

LIVE IMAGE LEARNING AND RECOGNITION

Powered by ArduCam and a NeuroMem network

Applications

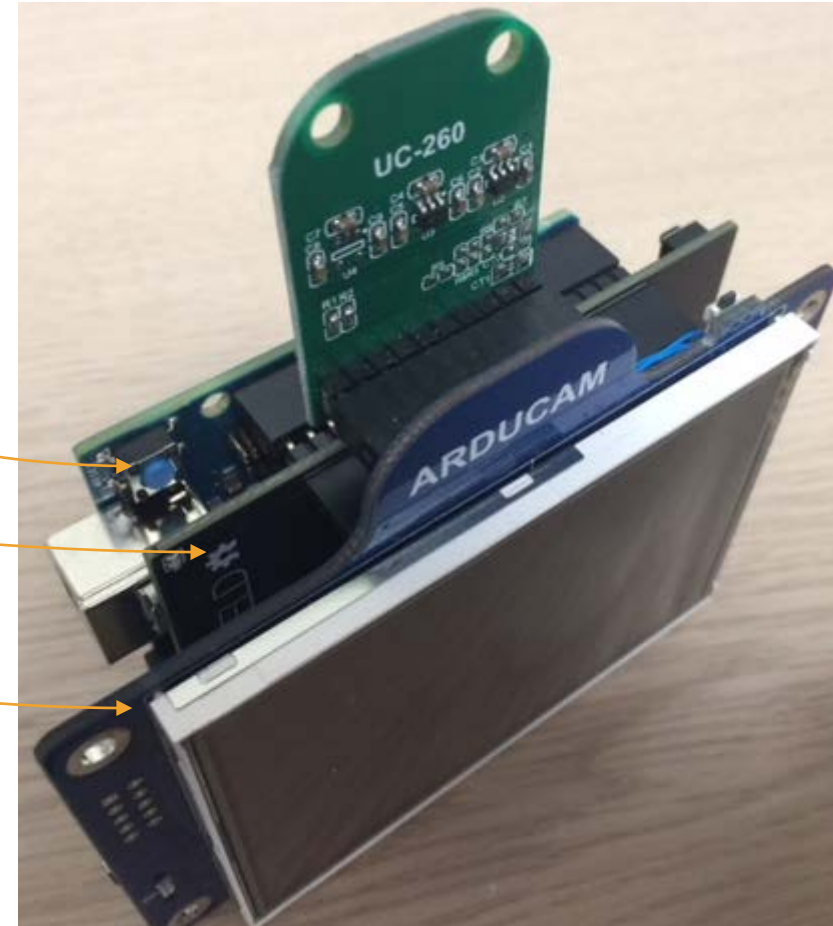
Learning and Monitoring of a fixed region

Including the detection of novelties

- Inspect a part passing on a conveyor
- Detect if someone is coming towards a door
- Monitor a cat door to ensure a raccoon is not sneaking in
- Etc.

Assembly

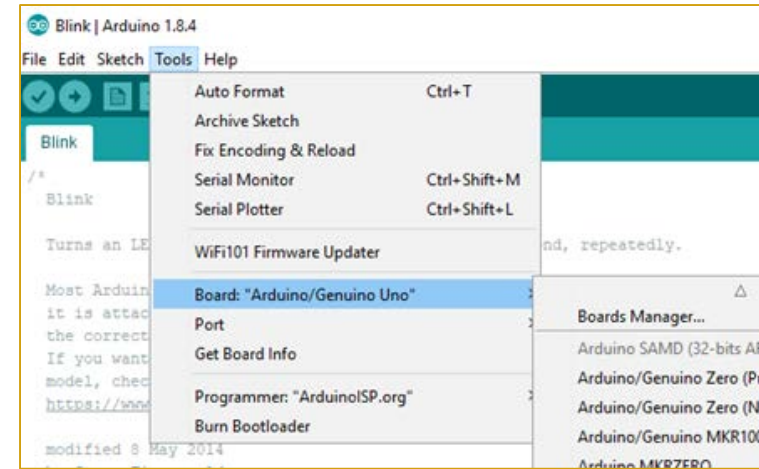
- Arduino microcontroller board with at least 5 KB memory
- NeuroShield featuring
 - InvenSense Accel/Gyro
 - 576 NeuroMem neurons
- ArduCam Shield V2 featuring
 - Color low-res CMOS sensor
 - LCD display
- 2 sets of spacers for Arduino
- Custom cable connecting the SPI 6-pin connector of the Arduino 101 to the ArduCam Shield



