Infosys®

POWERED BY INTELLECT
DRIVEN BY VALUES

A general approach of transfer learning in computer vision to classify objects in videos

Presented by Kitu Komya
On behalf of Nidhi Tiwari

September 7, 2018

Background

- Computer Vision is a field in which computers are taught to see, identify, and process images the same way humans do
- Object Detection locates and identifies objects such as bicycles, humans, or cats
- Transfer Learning is a machine learning technique where a model trained on one task is re-purposed on a second, related task
- Convoluted Neural Networks (CNNs) are used in object detection

Overview

Problems

Models can only classify a limited set of classes (perhaps cat and dog, but not aardvark or rhinos)

Creating new models from scratch is time-consuming

Solution

Use transfer learning to classify new objects and to build upon pre-existing models

Impacts

Can detect any type of object quickly

Solution is general and can thus be applied to any situation

Goals

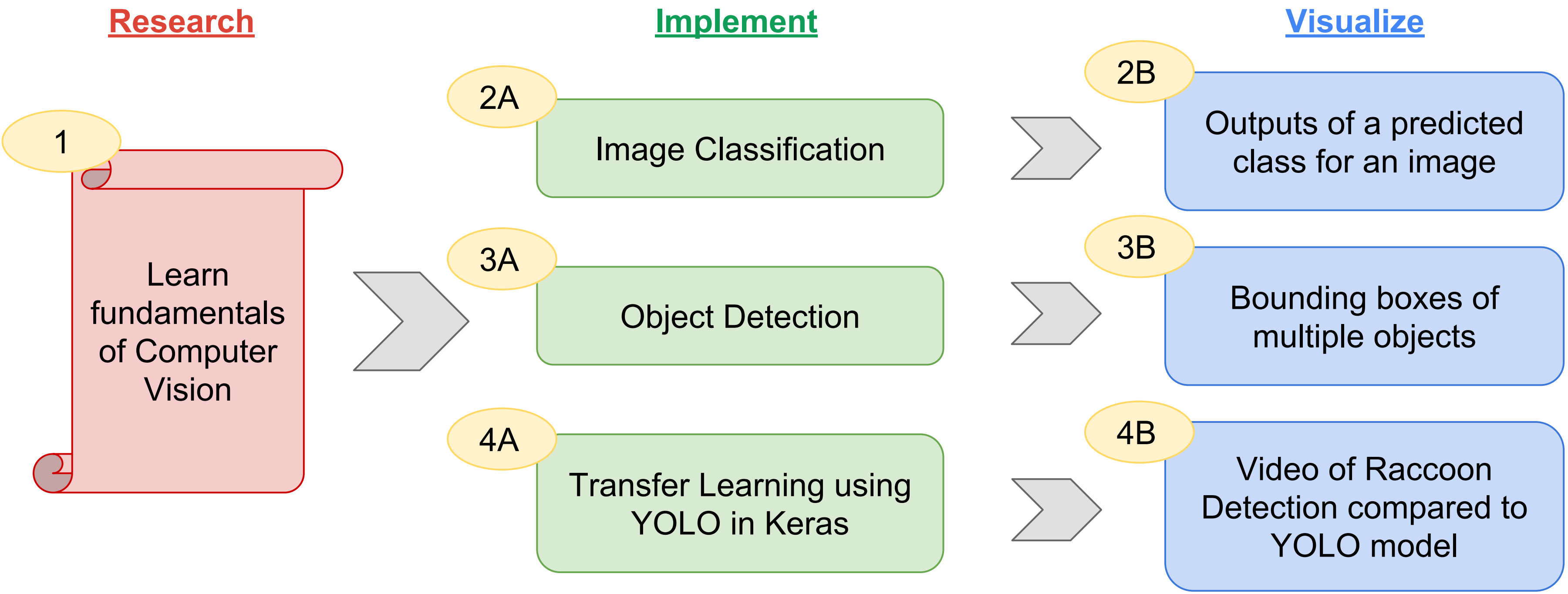
1. Learn the fundamentals and vocabulary of **Computer Vision**

