PILES OF PICTURES

PROCESS WITH TURI CREATE

@donmowry


MLMODEL

WHAT WE'LL DISCUSS

Journey

Process

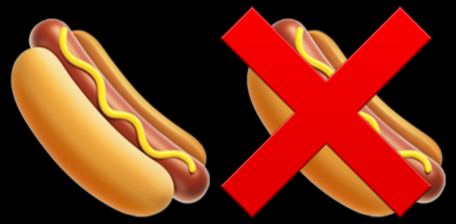
Lessons

WWDC 2017

Vision: Common computer vision tasks like face detection, facial landmarks, object tracking

CoreML: Easily integrate Machine Learning models into your applications

ARKit: Augmented Reality in your application



ROSETTA STONE

Immersion: Learn languages as you learned your first

In-app video

Words attached to real-world objects

Objects recognized…

Words augmenting reality…

I mean…

MACHINE LEARNING DEMOCRATIZED

Deep Learning, Neural Networks, many layers

CUDA , OpenCL: GPU parallelized matrix computations

Tools: TensorFlow PyTorch Keras Caffe MXNet

Many hours, much data, much power to train models

Datasets: ImageNet, Open Images, COCO

Pre-trained models: ResNet, DarkNet, YOLO

INNOVATION SPRINT

September 2017

Pre-trained models: Inception v3, ResNet50, SqueezeNet

Object detection vs. recognition

Detection of “classes” of objects: nouns

ARKit