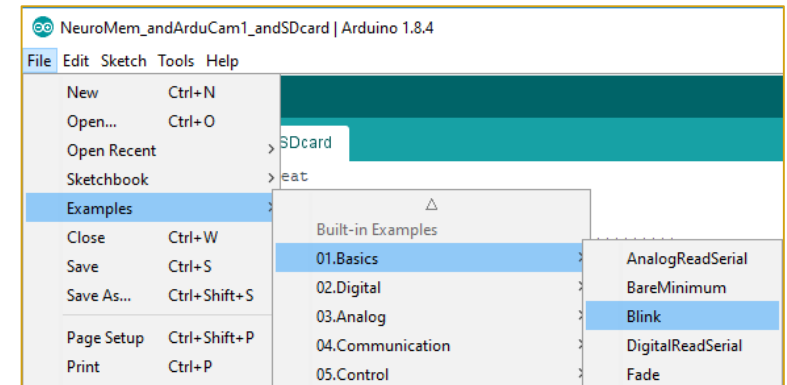
Installation

1. Requirements: Arduino IDE (you can download the latest version at <https://www.arduino.cc/en/Main/Software>)
2. Select your Arduino board under Tools\Board menu. If not in the list, select Board Manager and install its driver
3. Load the File\Examples\Basic\Blink script
4. Upload the script to your board
5. Verify that the LED of the microcontroller board is blinking

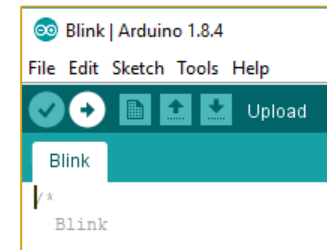
2



3



4



Setup

- Open the serial monitor
- Edit the NeuroMem platform
- Live video appears at the end of the setup
- The blinking rectangle at the center of the screen is the region being taught or monitored
- The recognition report displays "Unknown"