2. Implement **Image Classification** algorithms

3. Implement **Object Detection** algorithms

4. Combine results with **Transfer Learning** to detect objects in videos

Workflow


Research
Implement
Visualize
1
Learn fundamentals of Computer Vision
2A
Image Classification
3A
Object Detection
4A
Transfer Learning using YOLO in Keras
2B
Outputs of a predicted class for an image
3B
Bounding boxes of multiple objects
4B
Video of Raccoon Detection compared to YOLO model

Research fundamentals of Computer Vision

About Me

- ➢ Statistics background in R
- ➢ Switched from a Bayesian Statistics project to Computer Vision
- ➢ Zero to Hero? Almost…

Research

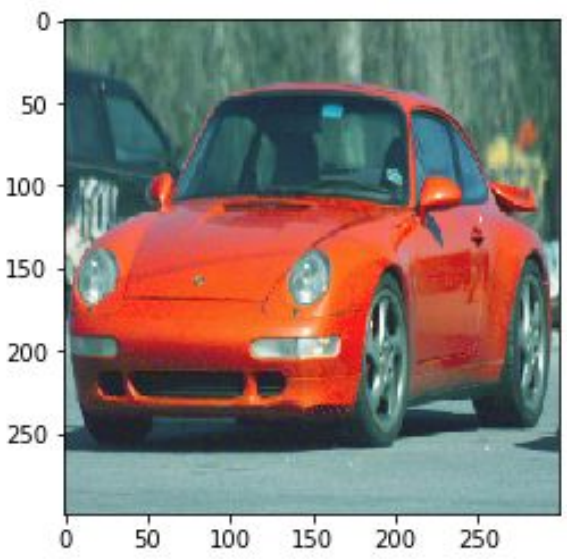
- ➢ Neural Networks
- ➢ Image Classification (VGG16, InceptionV3, ResNet50, MobileNet)
- ➢ Object Detection (Mask R-CNN, YOLO)
- ➢ Transfer Learning (in Keras)

Implementing Image Classification

Tried a few models of varying sizes, parameters, and top-5 accuracies that were validated from ImageNet dataset that classifies 1000 classes

Model Name	Size	Parameters	Top-5 Accuracy
VGG16	528 MB	138,357,544	0.901
InceptionV3	92 MB	23,851,784	0.944
ResNet50	99 MB	25,636,712	0.929
MobileNet	17 MB	4,253,864	0.871

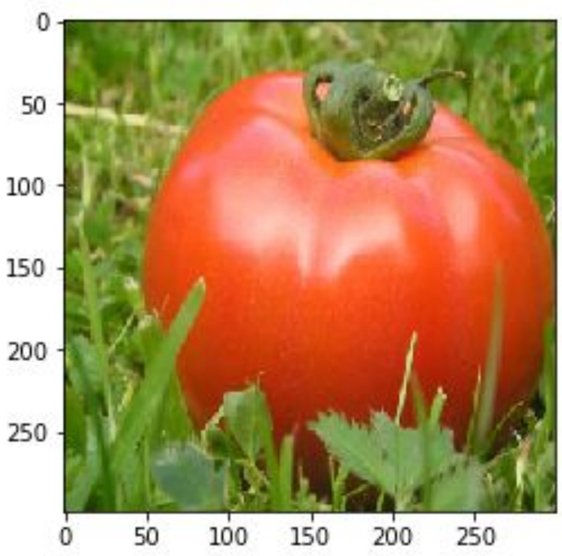
Visualizing Image Classification



InceptionV3	ResNet50	MobileNet
Sports Car (0.78)	Sports Car (0.66)	Sports Car (0.84)
Grille (0.08)	Racer (0.15)	Racer (0.06)
Racer (0.04)	Grille (0.14)	Cab (0.06)
Car Wheel (0.02)	Car Wheel (0.02)	Car Wheel (0.02)
Convertible (0.01)	Convertible (0.01)	Grille (0.01)

All methods resize images...values don’t add to 1.00 because rounding and because top 5 predictions only