INNOVATION SPRINT

September 2017

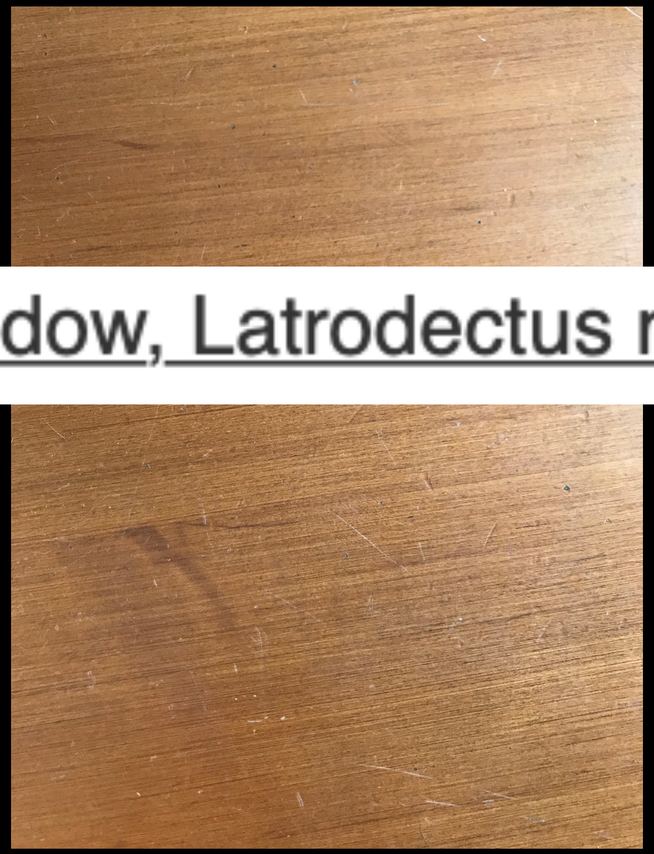
1 week

Decent recognition

ARKit depth placement difficulty

Far too many classes, filtered in code

ALARMING FALSE POSITIVES



INNOVATION SPRINT

December 2017

Focused subset: a challenge

Scenario: Classroom, ELL students

Bring your world to your learning

Local sharing of a challenge

Focused subset: Transfer learning

360|iDev

TRANSFER LEARNING

Similar tasks can reuse existing knowledge

Use pre-trained model, for example a CNN trained on ImageNet

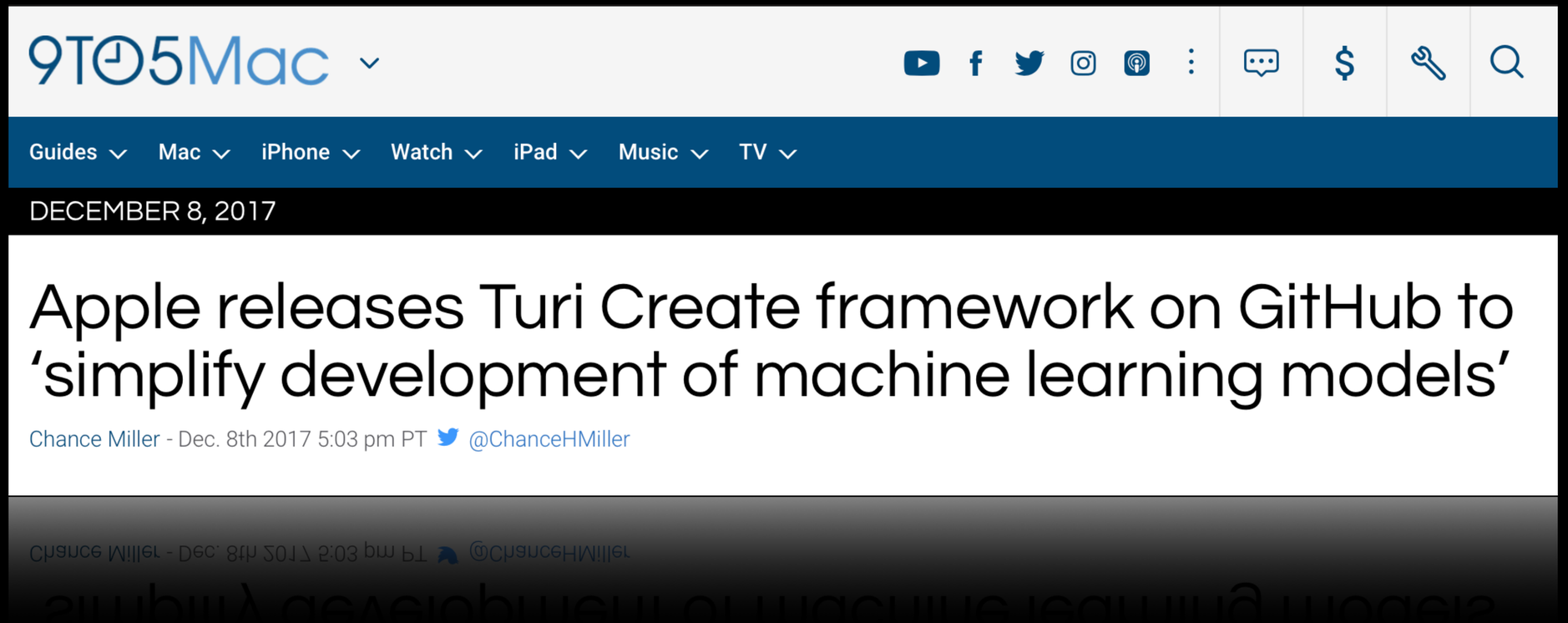
Theory: lower levels in network contain more low-level features

Higher levels contain more complex details

Transfer learning keeps low levels in the network and tunes upper levels on new data

Much faster to train a model

Lots o' math


9TO5Mac
Guides
Mac
iPhone
Watch
iPad
Music
TV
DECEMBER 8, 2017
Apple releases Turi Create framework on GitHub to ‘simplify development of machine learning models’
Chance Miller - Dec. 8th 2017 5:03 pm PT @ChanceHMiller

