Bringing face-swapping to the very edge with Phinets

Alberto Ancilotto, Francesco Paissan, Elisabetta Farella
Head of E3DA research unit @FBK
E-mail: efarella@fbk.eu
Edge Anonymization: A Privacy-By-Design Approach

- Visual Inspection
- Personal Data Leaves Device
- Anonymization Preserving Pose/Expression
- Edge Processing
- No Personal Data Leaves Device
Face Swapping & Video Anonymization

- **Swapping faces** between images or in a video, while maintaining the rest of the body and environment context

- More complex than simple classification/detection, requires image generation

- Usually performed on workstations using powerful GPUs
Hardware constrained scaling

- Architectural space exploration;
- Picks best performing network within hardware limitations;

However...

- This approach is **time consuming**, due to the amount of networks to train;
- **Cannot be ported** to different hardware architectures (no scaling strategy);

NN Models from literature trained on Google speech commands dataset.
Hardware aware scaling

- Enables **one-shot design of neural architectures** given the available resources on the platform;

- Designs the network to benefit from **all available resources** on the given platform (the bigger, the better);

- Can easily scale on **different hardware platforms**;

---

F. Paissan, A. Ancilotto, E. Farella "PhiNets: a scalable backbone for low-power AI at the edge", ACM Transactions on Embedded Computing Systems 2022
PhiNets
Convolutional Block

Three parameters:

- $t_0$ sets RAM usage
- $\alpha$ sets MACC requirements
- $\beta$ sets FLASH usage
Generative Networks on microcontrollers

- **GANs** require **orders of magnitude more computation** than image classification networks, making it prohibitive for edge applications.

- Eg:  
  - **GauGan:** 93 M parameters, 281 G MACs  
  - **CycleGAN:** 11 M parameters, 57 G MACs

- SOTA compression methods allow for 10x - 20x reduction in MACs [1], not sufficient.

- **Deconvolutional blocks are traditionally inefficient**, and SW/HW co-design is needed to achieve efficiency when executing GANs on edge devices [2].

---

PhiNets Upsampling Block

With respect to transposed convolutions:
- More than 85% parameter reduction
- More than 95% MACC reduction
The decoding section of GANs usually requires >90% the operations of the whole network. Optimizing upsampling blocks brings GANs to smaller edge devices.

We propose an efficient upsampling block based on bilinear filtering that makes Generative Networks able to fit on MCUs.
GANs running on Kendryte K210
RISC V dual core
400 Mhz
<300mW power consumption
Face Swapping on microcontrollers

- Traditionally only classical CV could be used, leading to subpar results [1]

- Face Swapping using **GAN** generated faces leads to high fidelity reconstruction, but requires **very large generative networks**:
  - **SimSwap**[3]: 107 M parameters, 56 G MACs
  - **FaceShifter**[4]: 421 M parameters, 98 G MACs

- Compression techniques have been applied to SOTA face swapping networks leading to architectures under 75 M parameters and 10 G MACs [2] - **still 2 orders of magnitude to go!**

Proposed Method

- Based on FSGAN, simpler but a decoder can only generate one target identity
- Replaces FSGAN encoder and decoder networks with PhiNets
- Modifies the approach to be fully convolutional for better efficiency on microcontroller runtimes
Proposed Method

- Train two autoencoder networks, one for the target face, one for a general face (using VGGface dataset)
- Train a discriminator network to identify real and fake faces
- Swap the decoders and use the discriminator as a loss to train the face swapping network
Final Architecture

Obtained using **Hardware Aware Scaling** starting from the specs of the K210 MCU.

- Flash: 1 MB
- RAM: 392 KB
- MACC: 63 M

Flash: 1 MB  \( t_0: 4.0 \)
RAM: 392 KB  \( \alpha: 1.0 \)
MACC: 63 M  \( \beta: 0.66 \)

Encoder: PhiNet
\( \{\alpha: 1.0, t_0: 4.0, \beta: 0.66\} \)

Decoder: PhiNet GAN
\( \{\alpha: 1.0, t_0: 4.0, \beta: 0.66\} \)
The challenge with GAN quantization

- Quantizing GANs is difficult, as small errors can propagate and lead to instability [1]

- We employ partial quantization, leaving the smallest layers in floating point

- Pooling layers downsample the image with less numerical instability than strided convolutions

- This leads to a <3% increase in latency but a huge increase in generated image quality. This is fundamental for subsequent analysis of the frames.

### Performance

<table>
<thead>
<tr>
<th></th>
<th>Simswap:</th>
<th>FSGAN:</th>
<th>Ours:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>127 M parameters</td>
<td>224 M parameters</td>
<td>0.98 M parameters</td>
</tr>
<tr>
<td></td>
<td>55.7 G MACC</td>
<td>24.4 G MACC</td>
<td>0.064 G MACC</td>
</tr>
<tr>
<td></td>
<td>142 FID</td>
<td>148 FID</td>
<td>152 FID</td>
</tr>
</tbody>
</table>

99.2% Reduction in parameters
98.4% Reduction in MACC
~6% Performance penalty
Phinets for face swapping

1. Face Detection
   Phinets + Yolo

2. Landmark Detection
   Phinets + PFLD

3. Face Alignment
   Phinets + PFLD

4. Face Generation
   Phinets GAN
### Results

<table>
<thead>
<tr>
<th></th>
<th>Face detection:</th>
<th>Landmark detection:</th>
<th>Face generation:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameters</td>
<td>27 K</td>
<td>64 K</td>
<td>984 K</td>
</tr>
<tr>
<td>MACC</td>
<td>6 M</td>
<td>11 M</td>
<td>62 M</td>
</tr>
<tr>
<td>RAM</td>
<td>122 KB</td>
<td>188 KB</td>
<td>392 KB</td>
</tr>
<tr>
<td>Energy</td>
<td>4.1 mJ/frame</td>
<td>7.3 mJ/frame</td>
<td>34.3 mJ/frame</td>
</tr>
</tbody>
</table>

**Overall**

46 mJ / frame on K210

6 fps @ 280 mW
Do you want to swap your face with the one of Nicolas Cage in real-time on an IoT device? Visit the Poster and Demo area and ask to Alberto!
Thanks for your attention!

For more information find us at:

efarella@fbk.eu
aancilotto@fbk.eu
fpaissan@fbk.eu

For more details:
Our article is under submission!
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.