NeuroShield_andArduCam1_andSDcard | Arduino 1.8.4

File Edit Sketch Tools Help

Verify

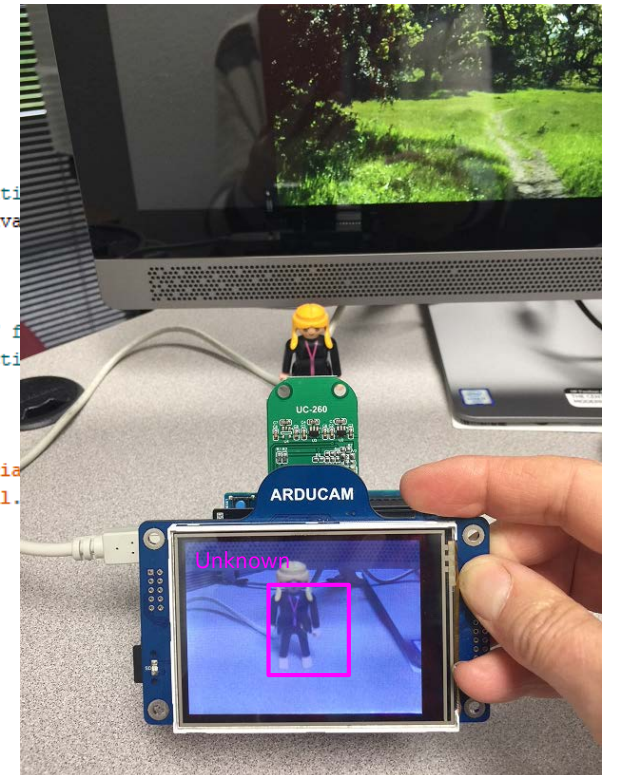
NeuroShield_andArduCam1_andSDcard \$

```
// Initialize the ArduCAM
myCAM.InitCAM();

// Check if the camera module type is OV2640
myCAM.wrSensorReg8_8(0xff, 0x01);
myCAM.rdSensorReg8_8(OV2640_CHIPID_HIGH, &vid);
myCAM.rdSensorReg8_8(OV2640_CHIPID_LOW, &pid);
if ((vid != 0x26) && ((pid != 0x41) || (pid != 0x42))) {
  Serial.println("Can't find OV2640 module!");
} else {
  Serial.println("OV2640 detected.");
}

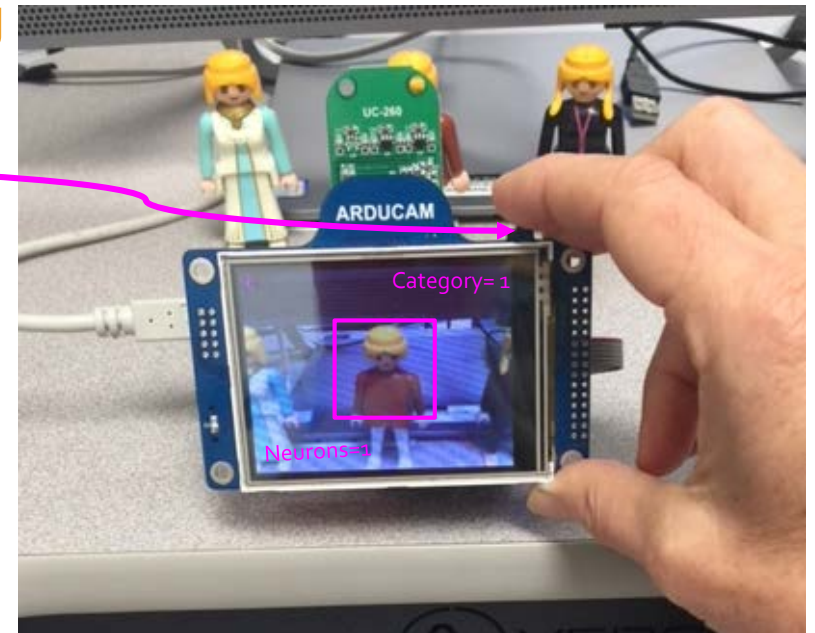
// Initialize the NeuroMem neural network
int NMplatform=2; //NeuroShield
if (hNN.begin(NMplatform) == 0)
{
  Serial.print("\nYour NeuroMem_Smart device is initialized");
  Serial.print("\nThere are "); Serial.print(hNN.nava);
}
else
{
  Serial.print("\nYour NeuroMem_Smart device is NOT initialized");
  Serial.print("\nCheck your device type and connection");
  while (1);
}

Serial.print("Image width="); Serial.print(fw); Serial.print("\n");
Serial.print("ROI width="); Serial.print(rw); Serial.print("\n");
displayLCD_res("Ready", 10,10);
delay(100);
}
```