TURI CREATE

https://github.com/apple/turicreate

Python Library

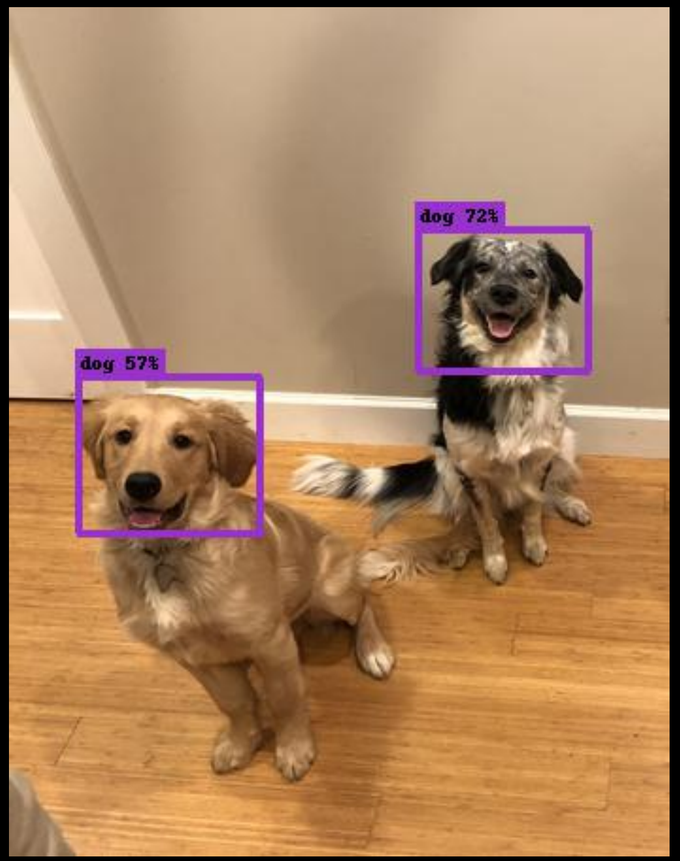
Many common tasks

Object Detection: Image and location

At least 30 images /annotations

Annotation: What and where

SFrame data structure: image and annotation



source:
https://apple.github.io/turicreate/docs/userguide/object_detection/

TURI CREATE OBJECT DETECTION

Based on TinyYOLOv3:

https://pjreddie.com/darknet/yolo/

Redmon, J., & Farhadi, A. (2018). YOLOv3: An Incremental Improvement. *ArXiv*.

Transfer Learning

INTRODUCTORY EXAMPLE

```
import turicreate as tc

# Load the data
data =  tc.SFrame('ig02.sframe')

# Make a train-test split
train_data, test_data = data.random_split(0.8)

# Create a model
model = tc.object_detector.create(train_data)

# Save predictions to an SArray
predictions = model.predict(test_data)

# Evaluate the model and save the results into a dictionary
metrics = model.evaluate(test_data)

# Save the model for later use in Turi Create
model.save('mymodel.model')

# Export for use in Core ML
model.export_coreml('MyCustomObjectDetector.mlmodel')
```

source: https://apple.github.io/turicreate/docs/userguide/object_detection/

FEATURE

FEATURE

Seek and Speak

Key: Integrate *your own images* into conversations

Object Detection

Focused Challenges

Around 20 classes

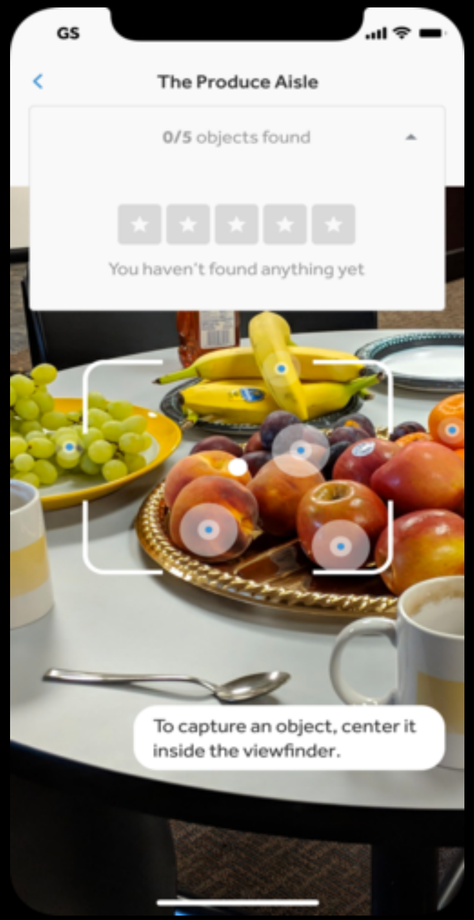


IMAGE COLLECTION FOR TRAINING

Open Images: https://storage.googleapis.com/openimages/web/download.html

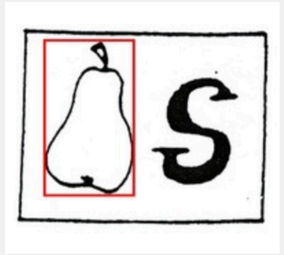
Wide variety of quality

Huge volume of data = way too much time training

Mislabeling

Illustrations

LESSON

Open_Images_Upload		○ Approved ○ Rejected	{"label"=>"pear", "type"=>"rectangle", "coordinates"=> {"x"=>183, "y"=>252, "width"=>195, "height"=>338}}
Open_Images_Upload		○ Approved ○ Rejected	{"label"=>"lemon", "type"=>"rectangle", "coordinates"=> {"x"=>378, "y"=>226, "width"=>186, "height"=>182}} {"label"=>"orange", "type"=>"rectangle", "coordinates"=> {"x"=>378, "y"=>229, "width"=>186, "height"=>180}} {"label"=>"grapefruit", "type"=>"rectangle", "coordinates"=> {"x"=>377, "y"=>228, "width"=>187, "height"=>188}}

