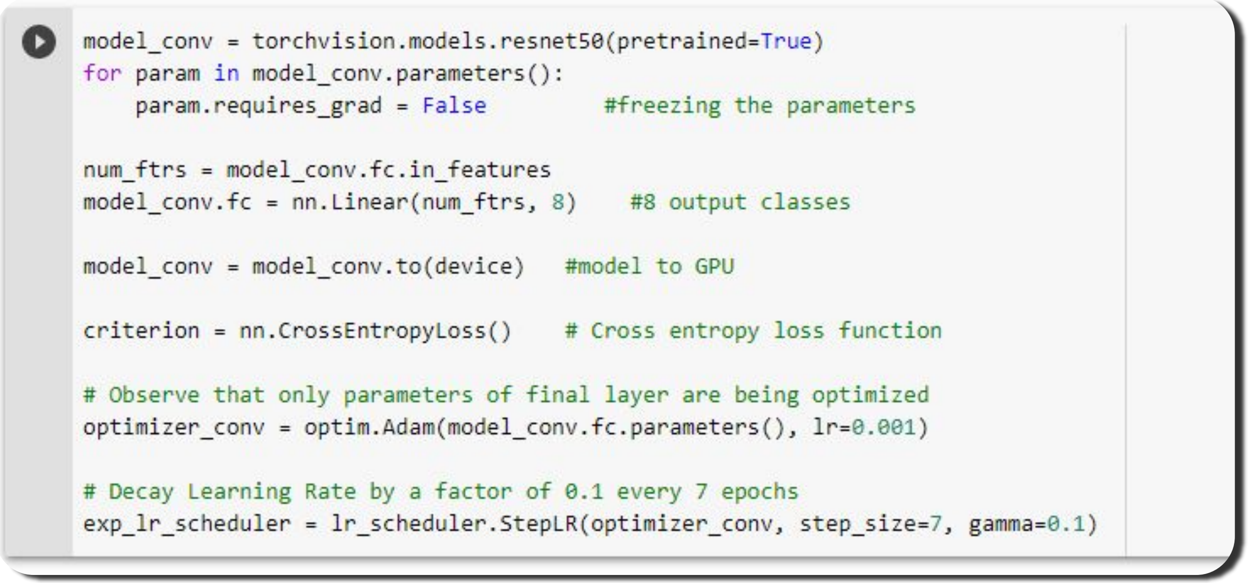






```
import albumentations as A

transform = A.Compose(
    [
        A.VerticalFlip(p=0.8),
        A.HorizontalFlip(p=0.6),
        A.Rotate(limit=30, p=0.8),
        A.Blur(blur_limit=3, p=0.7),
        A.RGBShift(r_shift_limit=25, g_shift_limit=25, b_shift_limit=25, p=0.9),
    ]
)
```

ResNet Pre-trained Model



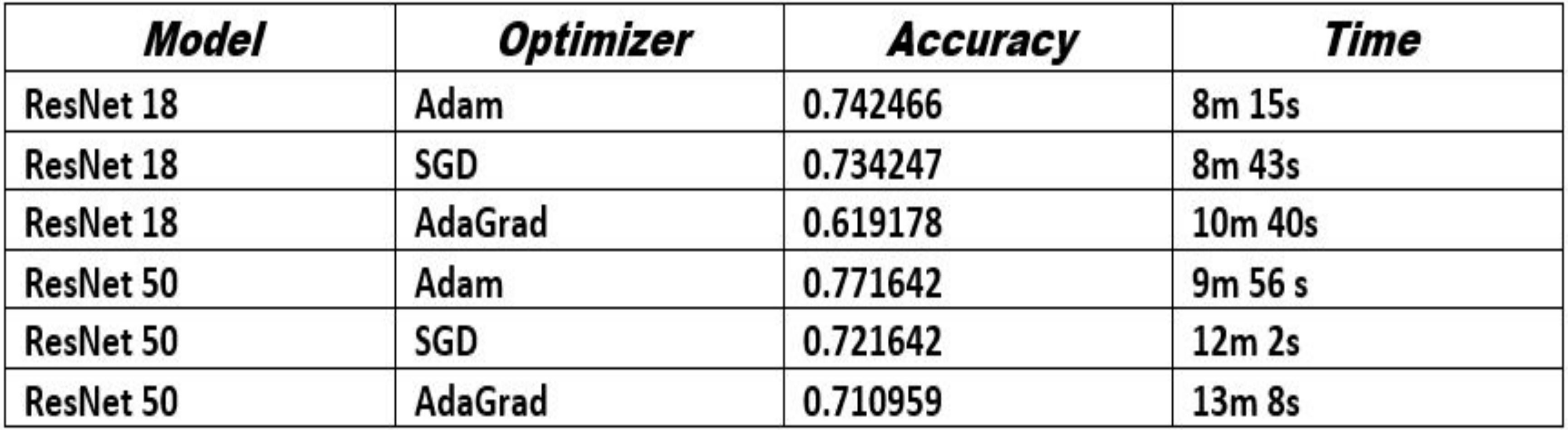
```
model_conv = torchvision.models.resnet50(pretrained=True)
for param in model_conv.parameters():
    param.requires_grad = False         #freezing the parameters

num_ftrs = model_conv.fc.in_features
model_conv.fc = nn.Linear(num_ftrs, 8)    #8 output classes

model_conv = model_conv.to(device)   #model to GPU

criterion = nn.CrossEntropyLoss()    # Cross entropy loss function

# Observe that only parameters of final layer are being optimized
optimizer_conv = optim.Adam(model_conv.fc.parameters(), lr=0.001)

# Decay Learning Rate by a factor of 0.1 every 7 epochs
exp_lr_scheduler = lr_scheduler.StepLR(optimizer_conv, step_size=7, gamma=0.1)
```

