Classifying Interesting/ Not-Interesting in an Image

Final Project Machine Learning II
Jennifer Crosby
5/1/2020

Introduction

- NOAA spends billions of dollars on expeditions, collecting so much data that much goes unanalyzed due to the enormity and difficulty in finding specific information in hundreds of hours of videos, plus the time needed to spend to watch the videos

Purpose

- This project identifies interesting/not-interesting in images, and is the first step in ultimately classifying video live from an ROV filming from the ocean floor
- Just this basic identification will cut a 20 hour video down to an hour of viable images
- Ultimately, Species classification is the goal, emailing the scientist about where in the video, is the species they are interested in studying

Dataset

- Obtained with authorization from NOAA Federal to download images from a deep ocean expedition
  - Images in 1 second intervals
  - Manually classified 1500 images

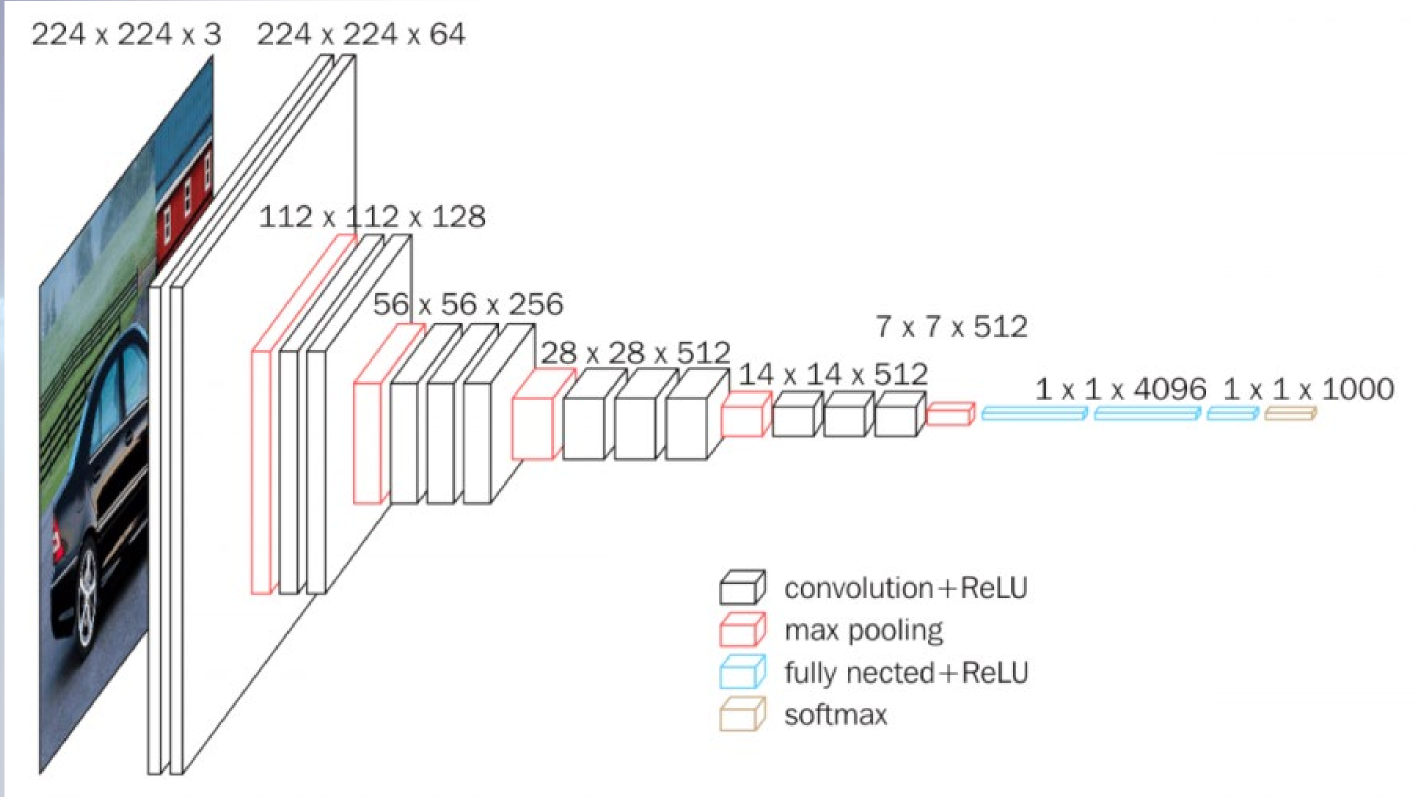
Experimental Setup

- Images will be put through a Convolution Neural Network using a pretrained model, VGG16.
- Adam is the optimizer used
- BCE with logits Loss will be used

Setup (cont)

- Training/Testing split is 80/20
- Batch sizes are 16
- 1500 images
- Original size 1920 (w) x 1080 (h)
- 24 bit depth

VGG16 – Convolutional Network for Classification and Detection



https://neurohive.io/en/popular-networks/vgg16/

VGG16 – Convolutional Network for Classification and Detection

- The model achieves 92.7% top-5 test accuracy in ImageNet, which is a dataset of over 14 million images belonging to 1000 classes.

https://neurohive.io/en/popular-networks/vgg16/

VGG16 – Convolutional Network for Classification and Detection

ConvNet Configuration					
A	A-LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv3-64	conv3-64 **LRN**	conv3-64 **conv3-64**	conv3-64 conv3-64	conv3-64 conv3-64	conv3-64 conv3-64
maxpool					
conv3-128	conv3-128	conv3-128 **conv3-128**	conv3-128 conv3-128	conv3-128 conv3-128	conv3-128 conv3-128
maxpool					
conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256	conv3-256 conv3-256 **conv1-256**	conv3-256 conv3-256 **conv3-256**	conv3-256 conv3-256 conv3-256 **conv3-256**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512	conv3-512 conv3-512 **conv1-512**	conv3-512 conv3-512 **conv3-512**	conv3-512 conv3-512 conv3-512 **conv3-512**
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

https://neurohive.io/en/popular-networks/vgg16/

Results

30 epochs/400 images

```
Validation Loss 0.56569
Validation Loss 0.56560
Validation Accuracy: 78.0
```

Overfitting was probably occurring, thus the better accuracy

30 epochs/1500 images

- 68% accuracy

Conclusion

- VGG16 had a marked improvement over my first model, and reached 78% accuracy initially
- Improvements could be made by adding another model to this, like AlexNet
- More data would help as well