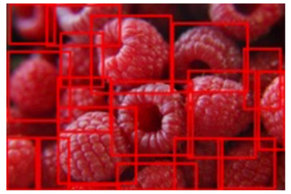
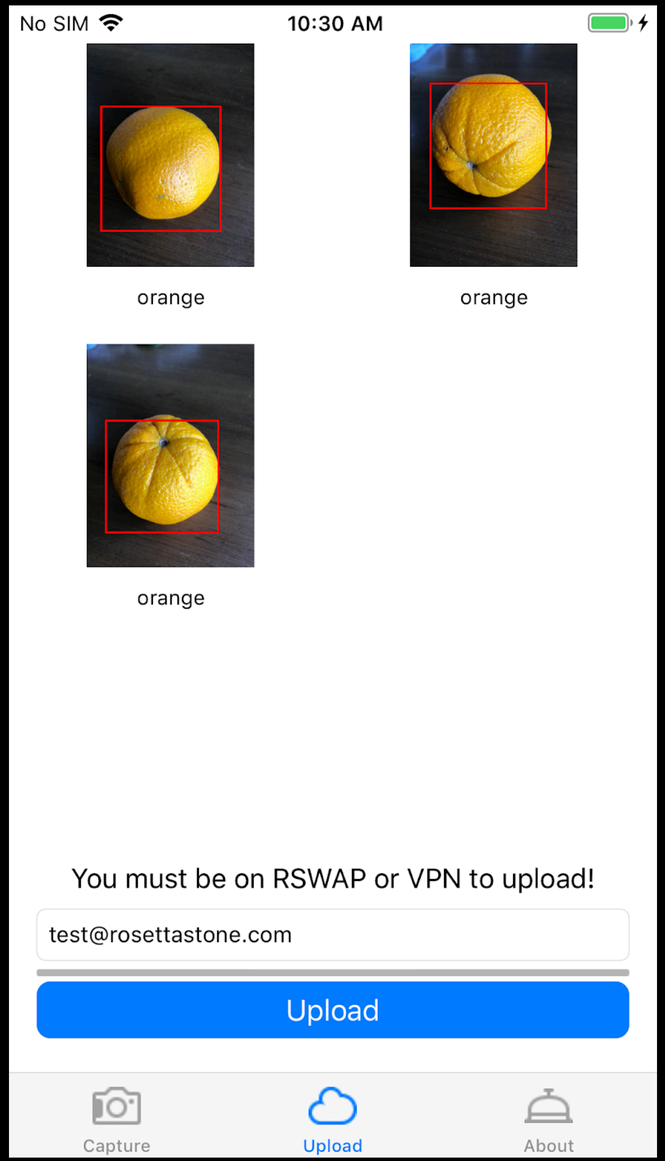
	○ Approved ○ Rejected	"coordinates"=>{"x"=>167, "y"=>217, "width"=>112, "height"=>116}} {"label"=>"strawberries", "type"=>"rectangle", "coordinates"=>{"x"=>195, "y"=>304, "width"=>167, "height"=>164}} {"label"=>"strawberries", "type"=>"rectangle", "coordinates"=>{"x"=>162, "y"=>121, "width"=>91, "height"=>115}} {"label"=>"strawberries", "type"=>"rectangle", "coordinates"=>{"x"=>187, "y"=>395, "width"=>109, "height"=>10}} {"label"=>"strawberries", "type"=>"rectangle", "coordinates"=>{"x"=>268, "y"=>109, "width"=>172, "height"=>166}} {"label"=>"strawberries", "type"=>"rectangle", "coordinates"=>{"x"=>313, "y"=>246, "width"=>173, "height"=>158}} {"label"=>"strawberries", "type"=>"rectangle", "coordinates"=>{"x"=>323, "y"=>371, "width"=>169, "height"=>57}}

