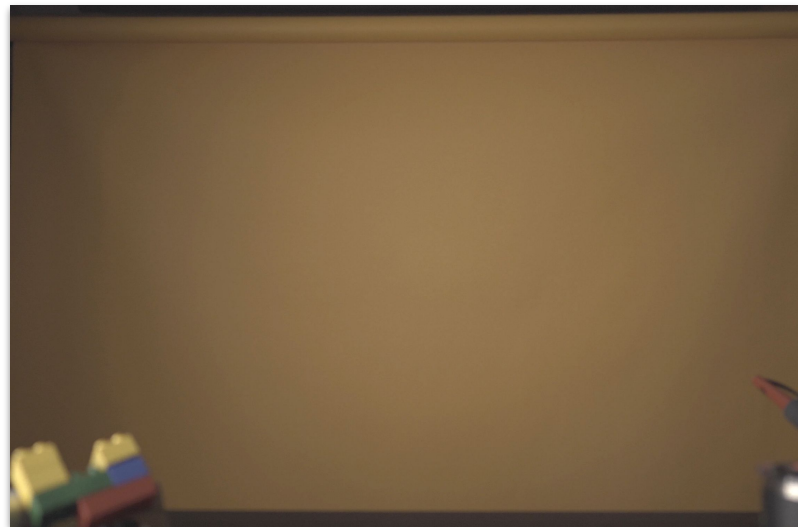


# Person Detection: Application Architecture



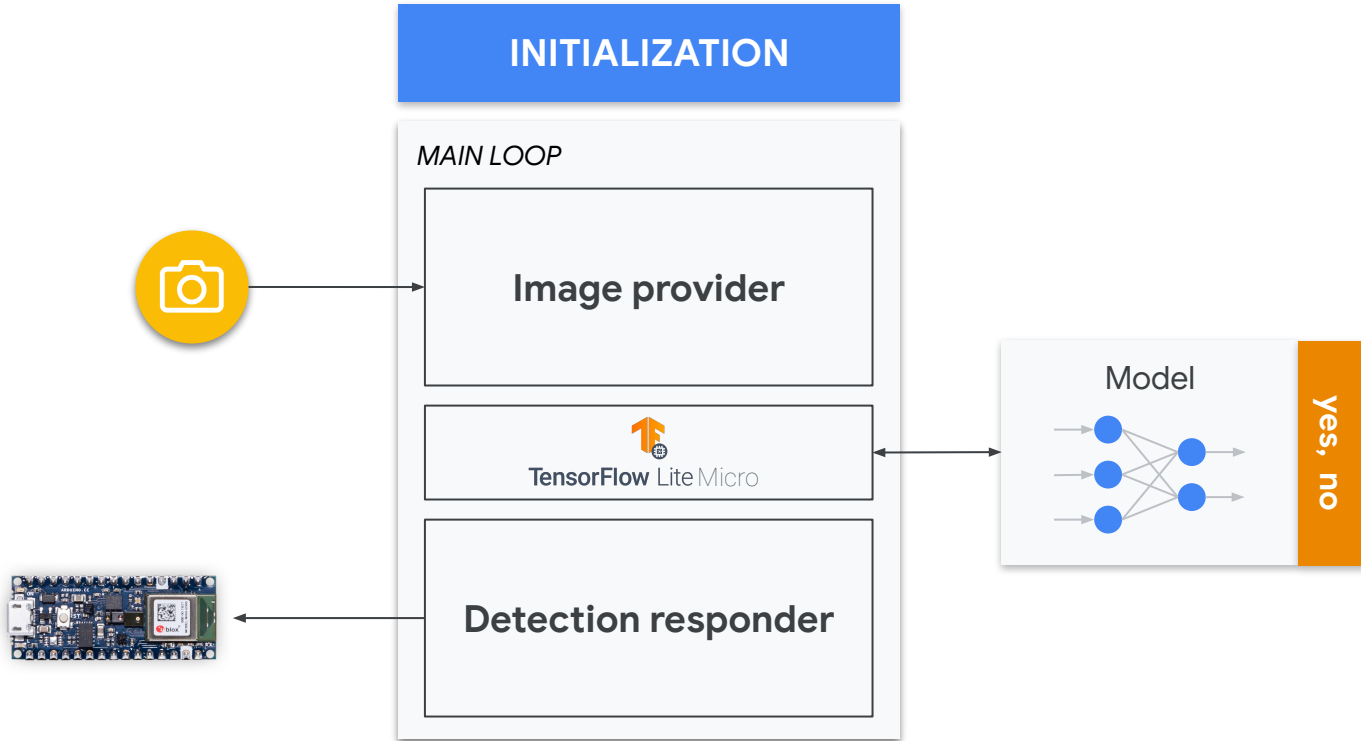


# Person Detection **Components**

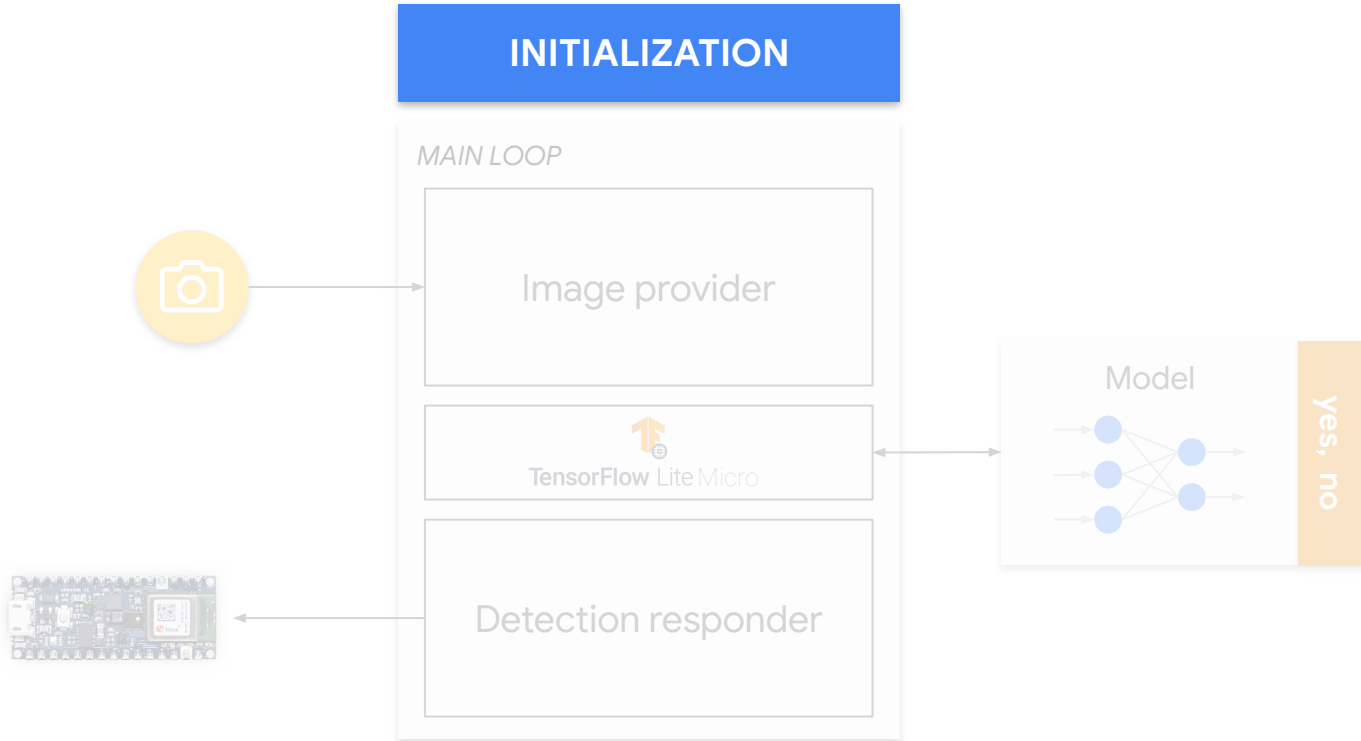
