ANIMAL DETECTION USING R-CNN

Author(s): John Jajeh, Masood Abdul Salam – Advisor(s): Dr. Mingon Kang

Abstract
By applying regions on CNN features, R-CNN provides computer vision solutions for multiple-object detection. In our research, we are utilizing AlexNet’s pretrained model in the Caffe framework to detect approximately 400 different animal species and are acclimating this work from KSU’s GPU server to the Android environment. After an individual downloads the application and an animal is detected, he/she can click on the animal, which will prompt Google to search the animal label. Essentially, this app will allow users to photograph unfamiliar (or familiar) animals for identification and better personal understanding.

Definitions
• Region-based Convolutional Neural Networks (R-CNN): a visual object detection system that combines bottom-up region proposals with rich features computed by a convolutional neural network
• Convolutional Neural Network (CNN): a class of deep, feed-forward artificial neural networks that has successfully been applied to analyzing visual imagery
• AlexNet: the name of a convolutional neural network
• Caffe: a deep learning framework
• Object Detection: a computer technology related to computer vision and image processing that deals with detecting instances of semantic objects of a certain class (such as humans, buildings, or cars) in digital images and videos
• Pattern Recognition: a branch of machine learning that focuses on the recognition of patterns and regularities in data
• Selective Search: a process by which a number of regions are chosen, which create images for object detection

Materials
GPU server w/ Pre-trained Alexnet, Caffe, FileZilla, Putty, NotePad++, Jupyter Notebook (w/ Python 3.6), Android Device

Methods
• Acquire data to test the pre-trained model.
• Create a memory-mapped database file.
• Create mean image values.
• Test to acquire results for labels and scores. Modify code as necessary and repeat testing until desired results are found.
• Acclimate into Android environment and grant Google Search capability.
• Test and modify accordingly.

Visual Layout
First, we chose an image containing many animals and used Selective Search to distinguish areas for animal detection. Here, we have an example of a chosen image – a chimpanzee. After accurately identifying the animal (producing a score > 0.5), we know that testing this in the Android environment will produce similar results.

Expected Results
We expect to complete our animal detection app called “Animal Detection” by July 1, 2018. Users can expect to use this technology on Android phones and tablets to detect approximately 400 different animals; some of which include:
• Bald eagle
• Stingray
• Terrapin
• Scorpion
• Tarantula
• Peacock
• Snaill
• Wombat
• Lion
• Hippo
• Hamster
• Porcupine
• Grasshopper
• King Penguin
• Jellyfish
• Indian cobra
• Ostrich
• Vulture

After completing the back-end computing portion of this project, we plan on acclimating towards the Android environment using Android Studio (Java) or Visual Studio (C#). Our product will contain no user interface.

Keywords
• R-CNN
• CNN
• AlexNet
• Object Detection
• Pattern Recognition
• Pattern Recognition
• Caffe
• Python
• Jupyter Notebook

Acknowledgments
We would like to thank Dr. Mingon Kang, Rehnuma Afrin, and Kritika Garg for all their guidance, support, and assistance in making this project possible.

Contact Information
John Jajeh – jajeh@gmail.com
Masood Abdul Salam – masood.anzar2010@gmail.com

References
• A Practical Introduction to Deep Learning with Caffe and Python by Adil Moujahid
• ImageNet Classification with Deep Convolutional Neural Networks by Alex Krizhevsky, Bya Sutskever, and Geoffrey E. Hinton
• Rehnuma Afrin (rafain@students.kennesaw.edu)