TinyM2Net: A Flexible System Algorithm Co-designed Multimodal Learning Framework for Tiny Devices
Hasib-Al Rashid1, Pretom Roy Ovi1, Carl Busart2, Arya Gangopadhyay1 and Tinoosh Mohsenin1
1. University of Maryland, Baltimore County
2. U.S. Army Research Laboratory

Motivation
- To integrate AI in our day-to-day life, it is being implemented on resource constrained mobile and edge platforms.
- With the exponential growth of resource constrained micro-controller (MCU) and micro-processor (MPU) powered devices, a new generation of neural network has emerged, one that is smaller in size and more concerned with model efficiency than model accuracy.
- These low-cost, low-energy MCUs and MPUs open up a whole new world of tiny machine learning (tinyML) possibilities.

TinyM2Net Framework

Why TinyM2Net?
- TinyM2Net is a novel flexible system-algorithm co-designed multimodal learning framework for resource constrained devices.
- TinyM2Net that can take multimodal inputs (images and audio) and be re-configure for application specific requirements.
- TinyM2Net allows the system and algorithms to quickly integrate new sensors that are customized to various types of scenarios.

Contribution Towards tinyML Implementation
- Performed network architecture optimization with depthwise separable CNN (DS-CNN) which reduces both the memory requirements and required computations.
- Performed model compression with mixed-precision quantization with the purpose of decreasing memory size for resource constrained hardware implementation while maintaining accuracy.
- Evaluated proposed TinyM2Net for two different case-studies.
  - Case-study 1 includes Covid-19 detection from multimodal images and audios.
  - Case-study 2: Battlefield Object Detection from Multimodal Images and Audios

Experimental Results
- We trained our model with categorical cross-entropy loss and Adam optimizer.
- Table: Summary of the TinyM2Net Framework Evaluation Results

Table: Detailed network architecture for Case-Study 1

<table>
<thead>
<tr>
<th>Model</th>
<th>Channels</th>
<th>Dilations</th>
<th>Output Size</th>
<th>Neurons</th>
</tr>
</thead>
<tbody>
<tr>
<td>TinyM2Net</td>
<td>1</td>
<td>1</td>
<td>64</td>
<td>64</td>
</tr>
<tr>
<td>DS-CNN</td>
<td>1</td>
<td>1</td>
<td>64</td>
<td>64</td>
</tr>
<tr>
<td>TinyM2Net</td>
<td>1</td>
<td>1</td>
<td>64</td>
<td>64</td>
</tr>
<tr>
<td>DS-CNN</td>
<td>1</td>
<td>1</td>
<td>64</td>
<td>64</td>
</tr>
<tr>
<td>TinyM2Net</td>
<td>1</td>
<td>1</td>
<td>64</td>
<td>64</td>
</tr>
<tr>
<td>DS-CNN</td>
<td>1</td>
<td>1</td>
<td>64</td>
<td>64</td>
</tr>
</tbody>
</table>

Table: Detailed network architecture for Case-Study 2

<table>
<thead>
<tr>
<th>Model</th>
<th>Channels</th>
<th>Dilations</th>
<th>Output Size</th>
<th>Neurons</th>
</tr>
</thead>
<tbody>
<tr>
<td>TinyM2Net</td>
<td>1</td>
<td>1</td>
<td>64</td>
<td>64</td>
</tr>
<tr>
<td>DS-CNN</td>
<td>1</td>
<td>1</td>
<td>64</td>
<td>64</td>
</tr>
<tr>
<td>TinyM2Net</td>
<td>1</td>
<td>1</td>
<td>64</td>
<td>64</td>
</tr>
<tr>
<td>DS-CNN</td>
<td>1</td>
<td>1</td>
<td>64</td>
<td>64</td>
</tr>
<tr>
<td>TinyM2Net</td>
<td>1</td>
<td>1</td>
<td>64</td>
<td>64</td>
</tr>
<tr>
<td>DS-CNN</td>
<td>1</td>
<td>1</td>
<td>64</td>
<td>64</td>
</tr>
</tbody>
</table>

Acknowledgement
We acknowledge the support of the U.S. Army Grant No. W911NF21-2076.

References