CSCI 4050U - Final Project: Animals-10 Classification

By: Pratham Patel, Sachmilansingh Gangarh, Lesley Ozurigbo

Group ID: 19

Overview

Animals-10 Dataset

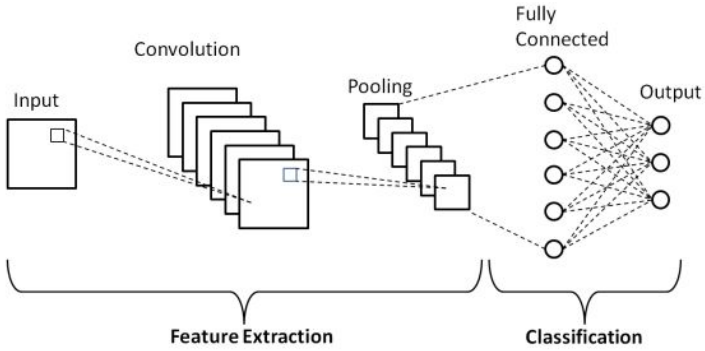
- The dataset we used was pulled from Kaggle: https://www.kaggle.com/datasets/alessiocorrado99/animals10?resource=download
- Contains image data of 10 different species of animals
- Approximately 26K images
- The data is slightly imbalanced, but we will address this later

Learning Problem

- Our goal was to experiment with a variety of machine learning models and techniques, and compare their accuracies. We selected techniques and models that would be feasible given the computational constraints.
- These were the techniques/models we used:
  - Deep Convolutional Neural Networks
  - Transfer learning on pretrained models
  - Image augmentation
  - Weighted loss functions

Model Development & Training

Training a Deep CNN

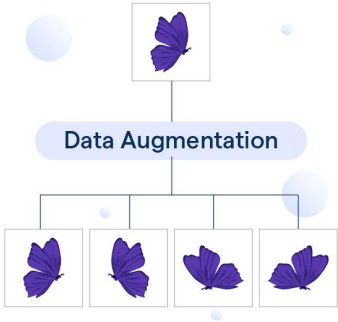


- We started by training a convolutional neural network with the following architecture:
  - Conv2D > ReLU > Max Pooling > Conv2D > ReLU > Max Pooling > Flatten > Linear Layer
- We had limited computational resources, so the model was quite small
  - The model accepted a 128x128 input image
  - 42K trainable parameters
  - Trained it for 3 epochs
- This model produced a 40% accuracy on validation data. It performs much better than randomly guessing, but still far from ideal.

Transfer Learning: MobileNetV3

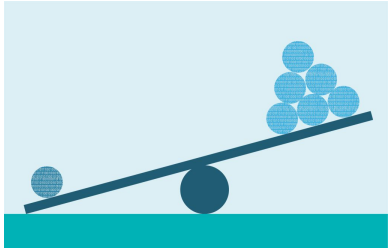
- Next, we used a lightweight pretrained model, MobileNetV3 as a starting point
  - We replaced the final layer with a linear layer with 10 outputs, to match the number of classes in the dataset
  - We also removed gradient tracking from all layers except for the final layer
  - This technique is called **transfer learning**
- We trained it for 3 epochs, and it went by much faster than training the model from scratch since only the last layer was being trained
- We achieved much better results this time (92% accuracy)

Transfer Learning: ResNet18 + Data Augmentation



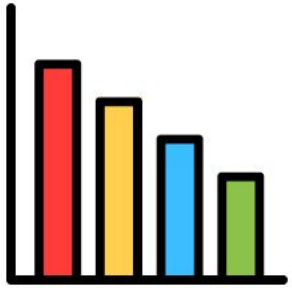
- This time we decided to use a larger pretrained model, ResNet18
- We also incorporated a data preprocessing technique called data augmentation
- Data augmentation modifies the images by randomly applying some set of transformations on it
    - For example: rotations, flips, crops, brightness changes
- Benefits include:
    - Generate more data (helps for data imbalance or shortage of data)
    - Prevent model overfitting
- We achieved an accuracy of 94% for this model

Transfer Learning: ResNet18 + Data Augmentation & Weighted Loss



- Lastly, we added a weighted loss function to our ResNet18 model
- This is one way to resolve the data imbalance problem in our dataset
- Classes that have less data will be weighted higher for the loss function. Classes that have more data will be weighted less.
    - For example: We have 4863 images of dogs, but only 1446 images of elephants, so the elephants will be weighted proportionally higher when calculating loss
- By incorporating all these techniques, we achieved a 96% model accuracy

Model Comparison Overview



Overall, these are our results:

- Deep convolutional neural network trained from scratch: **40% accuracy**
- MobileNetV3 transfer learning: **92% accuracy**
- ResNet18 transfer learning + data augmentation: **94% accuracy**
- ResNet18 transfer learning + data augmentation & weighted loss: **96% accuracy**

Deployment

ReactJs Web App

- We developed a React web application to deploy our model
- In this app a user inputs an image of one of the animals classes we trained on. Then the model makes a prediction on which animal it is.