IMAGE COLLECTION

On-device capture: Fits use case

Creative Commons search, CC0

Annotations (RectLabel, LabelImg)


No SIM
10:30 AM
orange
orange
orange
You must be on RSWAP or VPN to upload!
test@rosettastone.com
Upload
Capture
Upload
About

PILES O' PICTURES

Folders: image and XML annotation

Annotation has file path

Can't see annotation and image togeth

Random file names

Ad hoc association of image and annotation

LET'S WRITE A SERVICE!

yay

CUSTOM RAILS SERVICE

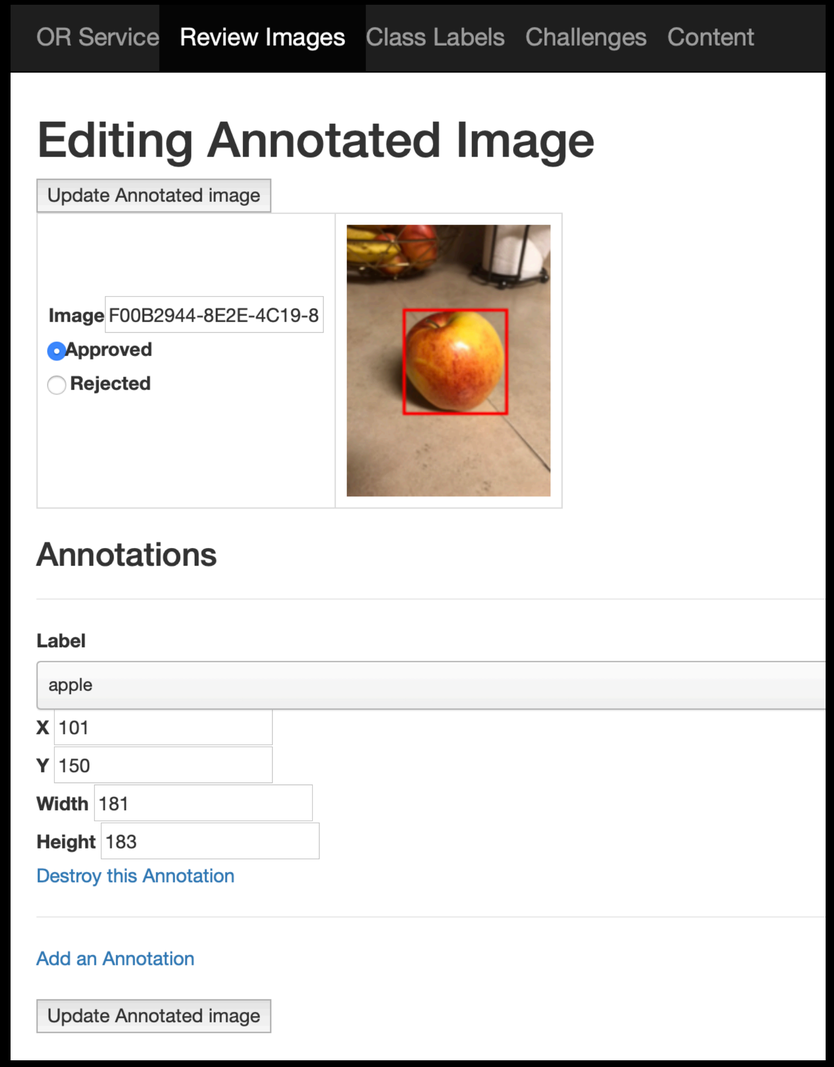
Server upload, API

Store reference

Annotation

Labels

Challenges



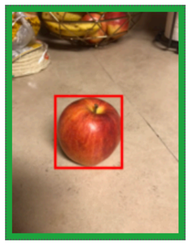
OR Service Review Images Class Labels Challenges Content signed in as dmowry sign out

apple

All Approved Rejected

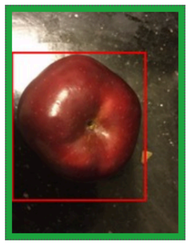
← Previous 1 2 3 4 5 6 7 8 9 … 27 28 Next →


Approved:
Destroy


Approved:
Destroy


Approved:
Destroy


Approved:
Destroy


Approved:
Destroy

OR Service | Review Images | Class Labels | Challenges | Content

signed in as dmowry sign out

Editing Challenge

Name In the Classroom

Description classroom

MImodel name classroom

Class Labels

scissors | tape | laptop | headphones | notebook | earbuds | tablet | book | calculator | crayon | eraser | flash drive | glue stick | marker | paper clip | pen | pencil | pencil sharpener | sticky notes | stapler

