How to build an OpenVisionCapsules Compatible Hardware
HW VisionCapsules System Architecture

Non smart vision device

Open Stream Gateway

Video Frames

Edge computing HW w/ AI Vision SW

BrainFrame

OpenVisionCapsules SDK

Smart vision device

Vision Capsules AI

Metadata filled in Video Frames

Video Frames

Load HW DNN Model

AI Metadata in Video Frames

Edge computing HW w/ AI Vision SW

BrainFrame

OpenVisionCapsules SDK

BrainFrame can be download from https://dililiabs.com/docs/downloads/. It is offered by Dilili Labs as an AI Software Vendor.
HW VisionCapsules Adaptor Architecture

Video Frames

HW_detector_person_metadata

HW_detector_adaptor.cap

recognizer_person.cap

classifier_person.cap

tracker_person.cap

Smart Vision Device

BrainFrame

AI metadata

AI results

Enterprise, Business & Industrial Applications

Software Vendor Implementation

HW partner implementation
VisionCapsules Hardware Adaptor Implementation

VisionCapsules Source code can be downloaded from github.com/opencv/open_vision_capsules. HW partner need to implement VisionCapsules HW adaptor. Follow the steps below based on a VisionCapsules sample implementation,

1. HW VisionCapsules receives METADATA, and pack it into proper AI results.
2. Inference should be bypass, as it is done on hardware.
3. If the device stack is not supported by TensorFlow, OpenCV DNN, or OpenVino, then a custom device needs to be created. Check “Add a Custom Device”
BrainFrame out-of-the-box supports TensorFlow, OpenCV DNN, or OpenVino. If a HW partner implementation is not supported by TensorFlow, OpenCV DNN, or OpenVino, then a custom device needs to be added for BrainFrame to load HW accelerated DNN model to a HW device. Follow the steps below to add a Custom device to BrainFrame,

1. **Adding Device Name to the system:** BrainFrame Canned VisionCapsules Runtime Environment uses TensorFlow Stack list_local_devices to find all available devices. HW partner may contact Software Vendor for the BrainFrame to support a Custom Device.

2. **Alternatively, HW partner can hack BrainFrame Implementation,** replace TensorFlow Device with a Custom Device, so BrainFrame will route the request to the list_local_devices of the Custom Device.

3. **User can always ignore 1) & 2) and manually load a DNN model with an external tool. BrainFrame will not be able to control the HW in this case.**
<table>
<thead>
<tr>
<th>Items</th>
<th>Existing Model Formats: Tensorflow - .pb/.meta/.pbtxt, .tflite</th>
<th>Keras - .h5, .keras</th>
<th>Caffe - .caffemodel, .prototxt</th>
<th>Torch - .pt, .pth, t7</th>
<th>ONNX - .onnx</th>
<th>OpenVisionCapsules Format .cap</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Design goal DNN training</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>(Portable format for all others)</td>
</tr>
<tr>
<td>DNN model Computing</td>
<td>AI computing target support: CPU, GPU, or other AI chip</td>
<td>✔</td>
<td></td>
<td></td>
<td>✔</td>
<td>(Self-contained, Executable)</td>
</tr>
<tr>
<td></td>
<td>Batch-deploy DNN models onto multiple computing targets</td>
<td>X</td>
<td></td>
<td></td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Separately batchable inference from parallelizable operations</td>
<td>X</td>
<td></td>
<td></td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>Self-contained DNN model &amp;</td>
<td>DNN weights &amp; model in a single file</td>
<td>X / ✔</td>
<td></td>
<td></td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>processing information</td>
<td>Separate files in most formats</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Executable logic for pre/post processing</td>
<td>X</td>
<td></td>
<td></td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Expose standardized APIs for runtime algorithm configuration</td>
<td>X</td>
<td></td>
<td></td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>DNN Model Interconnect,</td>
<td>Describes the interconnection to other DNN models</td>
<td>X</td>
<td></td>
<td></td>
<td>✔</td>
<td></td>
</tr>
<tr>
<td>Compatibility</td>
<td>Compatible with other formats</td>
<td>X</td>
<td></td>
<td></td>
<td>✔</td>
<td></td>
</tr>
</tbody>
</table>