INITIALIZATION

*MAIN LOOP*

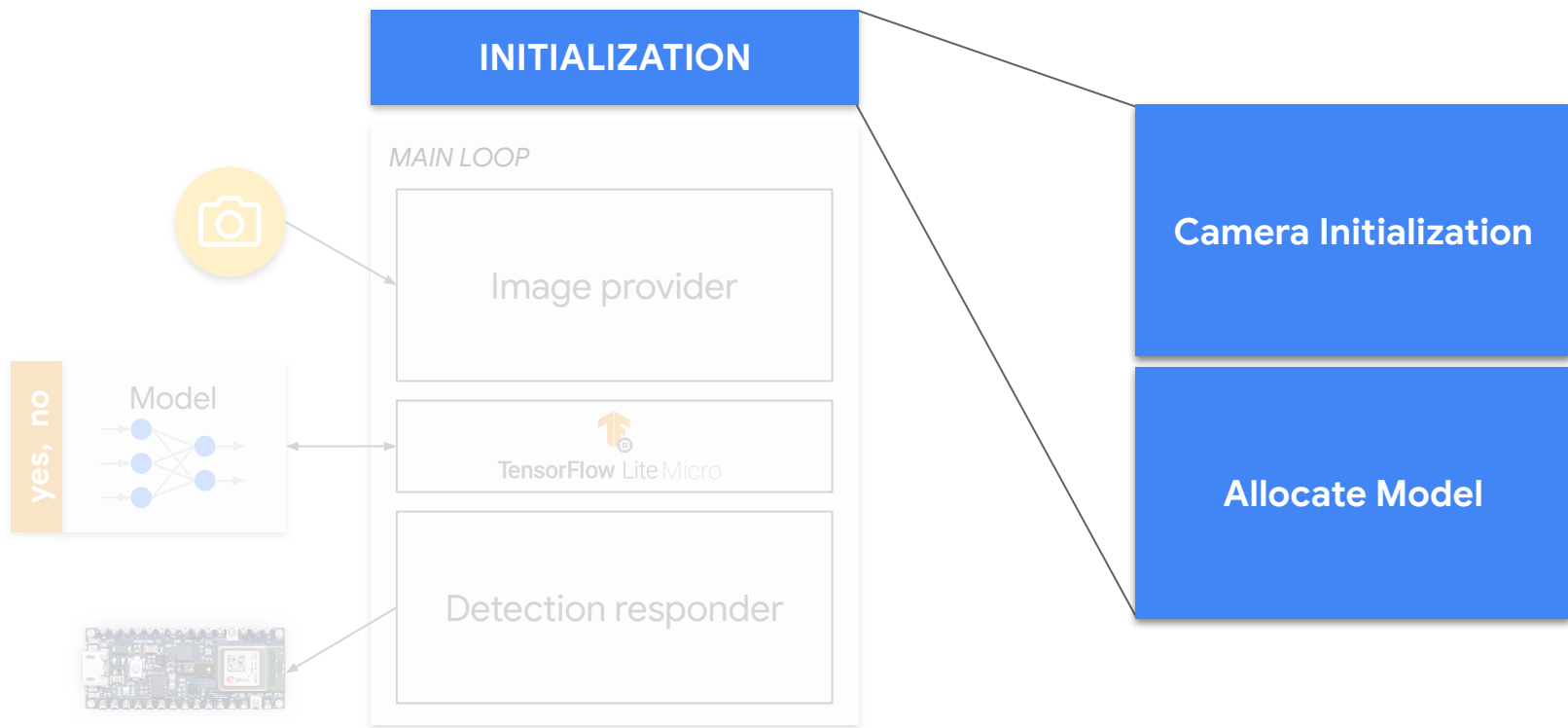
# Person Detection Components



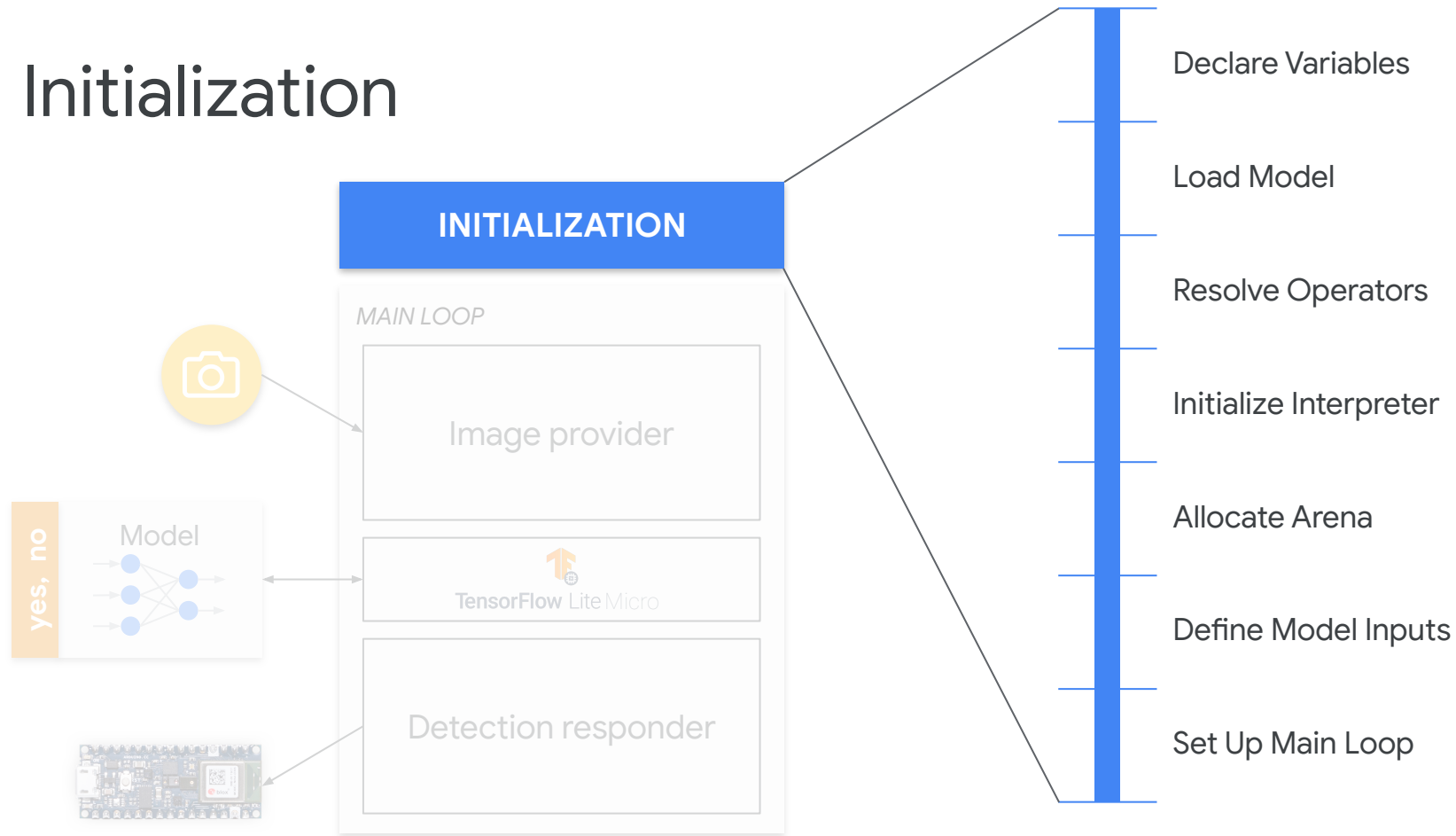
# Person Detection Components



# Initialization



# Initialization