Here we used torchvision library to load the pretrained resnet model. We tried playing with the learning rate and momentum to see how the model learns these datasets.

We also used torch.optim library to implement various optimisation algorithms, one of them being SGD

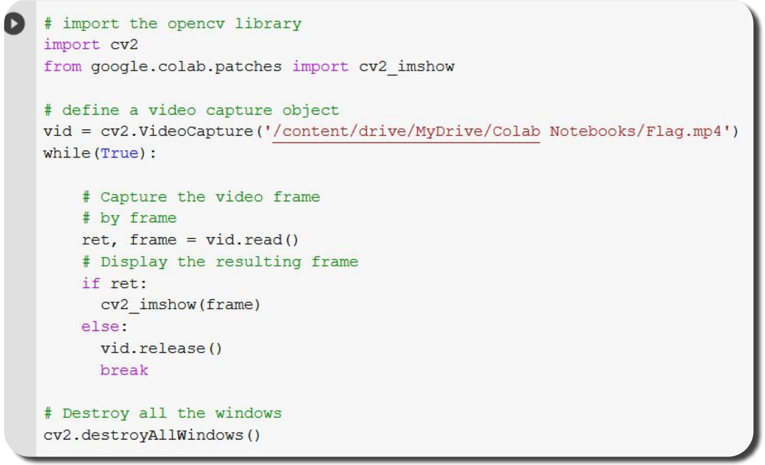
['India', 'Afghanistan', 'Sri Lanka', 'Sri Lanka', 'Pakistan', 'Maldives', 'Bangladesh', 'Maldives']
0
200
0
250
500
750
1000
1250
1500
1750

Choosing the right model

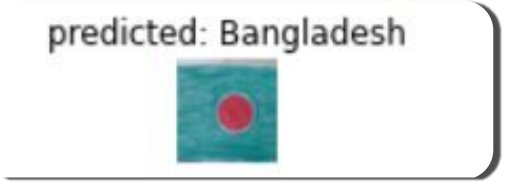


Model	Optimizer	Accuracy	Time
ResNet 18	Adam	0.742466	8m 15s
ResNet 18	SGD	0.734247	8m 43s
ResNet 18	AdaGrad	0.619178	10m 40s
ResNet 50	Adam	0.771642	9m 56 s
ResNet 50	SGD	0.721642	12m 2s
ResNet 50	AdaGrad	0.710959	13m 8s

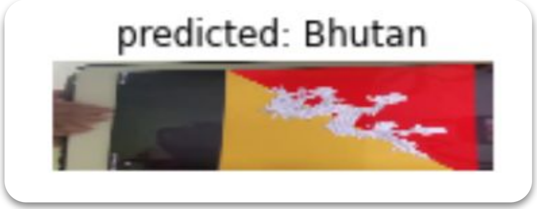
OPENCV



```
# import the opencv library
import cv2
from google.colab.patches import cv2_imshow

# define a video capture object
vid = cv2.VideoCapture('/content/drive/MyDrive/Colab Notebooks/Flag.mp4')
while(True):

    # Capture the video frame
    # by frame
    ret, frame = vid.read()
    # Display the resulting frame
    if ret:
      cv2_imshow(frame)
    else:
      vid.release()
      break

# Destroy all the windows
cv2.destroyAllWindows()
```

