

## بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

```
# %reset
```

## Imports

```
!pip install ipython-autotime
```

```
Requirement already satisfied: ipython-autotime in /usr/local/lib/python3.6/dist-packages
Requirement already satisfied: ipython in /usr/local/lib/python3.6/dist-packages (from ipython-autotime)
Requirement already satisfied: pexpect; sys_platform != "win32" in /usr/local/lib/python3.6/dist-packages (from ipython)
Requirement already satisfied: simplegeneric>0.8 in /usr/local/lib/python3.6/dist-packages (from ipython)
Requirement already satisfied: prompt-toolkit<2.0.0,>=1.0.4 in /usr/local/lib/python3.6/dist-packages (from ipython)
Requirement already satisfied: pickleshare in /usr/local/lib/python3.6/dist-packages (from ipython)
Requirement already satisfied: traitlets>=4.2 in /usr/local/lib/python3.6/dist-packages (from ipython)
Requirement already satisfied: pygments in /usr/local/lib/python3.6/dist-packages (from ipython)
Requirement already satisfied: decorator in /usr/local/lib/python3.6/dist-packages (from ipython)
Requirement already satisfied: setuptools>=18.5 in /usr/local/lib/python3.6/dist-packages (from ipython)
Requirement already satisfied: ptyprocess>=0.5 in /usr/local/lib/python3.6/dist-packages (from ipython)
Requirement already satisfied: wcwidth in /usr/local/lib/python3.6/dist-packages (from ipython)
Requirement already satisfied: six>=1.9.0 in /usr/local/lib/python3.6/dist-packages (from ipython)
Requirement already satisfied: ipython-genutils in /usr/local/lib/python3.6/dist-packages (from ipython)
time: 2.28 s (started: 2021-01-05 22:29:01 +00:00)
```

```
# necessary imports
import os
import cv2
import numpy as np
from imutils import paths
from sklearn.preprocessing import LabelBinarizer
from tqdm import tqdm
```

```
import matplotlib.pyplot as plt
%matplotlib inline
```

```
from google.colab.patches import cv2_imshow
```

```
%load_ext autotime
```

```
The autotime extension is already loaded. To reload it, use:
%reload_ext autotime
time: 4.5 ms (started: 2021-01-05 22:29:03 +00:00)
```

## ▼ Initializing

```
img_width = 90
img_height = 90
```

```
time: 1.15 ms (started: 2021-01-05 22:29:03 +00:00)
```

## ▼ NASNetLarge Model

```
from keras.applications import InceptionResNetV2
from keras.models import Model
from keras.layers import Dense
from keras.layers import Flatten
```

```
time: 1.01 ms (started: 2021-01-05 22:29:04 +00:00)
```

```
# from keras.applications import NASNetLarge
# # load NASNetLarge model without classification layers
# model = NASNetLarge(include_top=False, weights='imagenet', input_shape=(331, 331, 3))
```

```
time: 644 µs (started: 2021-01-05 22:29:05 +00:00)
```

```
# # load NASNetMobile model without classification layers
# model = keras.applications.nasnet.NASNetMobile(include_top=False, weights='imagenet', input_shape=(331, 331, 3))
```

```
time: 597 µs (started: 2021-01-05 22:29:06 +00:00)
```

```
model = InceptionResNetV2(include_top=False, weights='imagenet', input_shape=(img_width, img_height, 3))
```

```
time: 9.85 s (started: 2021-01-05 22:29:06 +00:00)
```

```
# add new classification layers
flat1 = Flatten()(model.layers[-1].output) # flatten last layer
class1 = Dense(1024, activation='relu')(flat1) # add FC layer on previous layer
output = Dense(6, activation='softmax')(class1) # add softmax layer
```

```
time: 20.9 ms (started: 2021-01-05 22:29:16 +00:00)
```

```
# define the new model
model = Model(inputs=model.inputs, outputs=output)
model.summary()
```