Visualizing Image Classification



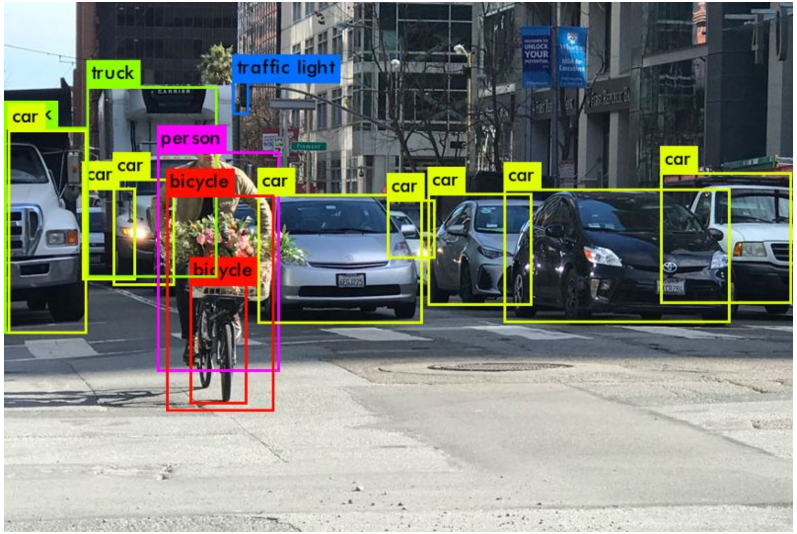
InceptionV3	ResNet50	MobileNet
Bell Pepper (0.53)	Croquet Ball (0.21)	Macaw (0.18)
Hip (0.08)	Bell Pepper (0.11)	Hip (0.16)
Pot (0.01)	Zucchini (0.05)	Croquet Ball (0.14)
Croquet Ball (0.01)	Agaric (0.05)	Boxer (0.08)
Balloon (0.01)	Spaghetti Squash (0.04)	Snail (0.06)

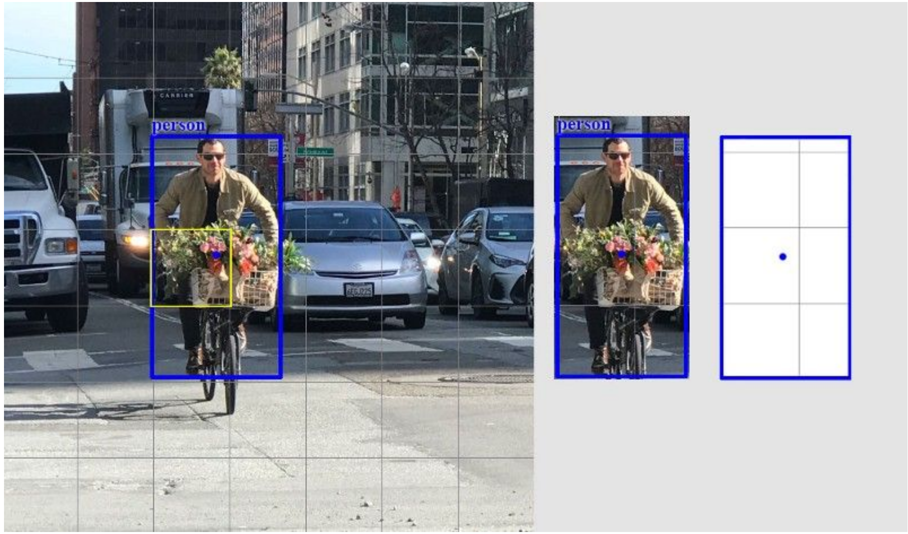
Tomato not a labelled class...at least low probability scores, but still...need transfer learning!

Implementing Object Detection

- Image Classification only detects one object per image
  - Most images are bounded by multiple objects
- With huge help of Gaston Bizel, implemented **YOLO** (You Only Look Once) detection in Keras
  - YOLO divides input into a **square grid**
  - Makes a prediction on each grid
  - Makes a predicted **bounding box** on each grid
  - Need training data with bounding boxes
  - Works well because it's generalized and fast

