tinyML® EMEA

Enabling Ultra-low Power Machine Learning at the Edge

October 10-12, 2022
Limassol, Cyprus

www.tinyML.org
FOMO (Faster Object, More Objects)

Louis Moreau, Senior DevRel Engineer
@luisomoreau
The developer-first edge ML platform

- Any data, any device
- No royalty, no impact on BOM
- Your IP, stays your IP
- Total explainability, no black boxes
Go to market faster with confidence
Deploy to any edge device with ease

- The largest silicon ecosystem
- Award-winning compiler
- Access to device source code
- Full firmware integration for a number of devices
### Any sensor, any data, any use case

<table>
<thead>
<tr>
<th>Sensor</th>
<th>Audio</th>
<th>Image</th>
<th>Video</th>
</tr>
</thead>
<tbody>
<tr>
<td>Memory</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Ultra low power</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Low-end MCU</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>High-end MCU</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>NPU</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>MPU</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>GPU</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
</tbody>
</table>

- Anomaly detection: 10kB
- Sensor fusion classification: 18kB
- Audio classification: 50kB
- Image classification: 256kB
- Object detection complex voice processing: 1MB
- Video classification: 1GB+
Image classification

The question the model is trying to answer is:

“Is there a face or not in the image?”
Object Detection using Bounding Boxes

The question the model is trying to answer is:

“Are there faces in the image, where and what size are they?”
Object Detection using Centroids

*The question the model is trying to answer is:*

“Are there faces in the image, where are they?”
Object Detection with

MobileNetV2 SSD
FPN-Lite 320x320
pre-trained model
MobileNetV2 SSD FPN-Lite 320x320 pre-trained model

In MobileNetV2 SSD FPN-Lite, we have:
- a base network (MobileNetV2),
- a detection network (Single Shot Detector or SSD)
- a feature extractor (FPN-Lite)
- transfer learning approach (pre-trained model)

Base Network - MobileNetV2:

The base network provides **high-level features** for classification or detection. If you use a fully connected layer and a softmax layer at the end of these networks, you have a classification.
MobileNetV2 SSD FPN-Lite 320x320 pre-trained model

In MobileNetV2 SSD FPN-Lite, we have:
- a base network (MobileNetV2),
- a detection network (Single Shot Detector or SSD)
- a feature extractor (FPN-Lite)
- transfer learning approach (pre-trained model)

Detection Network - SSD multi-box:

The most common detection networks are SSD (Single Shot Detection) and RPN (Regional Proposal Network). When using SSD, we only need to take one single shot to detect multiple objects within the image.
MobileNetV2 SSD FPN-Lite 320x320 pre-trained model

In MobileNetV2 SSD FPN-Lite, we have:
- a base network (MobileNetV2),
- a detection network (Single Shot Detector or SSD)
- a feature extractor (FPN-Lite)
- transfer learning approach (pre-trained model)

Feature Extractor - FPN-Lite:

Detecting objects in different scales is challenging in particular for small objects. Feature Pyramid Network (FPN) is a feature extractor designed with feature pyramid concept to improve accuracy and speed.
MobileNetV2 SSD FPN-Lite 320x320 pre-trained model

In MobileNetV2 SSD FPN-Lite, we have:
- a base network (MobileNetV2),
- a detection network (Single Shot Detector or SSD)
- a feature extractor (FPN-Lite)
- transfer learning approach (pre-trained model)

Transfer Learning

In transfer learning, we take a pre-trained model performing classification on a dataset and then apply this same model to another set of classification task by just optimising the hyperparameters a little bit. Transfer Learning has two benefits:

- It requires less time to train the model as it is already trained on a different task
- It can be used for tasks which have smaller dataset as the model is already trained on a larger dataset and the weights are transferred to the new task
Problem

Traditional object detection models are poorly suited to MCUs
Object Detection with

FOMO
(Faster Objects, More Objects)
Introducing FOMO: Faster objects, more objects

