What are Visual Wake Words?

Visual Wake Word (VWW) is an algorithm for detecting visual wake words on your local microcontroller.

So far, we have focused on gesture recognition and keyword spotting (KWS). With KWS, we empowered our microcontroller with a lightweight auditory modality, allowing the system to monitor the microphone input in real-time to be able to recognize and respond to several keywords of interest to users. Visual wake words (VWW) are the natural extension of this technique to visual data, providing a lightweight visual modality to our microcontroller system.

Image information is inherently more complex than auditory information to analyze using machine learning due to the higher dimensionality and complex spatial correlations. Consequently, image data requires greater computational and memory resources, making it challenging to deploy on resource-constrained microcontrollers. An additional complication, similar to keyword spotting, is the difficulty in procuring sufficient data to train our algorithm. In some scenarios, we may have to generate the dataset ourselves.

In the class, we have looked at the VWW task in some detail, from the implementation in Colab, to discussions of image privacy and copyright. We discussed a specific set of architectures, known as the **MobileNet** architectures, which leverage **depthwise separable convolutions** to minimize required memory and computational resources. We also looked at how **transfer learning** can be used to build upon pre-trained models that were used for similar tasks in the same domain to reduce our need to procure large datasets and train large deep learning models.

The main VWW task we focused on was **person detection**, which focuses on determining the presence or absence of an individual in an image. In this task, we used the VWW dataset to train our person detection model and then performed transfer learning using a relatively small amount of images to adapt our model to perform mask detection. It is important to discuss how to port and deploy such a model to our end-device, which is the main focus of the class “Person Detection (VWW) Application”.

Also note that the deployment procedure will require some pre- and post-processing (e.g., image downscaling, encoding/decoding, and interfacing with the onboard camera) unique to the VWW detection task.

Take in consideration, that is also possible the deployment of keyword spotting and VWW simultaneously using model “**multitenancy**.” More often than not, the typical focus is on running individual models in isolation, independent of other contextual activities. But increasingly there is a growing need to combine information from multiple sensors together to perform an intelligent action as audio-visual recognition, using both the microphone and camera. With model ‘multitenancy’, is possible to simultaneously

operate two models at the same time to mimic sensor fusion (i.e., the ability to tie together inputs from different sensors in close temporal proximity).

(We will not enter in detail about multitenancy in our course, but you can run the example code (multi_tenant.ino) on the Arduino IDE under Files/Examples/Harvard tab).