Visualizing Object Detection





YOLO can predict multiple objects

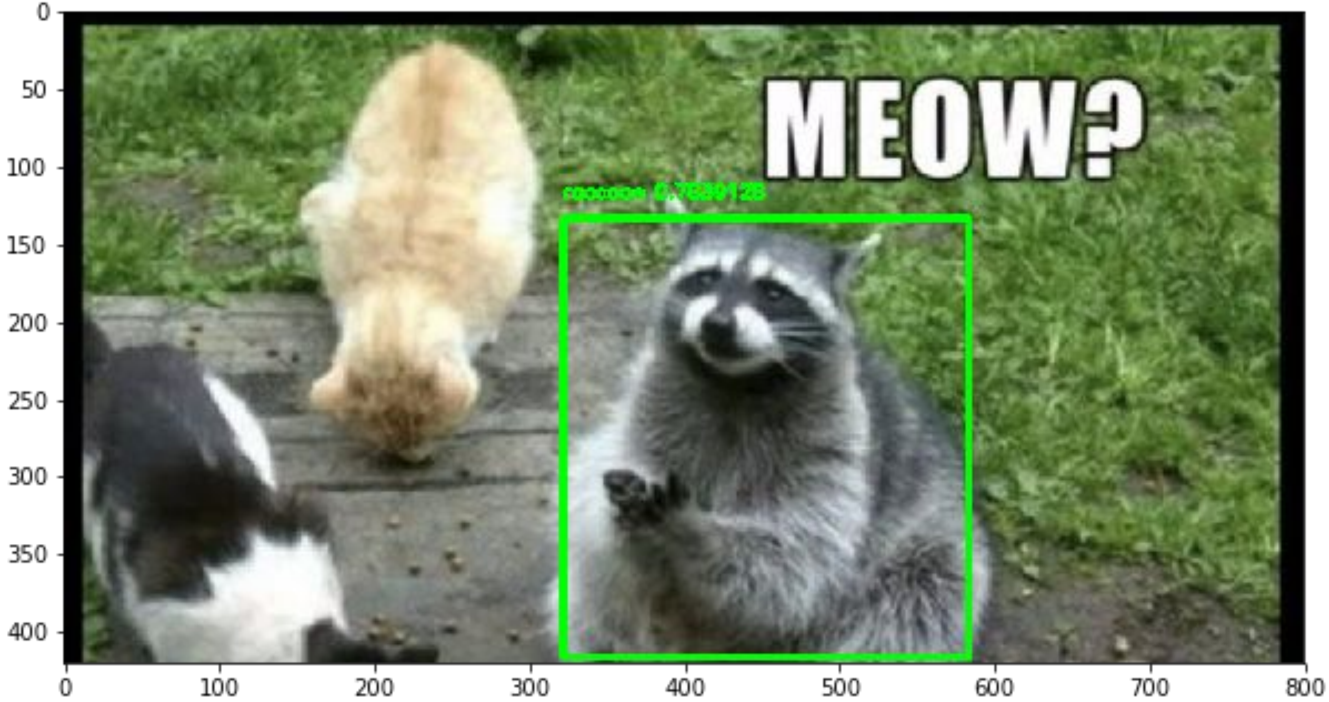
Will make prediction grid by grid...hence need training data that is labelled as such

Implementing Transfer Learning

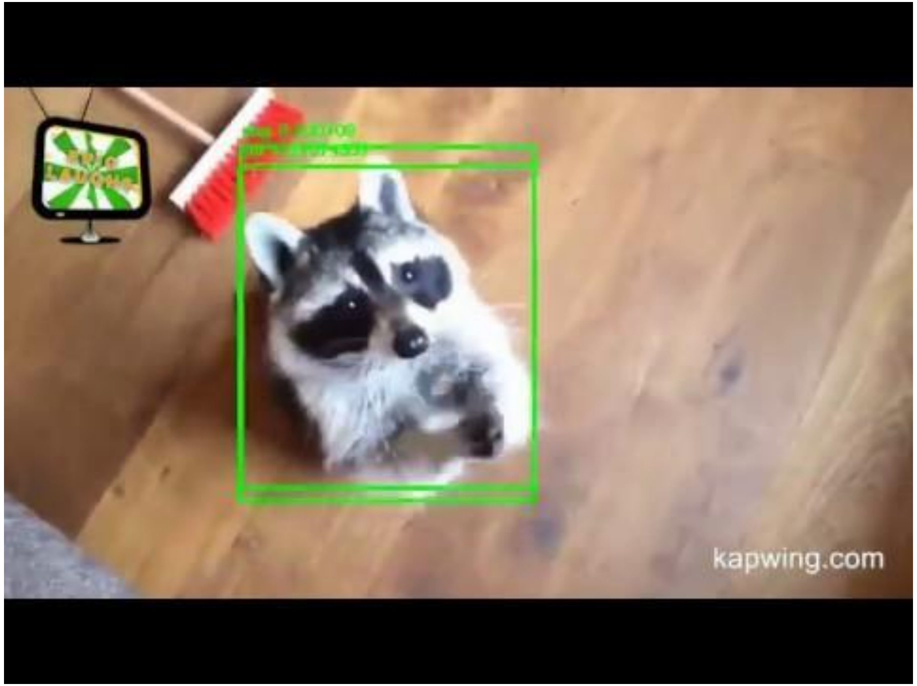
- ➢ Build upon YOLO detection by teaching model to **identify new classes**
- ➢ Building model from scratch very time-consuming, so **using layers of YOLO's pretrained weights** until last one
- ➢ For demo, using Raccoon detection because found **labelled dataset of bounding boxes**
- ➢ Can apply this general technique to any object (spiders, phones, donuts)


```
<?xml version="1.0"?>
- <annotation verified="yes">
    <folder>images</folder>
    <filename>raccoon-1.jpg</filenan
    <path>/Users/datitran/Desktop
  - <source>
      <database>Unknown</databa
    </source>
  - <size>
      <width>650</width>
      <height>417</height>
      <depth>3</depth>
    </size>
    <segmented>0</segmented>
  - <object>
      <name>raccoon</name>
      <pose>Unspecified</pose>
      <truncated>0</truncated>
      <difficult>0</difficult>
    - <bndbox>
        <xmin>81</xmin>
        <ymin>88</ymin>
        <xmax>522</xmax>
        <ymax>408</ymax>
      </bndbox>
    </object>
</annotation>
```