# Initialization

Camera Initialization

Allocate Model

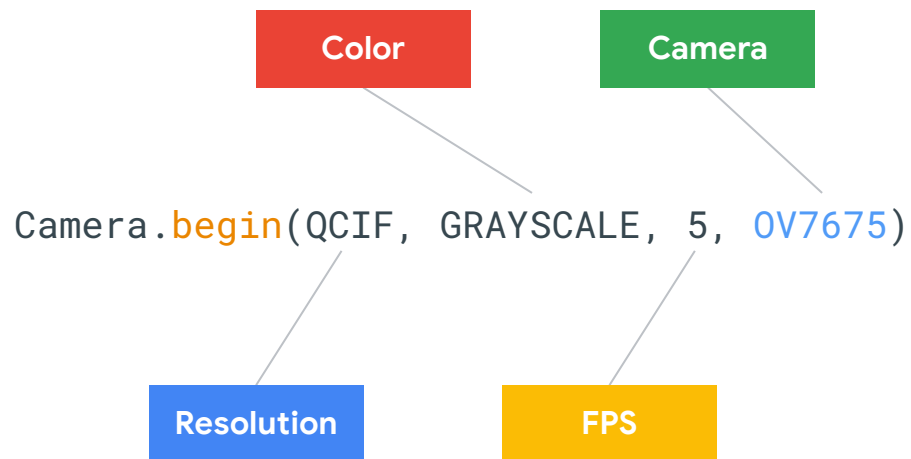
```
// Initialize camera if necessary
if (!g_is_camera_initialized) {
    if (!Camera.begin(QCIF, GRAYSCALE, 5, OV7675)) {
        TF_LITE_REPORT_ERROR(error_reporter, "Failed to
                                initialize
                                camera!");
        return kTfLiteError;
    }
    g_is_camera_initialized = true;
}
```



