Name: Esplanada, Borris

Section: CPE32S1

Instructions:

1. Choose any dataset applicable to an image classification problem
2. Explain your datasets and the problem being addressed.
3. Show evidence that you can do the following:
    - Using your dataset, create a baseline model of the CNN
    - Perform image augmentation
    - Perform feature standardization
    - Perform ZCA whitening of your images
    - Augment data with random rotations, shifts, and flips
    - Save augmented image data to disk
    - Develop a test harness to develop a robust evaluation of a model and establish a baseline of performance for a classification task
    - Explore extensions to a baseline model to improve learning and model capacity.
    - Develop a finalized model, evaluate the performance of the final model, and use it to make predictions on new images.
4. Submit the link to your Google Colab (make sure that it is accessible to me) and the link to your dataset/s

NOTE:

- Submit a well-prepared notebook for your report
- Include conclusion or learning

1. Choose any dataset applicable to an image classification problem

    Dataset Title: The Microsoft Cats vs. Dogs dataset

    Link: https://www.kaggle.com/datasets/shaunthesheep/microsoft-catsvsdogs-dataset?resource=download

2. Explain your datasets and the problem being addressed.

- The Microsoft Cats vs. Dogs dataset, available on Kaggle, is a collection of images of cats and dogs that have been labeled as either a cat or a dog. The dataset contains a total of 25,000 images, with 12,500 images of cats and 12,500 images of dogs. The problem being addressed with this dataset is image classification, specifically distinguishing between images of cats and dogs. This is a well-known and challenging computer vision problem that has many practical applications, such as in animal identification, surveillance, and medical imaging.

3. Show evidence that you can do the following:
    - Using your dataset, create a baseline model of the CNN
    - Perform image augmentation
    - Perform feature standardization
    - Perform ZCA whitening of your images

- Augment data with random rotations, shifts, and flips
- Save augmented image data to disk
- Develop a test harness to develop a robust evaluation of a model and establish a baseline of performance for a classification task
- Explore extensions to a baseline model to improve learning and model capacity.
- Develop a finalized model, evaluate the performance of the final model, and use it to make predictions on new images.

```
"""
Importing required data from kaggle
"""
import os
from google.colab import files
!pip install kaggle
files.upload()

!mkdir ~/.kaggle/
!cp kaggle.json ~/.kaggle/
!chmod 600 ~/.kaggle/kaggle.json
!kaggle datasets list

!mkdir datasets
!cd datasets
!kaggle datasets download -d ifeanyinneji/nike-adidas-shoes-for-image-classification-dataset
!unzip nike-adidas-shoes-for-image-classification-dataset.zip -d datasets
!mkdir aug-train
!mkdir aug-val

# Local Runtime
os.chdir("./datasets")
```

```
Requirement already satisfied: kaggle in /usr/local/lib/python3.10/dist-packages (1.5.16)
Requirement already satisfied: six>=1.10 in /usr/local/lib/python3.10/dist-packages (from kaggle) (1.16.0)
Requirement already satisfied: certifi in /usr/local/lib/python3.10/dist-packages (from kaggle) (2024.2.2)
Requirement already satisfied: python-dateutil in /usr/local/lib/python3.10/dist-packages (from kaggle) (2.8.2)
Requirement already satisfied: requests in /usr/local/lib/python3.10/dist-packages (from kaggle) (2.31.0)
Requirement already satisfied: tqdm in /usr/local/lib/python3.10/dist-packages (from kaggle) (4.66.2)
Requirement already satisfied: python-slugify in /usr/local/lib/python3.10/dist-packages (from kaggle) (8.0.4)
Requirement already satisfied: urllib3 in /usr/local/lib/python3.10/dist-packages (from kaggle) (2.0.7)
Requirement already satisfied: bleach in /usr/local/lib/python3.10/dist-packages (from kaggle) (6.1.0)
Requirement already satisfied: webencodings in /usr/local/lib/python3.10/dist-packages (from bleach->kaggle) (0.5.1)
Requirement already satisfied: text-unidecode>=1.3 in /usr/local/lib/python3.10/dist-packages (from python-slugify->kaggle) (1.3)
Requirement already satisfied: charset-normalizer<4,>=2 in /usr/local/lib/python3.10/dist-packages (from requests->kaggle) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.10/dist-packages (from requests->kaggle) (3.7)
```