Negative Class Labels

batteries | can opener | tweezers | nail clippers | camera | keyboard | wires | text | not-classroom

Update Challenge

BUILDING A MODEL

Script in Python uses Turi Create 🎉

Download challenge from API: images in common directory, annotation CSV

Generate SFrame

eGPU

Testing

Explicit Classes

“Negative” Classes

Quantization

QUANTIZATION

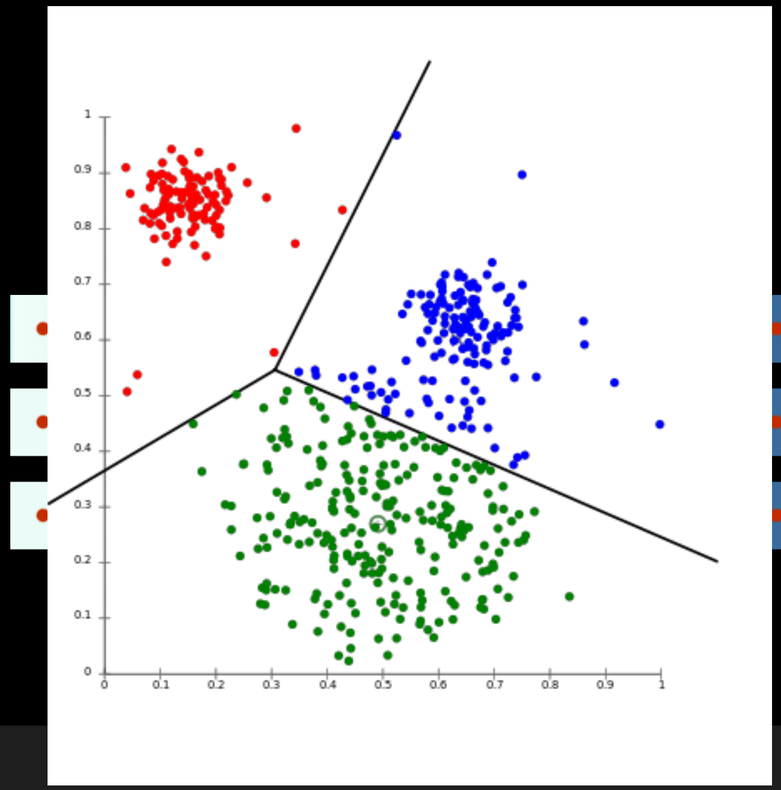
What's new in Core ML (WWDC 2018)

32-bit Float weights

Reduce weights by chunking into integers

Linear or look up table

Core ML Tools




```
class Quantizator:
    functions = ["linear", "linear_lut", "kmeans"]
    bits = [16,8]

    def quantize_model(self, model_directory, output_directory, model_name):
        model_file_name = "{0}.mlmodel".format(model_name)
        model = coremltools.models.MLModel(os.path.join(model_directory, model_file_name))
        for function in self.functions:
            for bit in self.bits:
                str_bit = str(bit)
                print("Processing {0} for {1} on {2}.".format(function, str_bit, model_name))
                lin_quant_model = quantize_weights(model, bit, function)
                lin_quant_model.short_description = "{0} bit per quantized weight, using {1}.".format(str_bit, function)
                lin_quant_model.save(os.path.join(output_directory, "{0}_{1}_{2}.mlmodel".format(model_name, function, str_bit)))
```

IMAGES AND ANNOTATIONS

Docs say at least 30: more work better the more objects in a challenge

Balance numbers across challenge

Image augmentation?

