**For OpenMV Cam:**

* Copy my\_model.tflite and label.txt file to Camera storage.
* Connect Pin 0 of GPIO to Buzzer for creating alerts.
* Code to be used in OpenMV Cam H7Plus
* Use OpenMV Cam IDE to modify the program. Download from https://github.com/openmv/openmv-ide/releases/download/v4.2.5/openmv-ide-windows-4.2.5.exe

import sensor, image, time, tf, pyb

# Initialize the camera and LCD

sensor.reset()                      # Reset the camera

sensor.set\_pixformat(sensor.RGB565) # Set pixel format to RGB565

sensor.set\_framesize(sensor.QVGA)   # Set frame size to QVGA (320x240)

sensor.skip\_frames(time=2000)       # Let the camera adjust

sensor.set\_auto\_gain(False)         # Disable auto gain

sensor.set\_auto\_whitebal(False)     # Disable white balance

clock = time.clock()                # Create a clock object to track FPS

# Initialize the pin (assuming pin P0; adjust as needed)

pin = pyb.Pin('P0', pyb.Pin.OUT\_PP)

pin.low()  # Start with the pin low

# Load the pre-trained TFLite model

net = tf.load(my\_model.tflite', load\_to\_fb=True) # Load model from SD card

# Load the labels (assuming labels.txt is in the same directory)

labels = []

with open('labels.txt') as f:

    for line in f:

        labels.append(line.strip())

while(True):

    clock.tick()                    # Track elapsed milliseconds

    img = sensor.snapshot()         # Capture an image from the camera

    # Run the image through the TensorFlow model

    objs = net.classify(img)

    bird\_detected = False  # Flag to check if a bird is detected

    for obj in objs:

        # Get the label with the highest probability

        label = labels[obj.output().index(max(obj.output()))]

        if label == 'bird':  # Adjust the label name as per your labels.txt

            bird\_detected = True

            # Draw a rectangle around the object and label it

            img.draw\_rectangle(obj.rect(), color=(0, 255, 0))

            img.draw\_string(obj.x(), obj.y() - 10, label, color=(255, 0, 0))

            print("Detected bird at [x=%d, y=%d, w=%d, h=%d]" % (obj.x(), obj.y(), obj.w(), obj.h()))

    # Set the pin high if a bird is detected

    if bird\_detected:

        pin.high()

    else:

        pin.low()

    # Print FPS

    print(clock.fps())