activation\_top101101

block8_9_conv (Conv2D)	(None, 1, 1, 2080)	933920	block8_9_mixed[0][0]
block8_9 (Lambda)	(None, 1, 1, 2080)	0	block8_8_ac[0][0] block8_9_conv[0][0]
block8_9_ac (Activation)	(None, 1, 1, 2080)	0	block8_9[0][0]
conv2d_403 (Conv2D)	(None, 1, 1, 192)	399360	block8_9_ac[0][0]
batch_normalization_403 (BatchN	(None, 1, 1, 192)	576	conv2d_403[0][0]
activation_403 (Activation)	(None, 1, 1, 192)	0	batch_normalization_
conv2d_404 (Conv2D)	(None, 1, 1, 224)	129024	activation_403[0][0]
batch_normalization_404 (BatchN	(None, 1, 1, 224)	672	conv2d_404[0][0]
activation_404 (Activation)	(None, 1, 1, 224)	0	batch_normalization_
conv2d_402 (Conv2D)	(None, 1, 1, 192)	399360	block8_9_ac[0][0]
conv2d_405 (Conv2D)	(None, 1, 1, 256)	172032	activation_404[0][0]
batch_normalization_402 (BatchN	(None, 1, 1, 192)	576	conv2d_402[0][0]
batch_normalization_405 (BatchN	(None, 1, 1, 256)	768	conv2d_405[0][0]
activation_402 (Activation)	(None, 1, 1, 192)	0	batch_normalization_
activation_405 (Activation)	(None, 1, 1, 256)	0	batch_normalization_
block8_10_mixed (Concatenate)	(None, 1, 1, 448)	0	activation_402[0][0] activation_405[0][0]
block8_10_conv (Conv2D)	(None, 1, 1, 2080)	933920	block8_10_mixed[0][0]
block8_10 (Lambda)	(None, 1, 1, 2080)	0	block8_9_ac[0][0] block8_10_conv[0][0]
conv_7b (Conv2D)	(None, 1, 1, 1536)	3194880	block8_10[0][0]
conv_7b_bn (BatchNormalization)	(None, 1, 1, 1536)	4608	conv_7b[0][0]
conv_7b_ac (Activation)	(None, 1, 1, 1536)	0	conv_7b_bn[0][0]
flatten_1 (Flatten)	(None, 1536)	0	conv_7b_ac[0][0]
dense_2 (Dense)	(None, 1024)	1573888	flatten_1[0][0]
dense_3 (Dense)	(None, 6)	6150	dense_2[0][0]
=====			
Total params: 55,916,774			
Trainable params: 55,856,230			
Non-trainable params: 60,544			
time: 320 ms (started: 2021-01-05 22:29:16 +00:00)			

## ▼ Loading Data

```
# A function to load data from a given directory
def load_data(data_dir):
    data = []
    labels = []
    class_dirs = os.listdir(data_dir)

    for direc in class_dirs:
        class_dir = os.path.join(data_dir, direc)
        for imagepath in tqdm(list(paths.list_images(class_dir))):
            image = cv2.imread(imagepath)
            image = cv2.resize(image, (img_width, img_height)) # incase images not of same size
            data.append(image)
            labels.append(direc)
    # normalizing and converting to numpy array format
    data = np.array(data, dtype='float')/255.0
    labels = np.array(labels)
    return data, labels
```

time: 7.43 ms (started: 2021-01-05 22:29:17 +00:00)

```
train_dir = "/content/drive/MyDrive/CV/Assignment 3/seg_train/seg_train/"
test_dir = "/content/drive/MyDrive/CV/Assignment 3/seg_test/seg_test/"
pred_dir = "/content/drive/MyDrive/CV/Assignment 3/pred/seg_pred/seg_pred/"
```

time: 804 µs (started: 2021-01-05 22:29:17 +00:00)

```
from google.colab import drive
drive.mount('/content/drive')
```

Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.moun  
time: 2.16 ms (started: 2021-01-05 22:29:17 +00:00)

```
print('loading train images')
X_train, y_train = load_data(train_dir)
```

```
loading train images
100%|██████████| 2191/2191 [00:06<00:00, 350.67it/s]
100%|██████████| 2271/2271 [00:06<00:00, 338.24it/s]
100%|██████████| 2404/2404 [00:06<00:00, 356.20it/s]
100%|██████████| 2512/2512 [00:07<00:00, 357.04it/s]
100%|██████████| 2274/2274 [00:06<00:00, 362.63it/s]
100%|██████████| 2382/2382 [00:06<00:00, 349.43it/s]
time: 44.9 s (started: 2021-01-05 22:29:17 +00:00)
```

```
X_valid, y_valid = load_data(test_dir)
```

```
100%|██████████| 437/437 [00:01<00:00, 316.11it/s]
100%|██████████| 474/474 [00:01<00:00, 311.99it/s]
100%|██████████| 553/553 [00:01<00:00, 336.27it/s]
100%|██████████| 525/525 [00:01<00:00, 345.10it/s]
100%|██████████| 510/510 [00:01<00:00, 340.71it/s]
100%|██████████| 501/501 [00:01<00:00, 341.91it/s]
time: 9.39 s (started: 2021-01-05 22:30:02 +00:00)
```

```
X_train = np.append(X_train, X_valid, axis=0)
y_train = np.append(y_train, y_valid, axis=0)
```

```
time: 3.54 s (started: 2021-01-05 22:30:11 +00:00)
```

```
lb = LabelBinarizer()
y_train = lb.fit_transform(y_train)
```

```
time: 32.9 ms (started: 2021-01-05 22:30:14 +00:00)
```

```
from sklearn.model_selection import train_test_split
(X_train, X_valid, y_train, y_valid) = train_test_split(X_train, y_train, test_size=0.2, rand
```

```
time: 1.04 s (started: 2021-01-05 22:30:15 +00:00)
```

## Compile the model

```
from keras.optimizers import SGD
sgd = SGD(lr=0.001, decay=1e-7, momentum=.9)
model.compile(loss='categorical_crossentropy', optimizer=sgd, metrics=['accuracy'])
```

```
time: 67.8 ms (started: 2021-01-05 22:30:16 +00:00)
```

## Train the model

```
H = model.fit(X_train, y_train, batch_size=128,
              epochs=10,
              validation_data=(X_valid, y_valid))
```

```
Epoch 1/10
107/107 [=====] - 42s 270ms/step - loss: 1.5247 - accuracy: 0.3
Epoch 2/10
107/107 [=====] - 26s 243ms/step - loss: 0.5779 - accuracy: 0.8
Epoch 3/10
107/107 [=====] - 26s 246ms/step - loss: 0.4025 - accuracy: 0.8
Epoch 4/10
107/107 [=====] - 27s 249ms/step - loss: 0.3058 - accuracy: 0.8
```

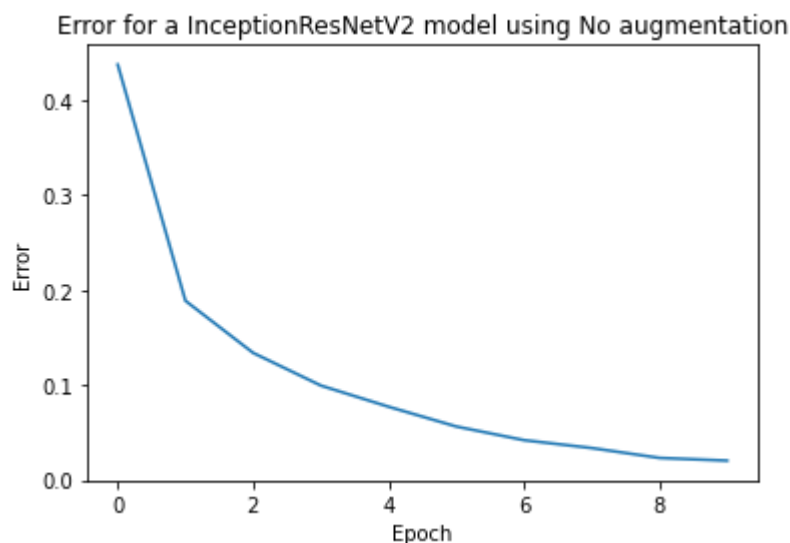
```
Epoch 5/10
107/107 [=====] - 27s 252ms/step - loss: 0.2404 - accuracy: 0.9
Epoch 6/10
107/107 [=====] - 27s 254ms/step - loss: 0.1763 - accuracy: 0.9
Epoch 7/10
107/107 [=====] - 27s 256ms/step - loss: 0.1338 - accuracy: 0.9
Epoch 8/10
107/107 [=====] - 27s 257ms/step - loss: 0.1109 - accuracy: 0.9
Epoch 9/10
107/107 [=====] - 28s 258ms/step - loss: 0.0781 - accuracy: 0.9
Epoch 10/10
107/107 [=====] - 28s 258ms/step - loss: 0.0627 - accuracy: 0.9
time: 4min 46s (started: 2021-01-05 22:30:16 +00:00)
```



```
# save the model's trained weights
model.save_weights('/content/drive/MyDrive/CV/Assignment 3/InceptionResNetV2_transfer_trained_1000000.h5')
time: 1.88 s (started: 2021-01-05 22:35:02 +00:00)
```

```
# model.load_weights('/content/drive/MyDrive/CV/Assignment 3/InceptionResNetV2_transfer_train_1000000.h5')
time: 1.04 ms (started: 2021-01-05 22:35:06 +00:00)
```