Visualizing Transfer Learning

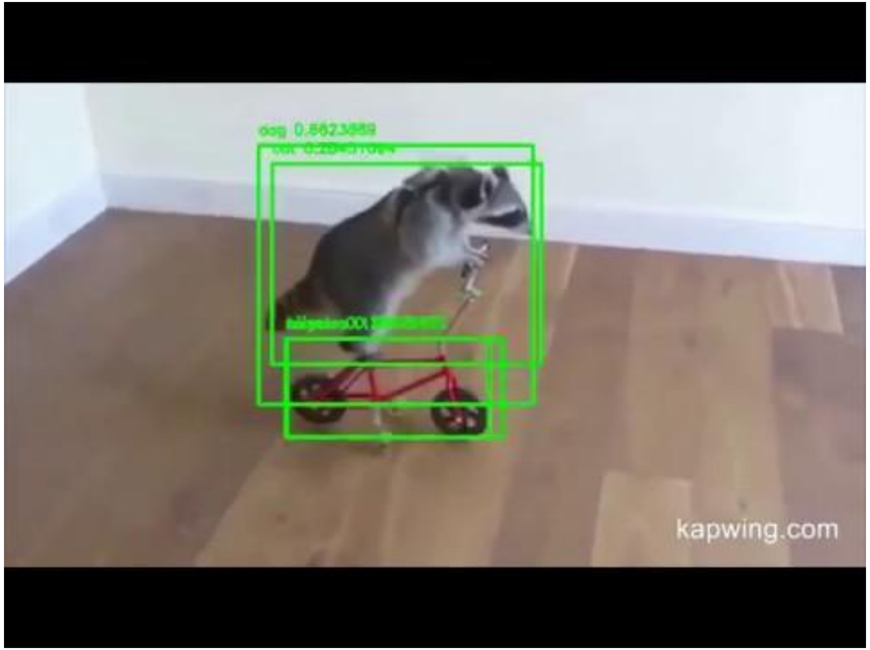


Visualizing Transfer Learning



Side by side of YOLO detector (left) and Raccoon detector (right).

Visualizing Transfer Learning



Side by side of YOLO detector (left) and Raccoon detector (right).

Summary

- Started with image classification
- Used object detection to bound multiple objects
- Implemented transfer learning to add different classes
    - Training on Raccoon dataset led to a Raccoon detector

Next Steps

- ➢ Use Transfer Learning to create model that can detect the original + trained classes
- ➢ Measure speed of objects, number of objects, time of objects and analyze these statistics
- ➢ Create online tool for ease of usability

Acknowledgements

Thank you for your time in listening to my presentation.

I would also like to acknowledge Infosys' lab resources and Ms. Nidhi Tiwari for supporting me in my endeavors and for giving me the opportunity to explore computer vision, despite my having no background in it.