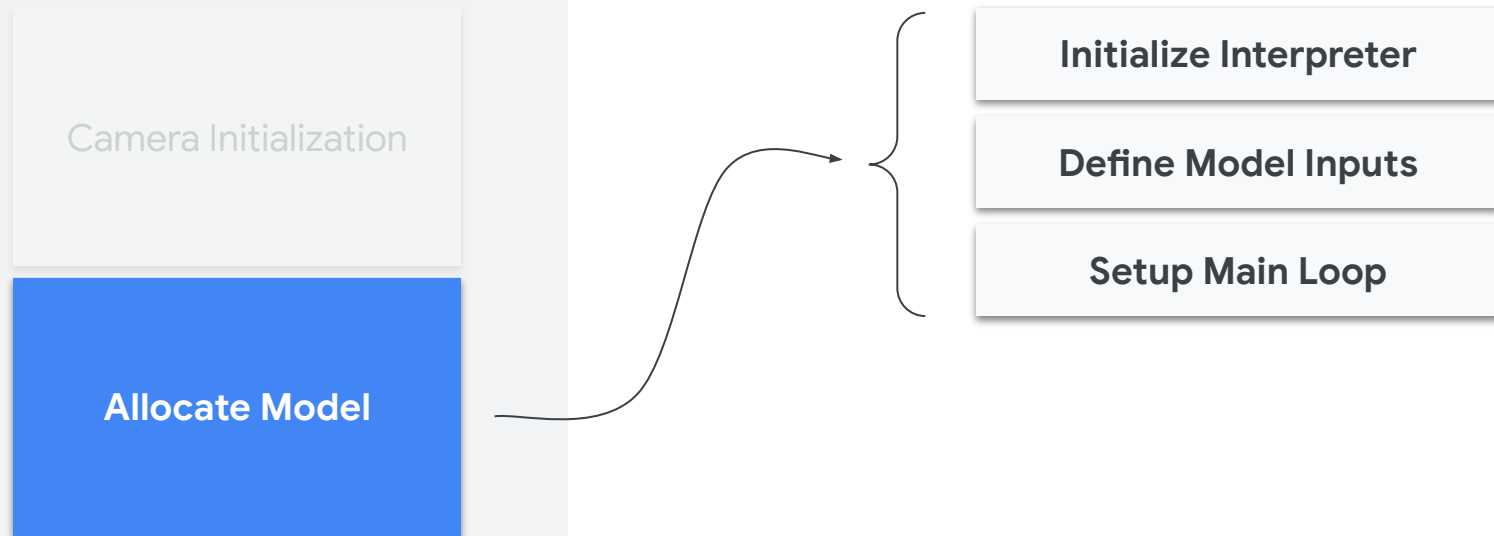
# Initialization

Camera Initialization

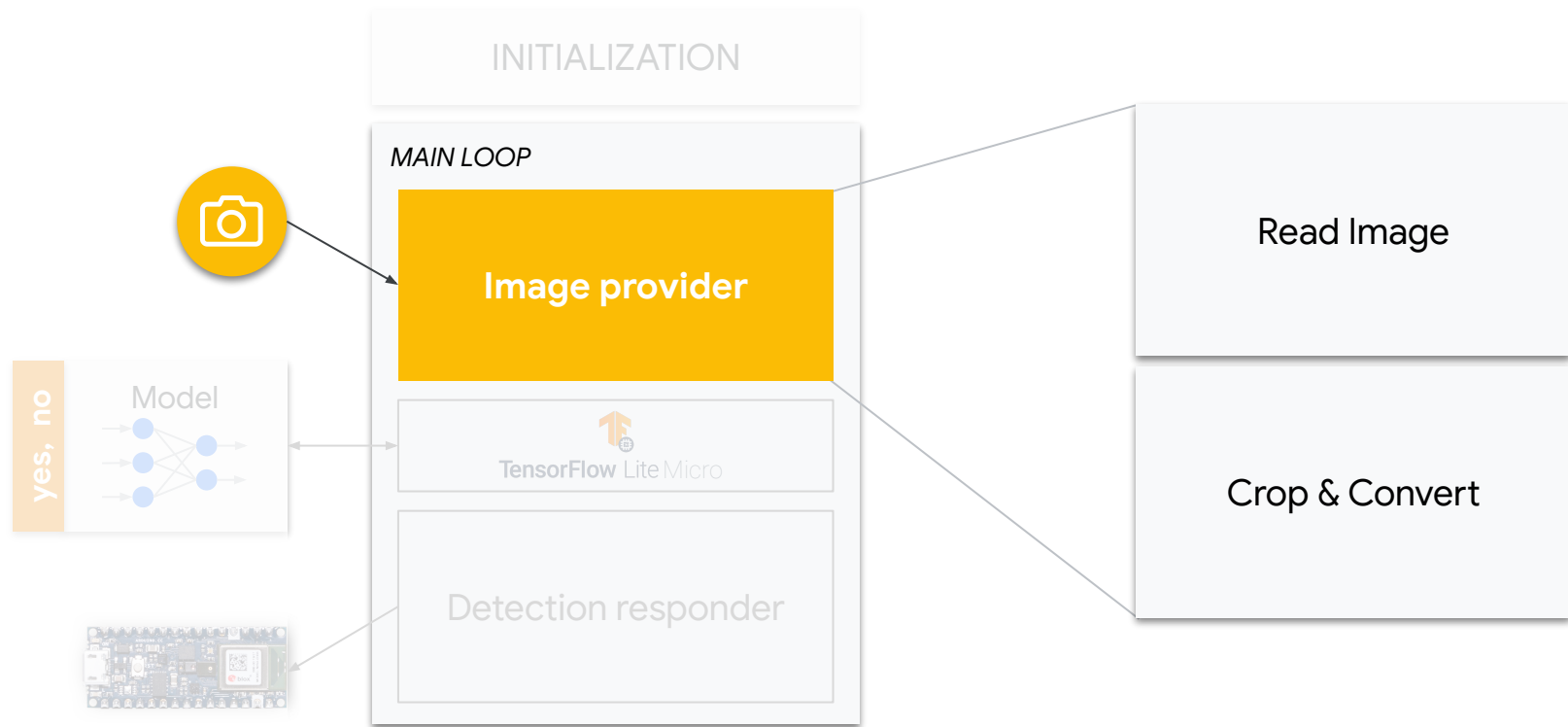
Allocate Model



# Initialization



# Pre-processing



# Pre-processing

Read Image

Crop & Convert



QCIF

144

176



```
// Get an image from the camera module
TfLiteStatus GetImage(tflite::ErrorReporter* error_reporter,
    int image_width, int image_height, int channels,
    int8_t* image_data)
```

# Pre-processing

Read Image

Crop & Convert



QCIF

144

176



```
// Read camera data  
Camera.readFrame(data);
```

# Pre-processing

Read Image

Crop & Convert

QCIF

144

176

Square

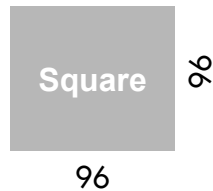
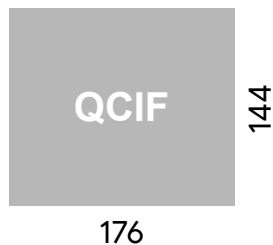
96