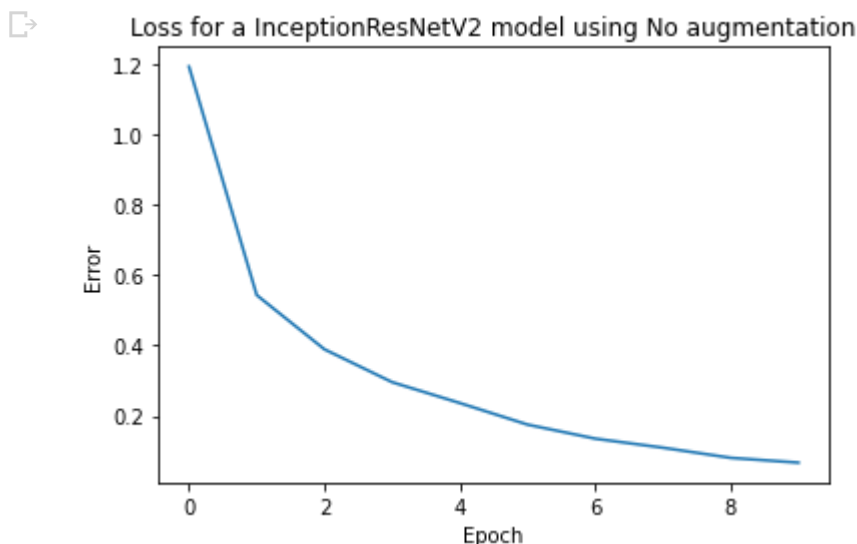
```
simple_acc = H.history['accuracy']
plt.plot([1 - acc for acc in simple_acc])
plt.title('Error for a InceptionResNetV2 model using No augmentation')
plt.ylabel('Error')
plt.xlabel('Epoch')
plt.savefig('/content/drive/MyDrive/CV/Assignment 3/InceptionResNetV2/simple_acc_error.png')
plt.show()
```



```
time: 195 ms (started: 2021-01-05 22:35:06 +00:00)
```

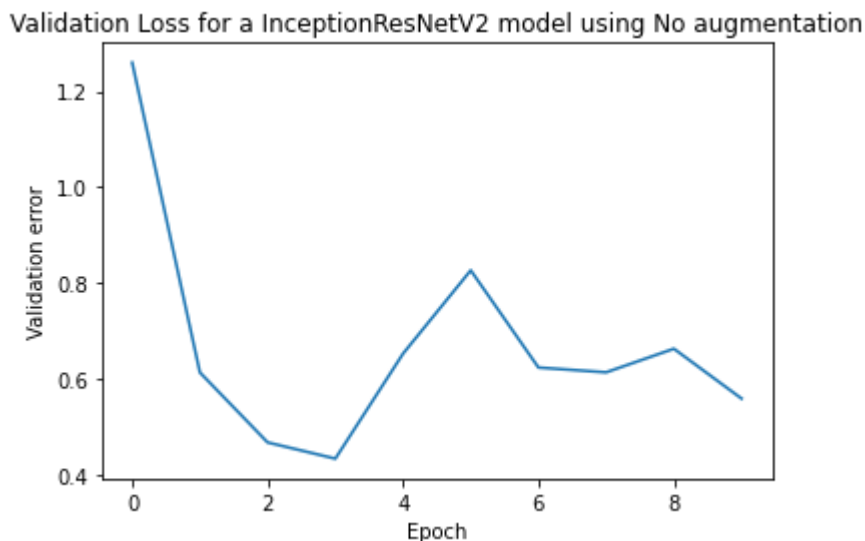
```
simple_loss = H.history['loss']
plt.plot([los for los in simple_loss])
```

```
plt.title('Loss for a InceptionResNetV2 model using No augmentation')
plt.ylabel('Error')
plt.xlabel('Epoch')
plt.savefig('/content/drive/MyDrive/CV/Assignment 3/InceptionResNetV2/simple_loss_error.png')
plt.show()
```



