Machine Learning for the Real World: What is Acceptable Accuracy, and How Can You Achieve It?

Tim Hartley
Director Product & Marketing
ML on the Edge or in the Cloud?

At or near the Edge

• Reduced round-trip latency
• Better privacy
• Reduced bandwidth costs
• Reduced cloud compute costs

In the Cloud

• Plenty of compute power
• Scalable
• Easy to deploy
Creating Distributed ML Applications

<table>
<thead>
<tr>
<th>In an ideal world</th>
<th>In reality</th>
</tr>
</thead>
<tbody>
<tr>
<td>• We could run ML where it makes best sense to run it</td>
<td>• Edge compute power varies hugely</td>
</tr>
<tr>
<td>• Moving workloads from edge to cloud and back at will</td>
<td>• ML workloads will not run across all platforms</td>
</tr>
<tr>
<td>• And from edge platform to edge platform with no stickiness</td>
<td>• Varying software APIs and libraries</td>
</tr>
<tr>
<td>• With a wide range of edge devices created with ML acceleration capabilities</td>
<td>• ML &amp; CV functionality often proprietary</td>
</tr>
<tr>
<td></td>
<td>• Performance-portable applications difficult to write</td>
</tr>
</tbody>
</table>
In an ideal world

• We could move and store media easily between edge and cloud
  • Applying ML compute on media freely captured and distributed
  • With guarantees of security for personal information

In reality

• GDPR and other privacy concerns
  • What can be captured, stored and distributed is under ever-increasing scrutiny
  • Anything that can personally identify you is coming under strict control
  • Difficult to prove system-wide security, particularly with the cloud
Compute Hierarchy: Edge to Gateway to Cloud

**Edge**
- Video streams
- Smart gateways process video streams from one or more cameras
- Run ML algorithms to determine location of people and other objects
- IP cameras and other sensors

**Gateway**
- Smart gateways
- IP cameras and other sensors
- Raw detection metadata stream

**Cloud**
- Analytics on processed metadata
- Provide live and historical analysis
- Deliver notifications
- Aggregate metadata from smart gateways
- Determine and deliver notifications

**Analytics**
- 3rd party apps
- Dashboards

**Normalised and categorised metadata**
- Raw detection metadata stream
- Cloud or on-prem server

© 2020 Arm Ltd
A Real World Case Study

People / object detection for determining space occupancy in real time
A Real World Case Study: Real-Time Space Occupancy

Overview

• People and object detection (not face recognition)
• To determine real-time space occupancy

Requirements opposite to determine
• Machine learning detection model
• And a suitable hardware platform

Key requirements

• Off-the-shelf IP cameras
• No personally identifiable information to be sent to the cloud
• Independent of viewing angle
• ~80ft detections, HD @ 50mm lens
• Low false negatives and false positives
• 1 frame every 15s minimum
Step 1
Find a Suitable Model
Datasets and Models

Dataset candidates

Many to choose from:

- COCO, Kitti, Open Images, etc.

Selected COCO for now

- Good range of classes and models
- And a high image count in the classes we cared most about

Model candidates

A good selection of COCO-trained models

- ranked using a “COCO mAP[^1]” metric
- and by execution time

These are excellent starting points

- Downloadable and usable quickly
Comparison of COCO-Trained Models

<table>
<thead>
<tr>
<th>Model Name</th>
<th>Detection Quality (COCO mAP[^1])</th>
<th>Execution Time (ms on desktop GPU)</th>
</tr>
</thead>
<tbody>
<tr>
<td>SSD_MOBILENET_V2_COCO</td>
<td>22</td>
<td></td>
</tr>
<tr>
<td>FASTER_RCNN_RESNET101_COCO</td>
<td>32</td>
<td></td>
</tr>
<tr>
<td>SSD_MOBILENET_V2_QUANTIZED_COCO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SSD_MOBILENET_V2_COCO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SSD_RESNET_50_FPN_COCO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SSD_MOBILENET_V1_FPN_COCO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SSD_MOBILENET_V1_PPN_COCO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SSD_MOBILENET_V1_0.75_DEPTH_QUANTIZED_COCO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SSD_MOBILENET_V1_QUANTIZED_COCO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SSD_MOBILENET_V1_0.75_DEPTH_COCO</td>
<td></td>
<td></td>
</tr>
<tr>
<td>SSD_MOBILENET_V1_COCO</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

[^1]: COCO mAP is a measure of detection quality.