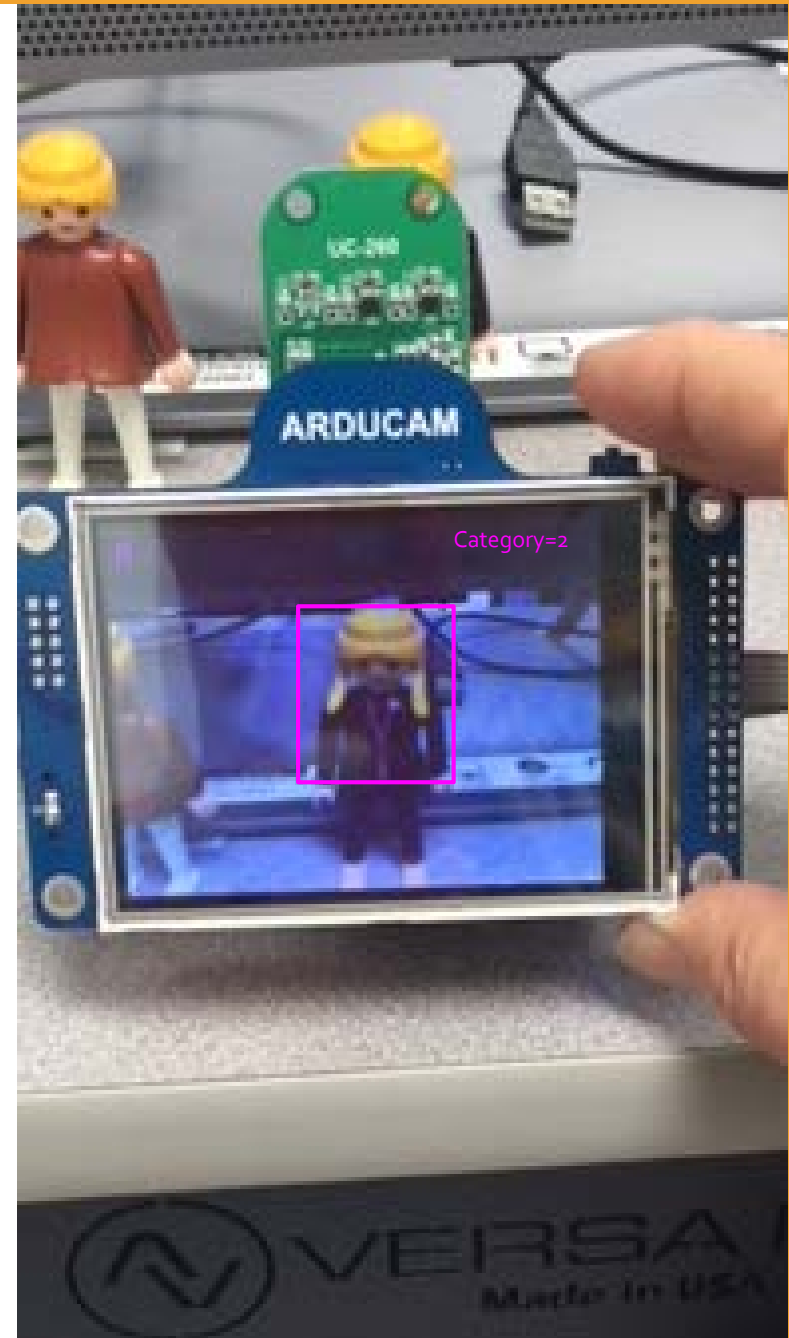
Running the demo

- Move camera so Object#1 appears within the blinking rectangle
- Press the upper right button for 2 seconds
- Bottom line displays "Neurons : 1"
- Upper line displays "Category: 1"
- Move camera so Object#2 appears within the blinking rectangle
- Press the upper right button for 2 seconds
- Bottom line displays "Neurons : 2"
- Upper line displays "Category: 2"



Running the demo (continued)

- Move camera back on Object#1
- Screen should display
 - Upper line displays "Category: 1"
 - If not, move camera to adjust positioning and scaling of the blinking rectangle
- Move camera back on Object#2
- Screen should display
 - Upper line displays "Category: 2"
 - If not, move camera to adjust positioning and scaling of the blinking rectangle
- Try learning a 3rd object!



Summary: A Simple and limited UI

- Operation modes
 - Interlaced video display and recognition
 - User-Interrupt for learning
 - Optional Save of the knowledge
- Input
 - Shutter button used to trigger learning
 - < 2 sec : learn a new category
 - > 2 sec : learn a background/ null category
 - ROI is fixed and centered in video frame
 - Category is incremented each time
 - You cannot show different examples of a same category! This is only a limitation of this script.
- Output
 - LCD overlay after each frame capture
 - ROI rectangle
 - Text result

What is next ?

- Design UI allowing category selection on the touch screen
- Improve UI to better synchronize the teaching instruction to the real-time image acquisition
- And more.....



Troubleshooting (General)

- Script does not load properly?
 - Verify the selected board platform under Tools menu
 - Verify the selected COM port under Tools menu
 - Use the external power supply of the microcontroller board instead of the USB power supply
 - Unplug all shields for the duration of the upload
- Serial monitor is blank and frozen?
 - Verify that its selected baud rate matches the one of the uploaded script
 - Close its window
 - Unplug / Replug the Arduino board (possibly closing / re-opening the Arduino IDE in between)
 - Wait a few seconds
 - Open the serial monitor
- If screen is frozen with the “Ready” message, you will have to reboot the Arduino