time: 176 ms (started: 2021-01-05 22:35:06 +00:00)

```
simple_val_loss = H.history['val_loss']
plt.plot([los for los in simple_val_loss])
plt.title('Validation Loss for a InceptionResNetV2 model using No augmentation')
plt.ylabel('Validation error')
plt.xlabel('Epoch')
plt.savefig('/content/drive/MyDrive/CV/Assignment 3/InceptionResNetV2/simple_Validation_loss_')
plt.show()
```

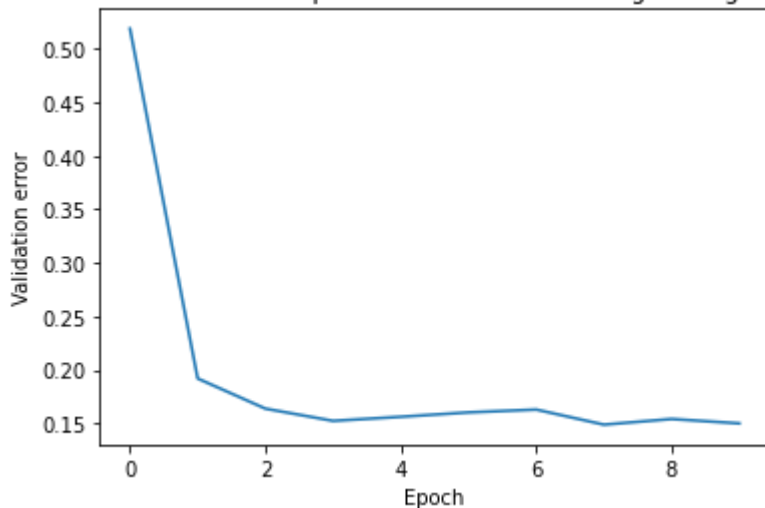


time: 229 ms (started: 2021-01-05 22:35:06 +00:00)

```
simple_val_acc = H.history['val_accuracy']
plt.plot([1 - acc for acc in simple_val_acc])
plt.title('Validation error for a InceptionResNetV2 model using No augmentation')
```

```
plt.ylabel('Validation error')
plt.xlabel('Epoch')
plt.savefig('/content/drive/MyDrive/CV/Assignment 3/InceptionResNetV2/simple_Validation_error')
plt.show()
```

Validation error for a InceptionResNetV2 model using No augmentation



time: 205 ms (started: 2021-01-05 22:35:06 +00:00)

```
print('loading test images')
X_test, y_test = load_data(pred_dir)
y_test = lb.fit_transform(y_test)
```

```
loading test images
100%|██████████| 1128/1128 [05:51<00:00, 3.21it/s]
100%|██████████| 1297/1297 [06:49<00:00, 3.17it/s]
100%|██████████| 1330/1330 [06:38<00:00, 3.34it/s]
100%|██████████| 1166/1166 [05:24<00:00, 3.60it/s]
100%|██████████| 1144/1144 [05:50<00:00, 3.26it/s]
100%|██████████| 1236/1236 [06:21<00:00, 3.24it/s]
time: 37min 6s (started: 2021-01-05 22:35:06 +00:00)
```

```
score = model.evaluate(X_test, y_test, batch_size=64)
print('Test Loss = ', score[0])
print('Test Accuracy = ', score[1])
```

```
115/115 [=====] - 6s 45ms/step - loss: 0.6013 - accuracy: 0.856
Test Loss = 0.601340651512146
Test Accuracy = 0.8502944707870483
time: 6.2 s (started: 2021-01-05 23:12:13 +00:00)
```

```
'''CONFUSION MATRIX'''
# Making prediction
y_pred = model.predict(X_test)
y_true = np.argmax(y_test, axis=-1)

# Plotting the confusion matrix
```



```
from sklearn.metrics import confusion_matrix
confusion_mtx = confusion_matrix(y_true, np.argmax(y_pred, axis=1))
```