- Object detection on MCUs
- Based on MobileNet architecture
- Ultra fast: 60 fps on RPi class, 30 fps on Cortex M7 (Arduino Nicla Vision), 10 fps on Cortex M4F
- Better at detecting smaller and more numerous objects
- Capable of segmentation and counting objects
MobileNet V2

240x240
Faster Objects, More Objects (FOMO)

- Height and width are each divided by 8 (default)
- 2D convolution with 1x1 kernel is used

240x240

- Convolutional layer(s) → Bottleneck residual block 1 → Bottleneck residual block 2 → Bottleneck residual block 3

- Fully Connected layer → Fully Connected layer → Fully Connected layer → Softmax

- Class predictions per cell

- Looks like segmentation of feature maps

- Weights in residual blocks are pre-trained from ImageNet

- 30x30 grid (or feature maps), each cell having 8x8 pixels
Faster Objects, More Objects (FOMO)

Each cell is given scores:
- \( P(\text{background}) \)
- \( P(\text{ball}) \)
- \( P(\text{dog}) \)
- \( P(\text{toy}) \)
Faster Objects, More Objects (FOMO)

Example: screws
- Grayscale
- Image: 96x96
- Feature maps: 12x12
FOMO Ground Truth

Example: screws
- Grayscale
- Image: 96x96
- Feature maps: 12x12

User draws bounding boxes, tool picks cell with centroid of bounding box
FOMO Ground Truth

Example: screws
- Grayscale
- Image: 96x96
- Feature maps: 12x12

User draws bounding boxes, tool picks cell with centroid of bounding box

Those cells are now representatives of that class
Example: screws
- Grayscale
- Image: 96x96
- Feature maps: 12x12
Example: screws
- Grayscale
- Image: 96x96
- Feature maps: 12x12

Neighboring cells with same class are removed (leaving highest scores)
<table>
<thead>
<tr>
<th>Specifications</th>
<th>MobileNetV2 SSD FPN</th>
<th>FOMO</th>
</tr>
</thead>
<tbody>
<tr>
<td>Labelling method</td>
<td>Bounding boxes</td>
<td>Bounding Boxes</td>
</tr>
<tr>
<td>Input size</td>
<td>320×320</td>
<td>Square (any size)</td>
</tr>
<tr>
<td>Image format</td>
<td>RGB</td>
<td>Greyscale + RGB</td>
</tr>
<tr>
<td>Output</td>
<td>Bounding boxes</td>
<td>Centroids</td>
</tr>
<tr>
<td>MCU</td>
<td>✗</td>
<td>✓</td>
</tr>
<tr>
<td>CPU/GPU</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Limitations</td>
<td>- Works best with big objects</td>
<td>- Works best when objects have similar sizes &amp; shapes</td>
</tr>
<tr>
<td></td>
<td>- Models use high compute resources (in the edge computing world)</td>
<td>- The size of the objects are not available</td>
</tr>
<tr>
<td></td>
<td>- Image size is fixed</td>
<td>- Objects should not be too close to each other</td>
</tr>
</tbody>
</table>

Copyright © Edge Impulse Inc.
Object detection on Raspberry Pi 4

- MobileNet SSD
- FOMO

Frames per second
To go further

Create your free account on edgeimpulse.com

Build your first model in 5 minutes: studio.edgeimpulse.com/evaluate

FOMO: edgeimpulse.com/fomo

Documentation: docs.edgeimpulse.com
Thank you!

@luisomoreau
Copyright Notice

This presentation in this publication was presented at the tinyML® EMEA Innovation Forum 2022. The content reflects the opinion of the author(s) and their respective companies. The inclusion of presentations in this publication does not constitute an endorsement by tinyML Foundation or the sponsors.

There is no copyright protection claimed by this publication. However, each presentation is the work of the authors and their respective companies and may contain copyrighted material. As such, it is strongly encouraged that any use reflect proper acknowledgement to the appropriate source. Any questions regarding the use of any materials presented should be directed to the author(s) or their companies.

tinyML is a registered trademark of the tinyML Foundation.