Using Convolutional Neural Networks to Identify Dog Breeds

Domain Background

Nowaday, AI and computer vision play an important role in detecting objects [1]. There are many models pretrained with a big dataset for this task. For example, there are models that can identify objects in the ImageNet dataset which have 1000 classes [2].
Howevers, there are limitations that these pretrained models can identify only the classes that appear in the trained dataset. For example, in order to specifically identify unseen classes like some dog breeds, the models need to be newly created. Also, there may be situations where there are only a few dog images that have the breed labels because it requires a lot of effort to label images [3]. Thus, the model must use only a few images to train.

Problem Statement

With only a few dog images with breed labels, it is challenging to create a model that does not overfit the training set and give high accuracy on the test set.

Datasets and Inputs

1. Human dataset:
   (https://s3-us-west-1.amazonaws.com/udacity-aind/dog-project/lfw.zip)
   This dataset contains human faces and will be used just for testing the human detector algorithm. For this project, we will use a face detection algorithm to detect the human face. As we know that existing face detection algorithms can work pretty well so this task is not the main focus for this project.

2. Dog dataset:
   (https://s3-us-west-1.amazonaws.com/udacity-aind/dog-project/dogImages.zip)
   This is the main dataset for developing a CNN model to identify dog breeds. The dataset contains dog images with 133 breed labels. The dataset consist of training (6680 images), validation (835 images), and test (836 images) sets

Solution Statement

In order to achieve high accuracy from fitting the model with only a few images, transfer learning must be used. Also, the chosen base models should be trained on the dataset that contain some different dog breeds. Fortunately, all base models from torchvision are trained on the ImageNet dataset which contains some image categories corresponding to dogs [2]. Thus, one of these base models will be chosen for transfer learning in this project.

Pseudo code of using transfer learning to create dog-breed classification model

```
import os
import torch.nn as nn
import torch.optim as optim
from torchvision import datasets
import torchvision.models as models


### TODO: Write data loaders for training, validation, and test sets
use_cuda = 1
loaders_transfer, data_transfer = create_loader()


## TODO: Specify model architecture
model_transfer =  models.selected_model(pretrained=True)
model_transfer.classifier[6].out_features = len(data_transfer['train'].classes)
for param in model_transfer.features.parameters():
    param.requires_grad = False

if use_cuda:
    model_transfer = model_transfer.cuda()

criterion_transfer = nn.CrossEntropyLoss()
optimizer_transfer = optim.SGD(model_transfer.parameters(), lr = 0.001, momentum = 0.9)


model_transfer = train(100, loaders_transfer, model_transfer, optimizer_transfer,
criterion_transfer, use_cuda, 'model_transfer.pt')

# load the model that got the best validation accuracy (uncomment the line below)
model_transfer.load_state_dict(torch.load('model_transfer.pt'))
```

Benchmark Model

There are some works that use the same datasets to create and test the models [4, 5] and give the similar benchmark results. For example, Maanav Shah created 4 models including VGG-49, ResNet-50, Inception V3, and Xception [4]. The model performance comparisons of his work are as follows.

Model	**Acc**	**Training Time in seconds (20 epochs, batch size = 32)**
VGG-19	46.0526%	24.16
ResNet-50	82.4163%	18.41
Inception v3	80.5024%	31.83
Xception	84.6890%	49.07

Also, there is some research that solves a similar problem to this project but with different dataset [6].

Evaluation Metrics

Accuracy is the acceptable and selected metric because classes are not extremely unbalanced.

Project Design

Solution is to write an algorithm that accepts a file path to an image and first determines whether the image contains a human, dog, or neither. Then, if a dog is detected in the image, return the predicted breed. if a human is detected in the image, return the resembling dog breed. if neither is detected in the image, provide output that indicates an error.
Main focus is to improve the accuracy of the CNN model that identifies dog breeds. The model architecture or parameters will be modified iteratively to achieve at least 60% accuracy on the test set.

Pseudo code of the solution and examples of the expected outcomes

```
def run_app(img_path):
    ## handle cases for a human face, dog, and neither

    img = cv2.imread(img_path)
    if dog_detector(img_path):
        breed = predict_breed_transfer(img_path)
        print(f'''Hi, {breed}''')
    elif face_detector(img_path):
        breed = predict_breed_transfer(img_path)
        cv_rgb = cv2.cvtColor(img, cv2.COLOR_BGR2RGB)
        print(f'''Hello, human, You look like a ... \n {breed}''')
    else:
        raise Exception("Neither is detected in the image")
    plt.imshow(cv_rgb)
    plt.show()
```

Hi, Affenpinscher 	Hi, Brittany

Reference

1. Medium. 2021. *Everything You Ever Wanted To Know About Computer Vision. Here's A Look Why It's So Awesome.*. [online] Available at: <https://towardsdatascience.com/everything-you-ever-wanted-to-know-about-computer-vision-heres-a-look-why-it-s-so-awesome-e8a58dfb641e> [Accessed 20 April 2021].

2. Pytorch.org. 2021. *torchvision.models — Torchvision master documentation*. [online] Available at: <https://pytorch.org/vision/stable/models.html> [Accessed 20 April 2021].

3. Anolytics. 2021. *Top Data Labeling Challenges Faced by the Data Annotation Companies*. [online] Available at: <https://www.anolytics.ai/blog/top-data-labeling-challenges-faced-by-annotation-companies/> [Accessed 20 April 2021].
4. Medium. 2021. *Dog Breed Classifier using CNN*. [online] Available at: <https://medium.com/@maanavshah/dog-breed-classifier-using-cnn-f480612ac27a> [Accessed 20 April 2021].
5. Medium. 2021. *Deep Learning: Build a dog detector and breed classifier using CNN*. [online] Available at: <https://towardsdatascience.com/deep-learning-build-a-dog-detector-and-breed-classifier-using-cnn-f6ea2e5d954a> [Accessed 20 April 2021].
6. Ayanzadeh, Aydin & Vahidnia, Sahand. (2018). Modified Deep Neural Networks for Dog Breeds Identification.