FASTER_RCNN_RESNET101_COCO Detection quality = 32

SSD_MOBILENET_V2_COCO Detection quality = 22

Smaller is better

Bigger is better
Comparison of Detection Performance

SSD_MOBILENET_V2_COCO

- 53%
- 79%

FASTER_RCNN_RESNET101_COCO

- 77%
- 69%
- 95%
- 98%
- 98%
Step 2
Tuning the Model
Tuning the Model Using Transfer Learning

1. Gather and label new set of sample images (~250)
2. Crop & create rotated, brightness, quality variations
3. Feed through training pipeline using existing network
4. Repeat for each image

Pre-trained model based on FasterRCNN & ResNet101
Transfer Learning: Reducing the Corner Cases

Before

After
### Impact of Transfer Learning

<table>
<thead>
<tr>
<th>Detection quality (COCO mAP[^1])</th>
<th>Execution time (ms on desktop GPU)</th>
</tr>
</thead>
<tbody>
<tr>
<td>15</td>
<td>0 - 200</td>
</tr>
<tr>
<td>20</td>
<td>200 - 400</td>
</tr>
<tr>
<td>25</td>
<td>400 - 600</td>
</tr>
<tr>
<td>30</td>
<td>600 - 800</td>
</tr>
<tr>
<td>35</td>
<td>800 - 1000</td>
</tr>
<tr>
<td>40</td>
<td>1000 - 1200</td>
</tr>
<tr>
<td>45</td>
<td>1200 - 1400</td>
</tr>
<tr>
<td>50</td>
<td>1400 - 1600</td>
</tr>
<tr>
<td>55</td>
<td>1600 - 1800</td>
</tr>
<tr>
<td>60</td>
<td>1800 - 2000</td>
</tr>
</tbody>
</table>

**Bigger is better**

**Smaller is better**

**Detection quality = 42**
(post-transfer learning)

**Detection quality = 32**
(pre-transfer learning)
Product Implications of Tuned Networks

The retraining process

• Additional training takes ~2 hours
  • Using around 250 images
  • On a portable training rig
• Pipeline allows for tuning to support...
  • corner cases
  • lower quality cameras
  • difficult lighting conditions

Product considerations

• Significantly reduce false +ve & – ve
• Allows for tuning to specific venues
  • e.g. customising for specific desk and chair types
  • Training pipeline relatively painless compared to retraining the entire model
• Implies more skilled installation labor
Step 3
Applying to the Real World
Applying to the Real-World

Working with cameras

“Everything matters…”
• Model choice key to quality
• Quality of lens and sensor, resolution
• Sharpness, noise, compression
• Lighting & lighting variability

Real-world product implications

Installation / configuration
• Complex and skilled, unless...
  • Can we mitigate low quality devices by training with degraded images?
  • Use AI to automate camera setup?
• Camera cost vs features
  • e.g. HDR expensive but can help with variable lighting
## Summary

### Choice of model fundamental to successful real-world vision-based use cases
- As is everything else 😊
- Model tuning can provide significant benefits

### Useful models often need significant compute
- Today’s off-the-shelf cameras unlikely to have sufficient capability
- Aggregating multiple devices into gateways can work well, and this creates real opportunities today

### Compute requirement will grow, enabling new use cases
- More sophisticated models as ML continues to evolve
- Edge-based complex action recognition is particularly challenging
Useful Resources

Useful Links

COCO dataset

http://cocodataset.org/

COCO-trained models

https://github.com/tensorflow/models/blob/master/research/object_detection/g3doc/tf2_detection_zoo.md

Arm Insight Platform

https://www.arm.com/products/arm-insight-platform

2020 Embedded Vision Summit

Machine Learning for the Real World: What is Acceptable Accuracy, and How Can You Achieve It?

Tim Hartley
Trademark and copyright statement
The trademarks featured in this presentation are registered and/or unregistered trademarks of Arm (or its subsidiaries) in the EU and/or elsewhere. All rights reserved. All other marks featured may be trademarks of their respective owners.

Copyright © 2020

Thank You!