MODEL GENERATION RUNS

Challenge

Explicit test images

Can set number of iterations

Save output to analyze data

Average Precision

$$\text{Precision} = \frac{\text{True Positive}}{\text{True Positive + False Positive}}$$

```
Setting 'batch_size' to 32
Using GPU to create model (AMD Radeon Pro 580)
Setting 'max_iterations' to 7000
+--------------+--------------+--------------+
| Iteration    | Loss         | Elapsed Time |
+--------------+--------------+--------------+
| 1            | 11.565       | 7.4          |
| 27           | 11.038       | 17.5         |
| 50           | 9.721        | 27.9         |
| 72           | 8.707        | 38.2         |
| 93           | 8.104        | 48.2         |
| 115          | 7.251        | 58.3         |
| 135          | 6.915        | 68.6         |

...

| 6747         | 1.511        | 3331.5       |
| 6768         | 1.647        | 3342.0       |
| 6789         | 1.618        | 3352.1       |
| 6809         | 1.553        | 3362.2       |
| 6830         | 1.512        | 3372.6       |
| 6851         | 1.564        | 3382.7       |
| 6871         | 1.510        | 3392.9       |
| 6891         | 1.558        | 3403.0       |
| 6912         | 1.606        | 3413.7       |
| 6934         | 1.567        | 3424.0       |
| 6955         | 1.538        | 3434.3       |
| 6975         | 1.571        | 3444.4       |
| 6996         | 1.561        | 3454.5       |
+--------------+--------------+--------------+
trained
Predicting   1/257
Predicting  48/257
Predicting  95/257
Predicting 137/257
Predicting 183/257
Predicting 225/257
Predicting 257/257
{'average_precision_50': {'kiwi': 1.0, 'blueberries': 0.8004201680672
0.5137879634612466, 'blackberries': 0.7173166384070258, 'avocado': 0.
'grapes': 0.18925190618662266, 'raspberries': 0.6660938813453046, 'ma
'pineapple': 0.9166666666666666, 'orange': 0.5936529367525131, 'straw
'mean_average_precision_50': 0.6709707434675414}
evaluated
```

360|iDev

MODEL EVALUATION

Average Precision

Quantization - Reza Shirazian: https://github.com/kingreza/quantization

model	size	blackberries	blueberries	cherries
fruit.mlmodel	64522921	0.765306122	0.76744186	0.795454545
fruit_kmeans_16.mlmodel	32264761	0.765306122	0.76744186	0.795454545
fruit_kmeans_8.mlmodel	16148838	0.765306122	0.76744186	0.795454545
fruit_linear_16.mlmodel	32264761	0.765306122	0.76744186	0.795454545
fruit_linear_8.mlmodel	16153748	0.775510204	0.76744186	0.795454545
fruit_linear_lut_16.mlmodel	32264765	0.765306122	0.76744186	0.795454545
fruit_linear_lut_8.mlmodel	16148842	0.744897959	0.790697674	0.795454545

Test Harness - Walking around

TAKE IT FOR A SPIN

Test harness hosts the component used in app

Can use on images from web search (unseen test cases)

Must fit use case (real settings, not white background stock shots)

Walking around

Helps uncover false positives for negative classes

Once satisfied, give to QA for objective evaluation

If not, tweak

MODEL ITERATION

Need to go back to first principles:

- Image fit to use case
- Bounding boxes
- Balance
- Quality
- Negative classes

Regenerate and evaluate

DEPLOYMENT

Versioned

Same process as existing asset deployment (CDN)

Manifest published

App detects manifest changes

App offers learner chance to update

LESSONS

IMAGES AND ANNOTATIONS

Manage collection and annotation

Fit to use case

Tight bounding boxes

Complete annotation of all objects in image

Balanced numbers

“Negative” classes

Centralize storage and management

BUILDING

Control over generation runs

Explicit test cases

eGPU

Scripting

Evaluation & Testing

Quantization

Iteration

USING THE MODELS

Decouple content and model

Use confidence values from Vision

Manage user expectations with messaging

Fit application use case

GIANT SHOULDERS

FUTURE DIRECTIONS

On-device improvements with Core ML 3

Sharing improvements back to server

Annotate on server

Swift for TensorFlow: Share with Android

CreateML App (WWDC 2019)

RESOURCES

Apple ML blog https://machinelearning.apple.com/

Vision sample https://developer.apple.com/documentation/vision/classifying_images_with_vision_and_core_ml

Turi Create GitHub site https://github.com/apple/turicreate

Turi Object Detection https://github.com/apple/turicreate/blob/master/userguide/object_detection/README.md

2018 WWDC video https://developer.apple.com/videos/play/wwdc2018/712/

2019 WWDC video https://developer.apple.com/videos/play/wwdc2019/420/

2019 WWDC Create ML app video https://developer.apple.com/videos/play/wwdc2019/424

THANK YOU

QUESTIONS?

@donmowry