96

# Pre-processing

Read Image

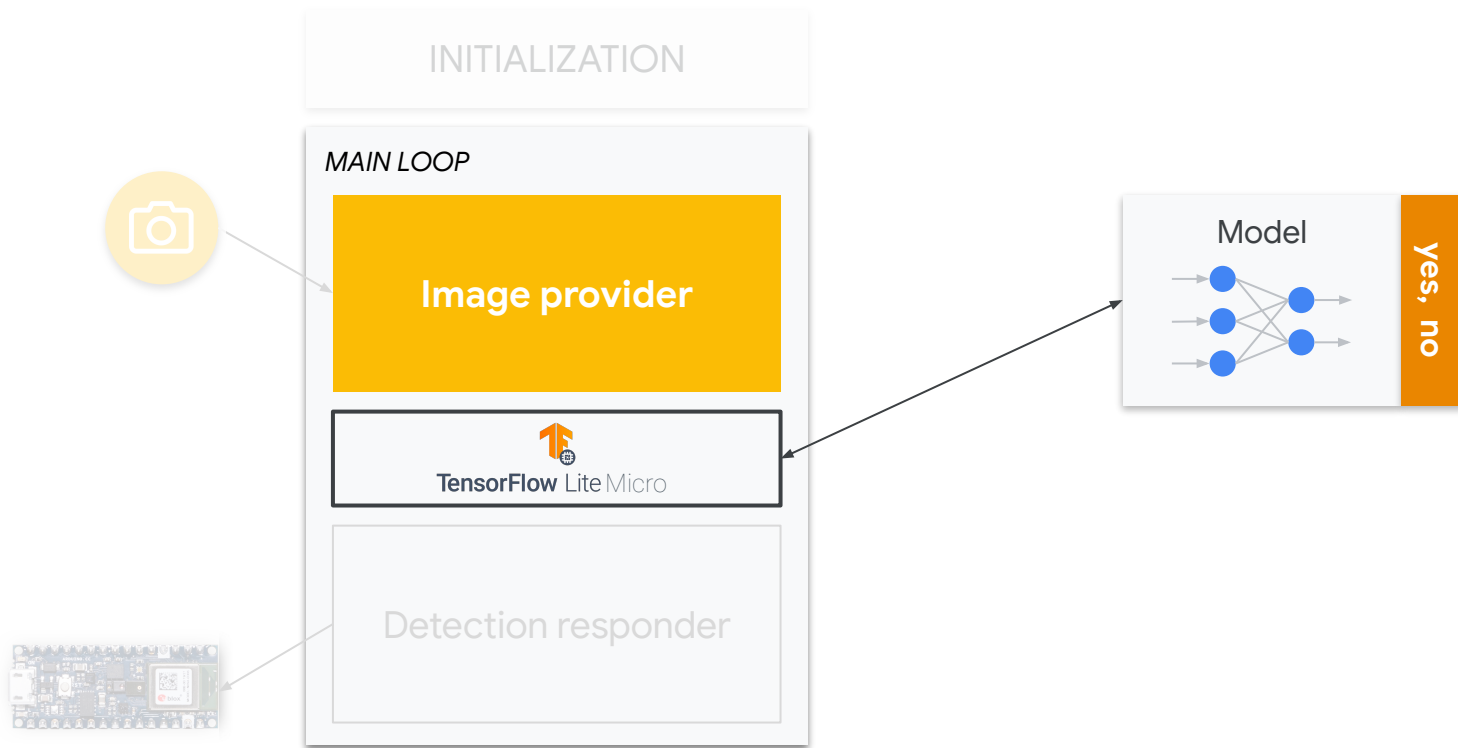
Crop & Convert



```
int min_x = (176 - 96) / 2;  
int min_y = (144 - 96) / 2;  
int index = 0;
```

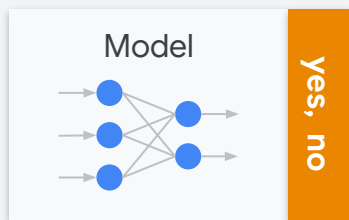
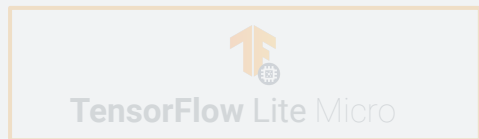
```
// Crop 96x96 image. This lowers FOV, ideally we should downsample  
for (int y = min_y; y < min_y + 96; y++) {  
    for (int x = min_x; x < min_x + 96; x++) {  
        image_data[index++] = static_cast<int8_t>(data[(y * 176) + x] - 128);  
        // convert TF input image to signed 8-bit  
    }  
}
```

# Interpreter + Model





# Interpreter + Model



```
kTfLiteOk != vww_interpreter->Invoke()
```

# Post-processing

