Challenge 6: Detecting Safety

Background

AdventureWorks has partnered with a local guide service to help them collect images of mountaineers and climbers wearing helmets in an effort to encourage the use of helmets for safety. AdventureWorks wants a solution that can locate every helmet present in an image.

Object detection adds a layer of complexity and usefulness on top of classification. It predicts whether something is present, and where in the image it is located. This is important if the model needs to identify multiple instances of a class in a given image.

Prerequisites

* The new helmet dataset (a list of image URLs for training and testing)
  + Cloud setup ! curl -O https://challenge.blob.core.windows.net/challengefiles/summit\_post\_urls\_selected.txt
  + Local setup [Download](https://challenge.blob.core.windows.net/challengefiles/summit_post_urls_selected.txt)
* Team decides upon a deep learning framework to use in this Challenge, after reading about several different frameworks (See [References](https://openhacks.azurewebsites.net/labs/player/microsoft-open-hack-ai" \l "references" \t "_blank))
* Team performs any installs or updates to the latest versions of frameworks in their chosen setup

Challenge

Obtain the images from the URL list in the given file in a format conducive to the framework of choice.

Using the deep learning framework, create an object detection solution utilizing a modern model like Faster R-CNN. This model should be able to detect and create a bounding box around each helmet present in an image.

Success Criteria

* Demonstrate logging or TensorBoard output from your deep learning solution showing a maximum Mean Average Precision (mAP) over 80%

References

Read this first

* What is object detection [Ref](https://tryolabs.com/blog/2017/08/30/object-detection-an-overview-in-the-age-of-deep-learning/" \t "_blank)
* What is mAP [Ref](http://fastml.com/what-you-wanted-to-know-about-mean-average-precision/" \t "_blank)

Tooling

* Visual Object Tagging Tool VoTT. Works for TensorFlow and CNTK [Ref](https://github.com/Microsoft/VoTT" \t "_blank)
* When on Linux, the Tensorflow Object Detection API [Ref](https://github.com/tensorflow/models/tree/master/research/object_detection" \t "_blank)
* CNTK Documentation [Ref](https://www.microsoft.com/en-us/cognitive-toolkit/" \t "_blank)

Hints

* When the team is planning for image processing, explore what functionality your deep learning framework offers