Choose Files No file chosen Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

```
import numpy as np
import pandas as pd
```

```
path = "./datasets"
# Load Dataset
df = path
```

2. Show evidence that you can do the following:

- Using your dataset, create a baseline model of the CNN

- Perform image augmentation

- Perform feature standardization

- Perform ZCA whitening of your images

- Augment data with random rotations, shifts, and flips

```
import numpy as np
import os
import cv2
import tensorflow as tf
from keras.preprocessing.image import ImageDataGenerator

# Define paths to the dataset
DATADIR = "./datasets"
CATEGORIES = ["adidas", "nike"]

datagen = ImageDataGenerator(featurewise_center= True, featurewise_std_normalization= True, zca_whitening= True, rotation_range= 50, width_shift_range= 0.5, height_shift_range= 0.5, hori

train = datagen.flow_from_directory('./train', target_size=(150,150),batch_size=20,class_mode='categorical', subset='training')
test = datagen.flow_from_directory('./test', target_size=(150,150),batch_size=20,class_mode='categorical', subset='validation')


# Create a baseline CNN model
model = tf.keras.Sequential([
    tf.keras.layers.Conv2D(32, (3,3), activation='relu', input_shape=(150, 150, 3)),
#    tf.keras.layers.MaxPooling2D(2, 2),
    tf.keras.layers.Conv2D(64, (3,3), activation='relu'),
    tf.keras.layers.MaxPooling2D(2,2),
    tf.keras.layers.Conv2D(128, (3,3), activation='relu'),
#    tf.keras.layers.MaxPooling2D(2,2),
    tf.keras.layers.Flatten(),
    tf.keras.layers.Dense(1024, activation='relu'),
    tf.keras.layers.Dense(512, activation='relu'),
    tf.keras.layers.Dense(2, activation='softmax')
])

# Compile the model
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])

# Train the model
history = model.fit(train, batch_size=32, epochs=10, validation_data= test)

# Evaluate the model on the test set
test_loss, test_acc = model.evaluate(test, verbose=2)
print(f"Test accuracy: {test_acc}")
```

- Save augmented image data to disk

```
import matplotlib.pyplot as plt

augmented_images = train.next()

for i in range(0, 9):
    image = augmented_images[0][i]
    label = augmented_images[1][i]
    plt.subplot(3, 3, i+1)
    plt.imshow(image)
    plt.title(f"Label: {CATEGORIES[np.argmax(label)]}")
    plt.axis('off')
plt.show()
```

```
train = datagen.flow_from_directory('./train', target_size=(150,150),batch_size=20,class_mode='categorical', subset='training', save_to_dir= '../aug-train', save_prefix= 'jpg')
test = datagen.flow_from_directory('./test', target_size=(150,150),batch_size=20,class_mode='categorical', subset='validation', save_to_dir= '../aug-val', save_prefix= 'jpg')

for _ in range(int(np.ceil(train.classes.__len__()/train.batch_size))):
    x = train.next()

for _ in range(int(np.ceil(test.classes.__len__()/test.batch_size))):
    x = test.next()
```

```
Found 460 images belonging to 2 classes.
Found 0 images belonging to 2 classes.
```

- Develop a test harness to develop a robust evaluation of a model and establish a baseline of performance for a classification task

- Explore extensions to a baseline model to improve learning and model capacity.

- Develop a finalized model, evaluate the performance of the final model, and use it to make predictions on new images.

4. Submit the link to your Google Colab (make sure that it is accessible to me) and the link to your dataset/s

Link Google Colab: https://colab.research.google.com/drive/1z9shGZxqS5miZZQMCWbv5POHjg-DlDj3#scrollTo=J7u1vJjlNA8w

Link of my Dataset: https://www.kaggle.com/datasets/shaunthesheep/microsoft-catsvsdogs-dataset

Conclusion

In conclusion convolutional neural networks, or CNNs, are an effective tool for image recognition and classification applications. It is sometimes advantageous to preprocess photos using methods like ZCA whitening before feeding them into a CNN in order to reduce input redundancy and enhance network performance. Adding random rotations, shifts, and flips to the training dataset can also help the network become more resilient to input changes and better able to generalize to new data. CNNs may achieve high accuracy and significantly improve a variety of applications, from autonomous driving to medical imaging, by utilizing these approaches.