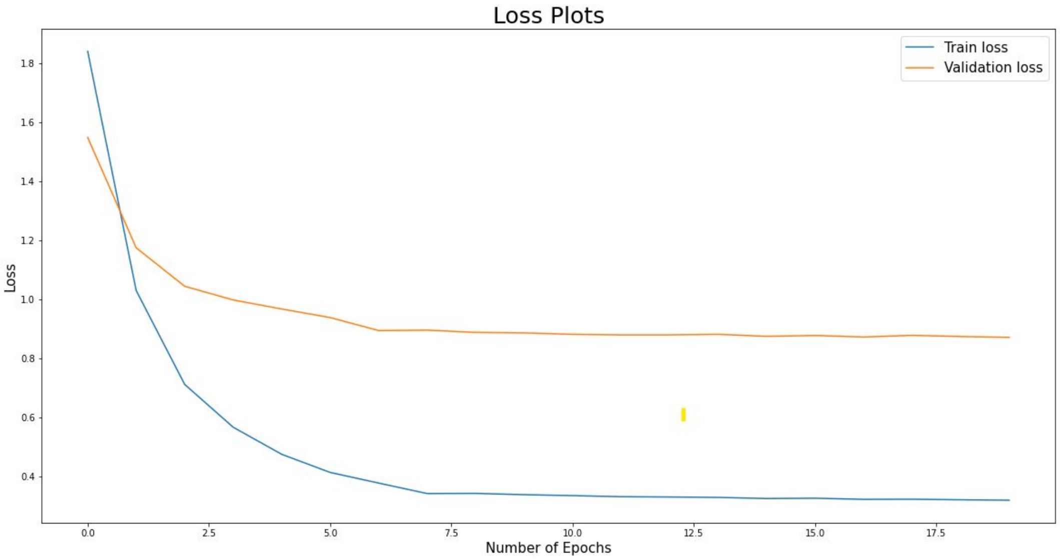


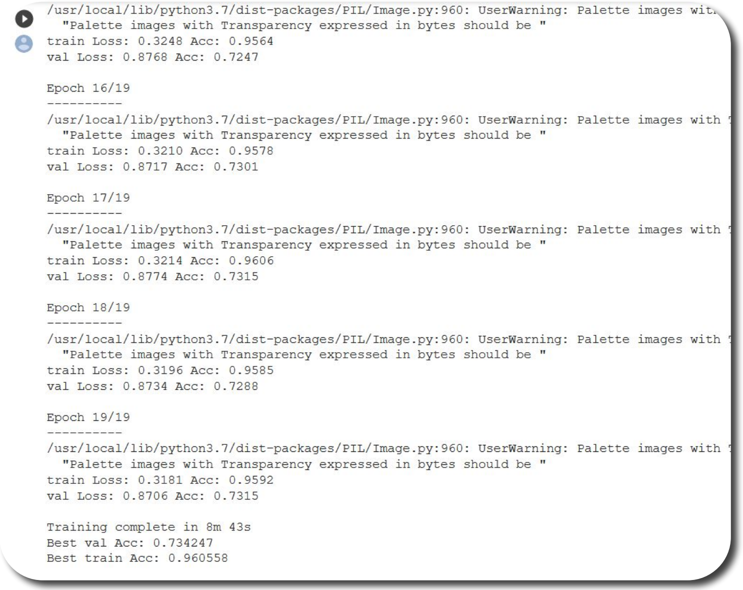
predicted: Bangladesh



predicted: Bhutan

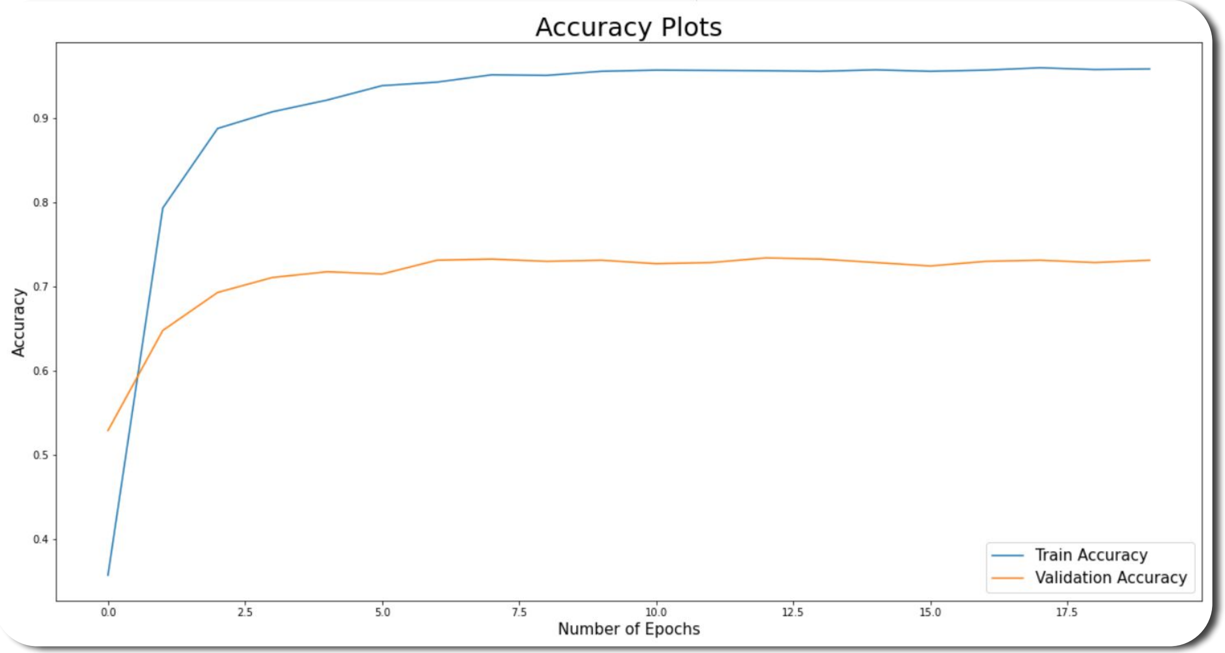
Runthrough - Epochs and loss plot

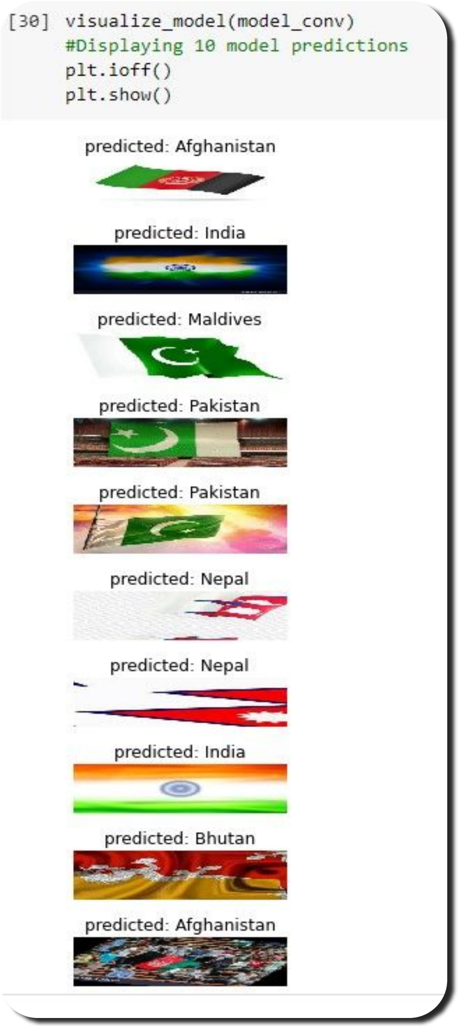




```
/usr/local/lib/python3.7/dist-packages/PIL/Image.py:960: UserWarning: Palette images wit
  "Palette images with Transparency expressed in bytes should be "
train Loss: 0.3248 Acc: 0.9564
val Loss: 0.8768 Acc: 0.7247

Epoch 16/19
----------
/usr/local/lib/python3.7/dist-packages/PIL/Image.py:960: UserWarning: Palette images with
  "Palette images with Transparency expressed in bytes should be "
train Loss: 0.3210 Acc: 0.9578
val Loss: 0.8717 Acc: 0.7301

Epoch 17/19
----------
/usr/local/lib/python3.7/dist-packages/PIL/Image.py:960: UserWarning: Palette images with
  "Palette images with Transparency expressed in bytes should be "
train Loss: 0.3214 Acc: 0.9606
val Loss: 0.8774 Acc: 0.7315

Epoch 18/19
----------
/usr/local/lib/python3.7/dist-packages/PIL/Image.py:960: UserWarning: Palette images with
  "Palette images with Transparency expressed in bytes should be "
train Loss: 0.3196 Acc: 0.9585
val Loss: 0.8734 Acc: 0.7288

Epoch 19/19
----------
/usr/local/lib/python3.7/dist-packages/PIL/Image.py:960: UserWarning: Palette images with
  "Palette images with Transparency expressed in bytes should be "
train Loss: 0.3181 Acc: 0.9592
val Loss: 0.8706 Acc: 0.7315

Training complete in 8m 43s
Best val Acc: 0.734247
Best train Acc: 0.960558
```

Sample Predictions and accuracy Plot





```
[30] visualize_model(model_conv)
     #Displaying 10 model predictions
     plt.ioff()
     plt.show()
```

```
predicted: Afghanistan
predicted: India
predicted: Maldives
predicted: Pakistan
predicted: Pakistan
predicted: Nepal
predicted: Nepal
predicted: India
predicted: Bhutan
predicted: Afghanistan
```

Training complete in 8m 43s
Best Validation Accuracy: 0.734247

ML SCOUTS

FUTURE SCOPE

An android application can be created which uses the mobile camera to detect and identify flags at conferences or events.

Also an object detection Model can be prepared on the same.

When a country’s flag is identified, a fun fact and current news of that country can be displayed alongside the name.

Identifying symbols or texts embedded on top of country flags can also be a future aspect of this project.

THANK YOU