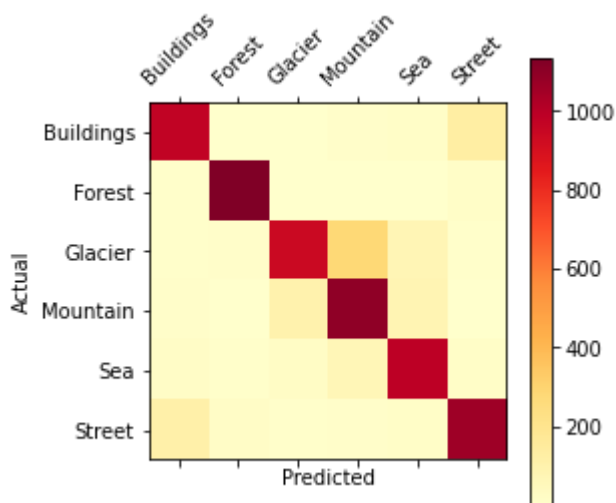
time: 9.32 s (started: 2021-01-05 23:12:19 +00:00)

confusion\_mtx

```
array([[ 975,    5,    1,   14,   17,  132],
       [    6, 1136,    1,    3,    5,   15],
       [    6,   12,  944,  284,   75,    9],
       [   10,    5,   99, 1098,   83,    2],
       [   19,    8,   24,   71,  991,   15],
       [  114,   22,    7,   11,   18, 1064]])
```



```
def plot_confusion_matrix(df_confusion, title='Confusion matrix', cmap=plt.cm.YlOrRd):
    plt.matshow(df_confusion, cmap=cmap) # imshow
    plt.colorbar()
    tick_marks = np.arange(6)
    names = ["Buildings", "Forest", "Glacier", "Mountain", "Sea", "Street"]
    plt.xticks(tick_marks, names, rotation=45)
    plt.yticks(tick_marks, names)
    plt.ylabel("Actual")
    plt.xlabel("Predicted")
#call function
plot_confusion_matrix(confusion_mtx)
```



time: 189 ms (started: 2021-01-05 23:12:28 +00:00)

```
import seaborn as sns
```

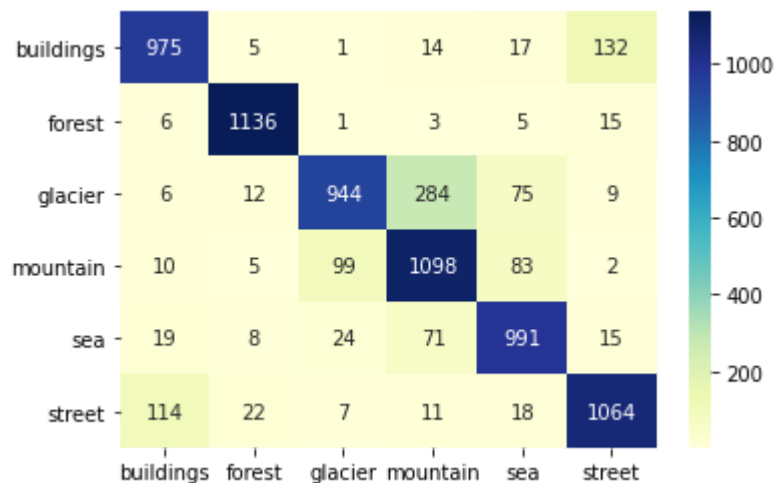
```
class_names = ['buildings','street','forest','glacier','mountain','sea']
```

```
class_names = sorted(class_names)
```

```
sns.heatmap(confusion_mtx, xticklabels=class_names, yticklabels=class_names,
```

```
annot=True, fmt='d', cmap="YlGnBu")
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fe09cfb9dd8>
```



```
time: 506 ms (started: 2021-01-05 23:12:28 +00:00)
```

Google Colab Link:

<https://colab.research.google.com/drive/1VxFnPZbg0gKJU8KXDUHK5Cy3Gg4asQrv?usp=sharing>