Artificial Intelligence and Image Recognition Using HuskyLens

by AkarshA2

Hey, what's up, Guys! Akarsh here from CETech.

In this project, we are going to have a look over the HuskyLens from DFRobot. It is an AI-powered camera module that is capable of doing several Artificial Intelligence operations such as Face Recognition, Object Recognition, and Line Recognition, etc. It is somewhat similar to the MatchX module which we discussed some time back in this project. As the MatchX module was a bit expensive, I decided to make something similar on my own and for that, I found HuskyLens as a great choice because it is cheaper in comparison to the MatchX module and can do everything that the MatchX can except one i.e transmission of data and for that purpose we will interface the Huskylens module with RYLR907 LoRa module from Reyax and we will be good to go. After the interfacing, we will use this HuskyLens to detect an object and send that detected data using the LoRa module to another LoRa module at the receiver side.

So let's get to the fun part now.

Supplies:

Parts Used:

Husky Lens: https://www.dfrobot.com/product-1922....

Reyax RYLR907: https://reurl.cc/n0OOA6

Firebeetle ESP8266: https://www.dfrobot.com/product-1634....

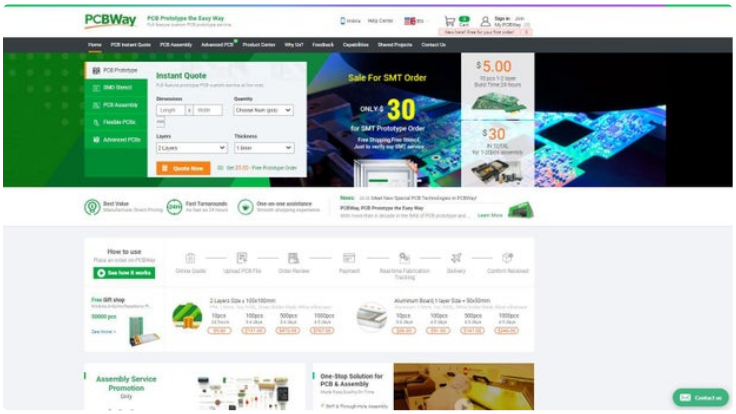
Arduino: https://www.dfrobot.com/product-838.html

https://youtu.be/RzYdEAAPvSg

Step 1: Get PCBs for Your Projects Manufactured

You must check out PCBWAY for ordering PCBs online for cheap!

You get 10 good quality PCBs manufactured and shipped to your doorstep for cheap. You will also get a discount on shipping on your first order. Upload your Gerber files onto PCBWAY to get them manufactured with good quality and quick turnaround time. Check out their online Gerber viewer function. With reward points, you can get free stuff from their gift shop.



Step 2: About HuskyLens Module

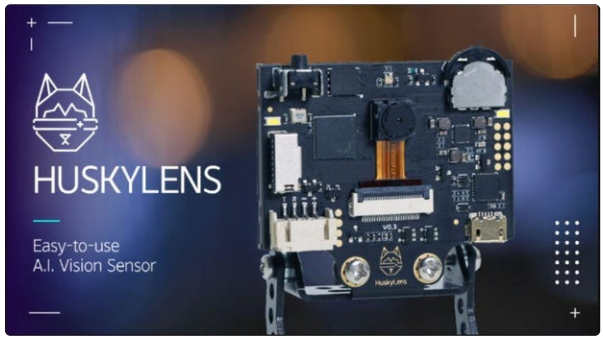
HuskyLens is an easy-to-use AI machine vision sensor with 6 built-in functions: face recognition, object tracking, object recognition, line-following, color detection, and tag detection. It is a pretty neat module that comes with a camera on the front side and an LCD display on the backside and 3 LEDs(2 white and 1 RGB) onboard which can be controlled through the software. It has two buttons on it, One a slider switch to toggle between the modes of operations and a pushbutton to capture and learn about the objects in front of the camera. The more it learns, the smarter it is. The adoption of the new generation AI chip allows HuskyLens to detect faces at 30 frames per second. Through the UART / I2C port, HuskyLens can connect to Arduino, Raspberry Pi, or micro:bit to help you make very creative projects without playing with complex algorithms.

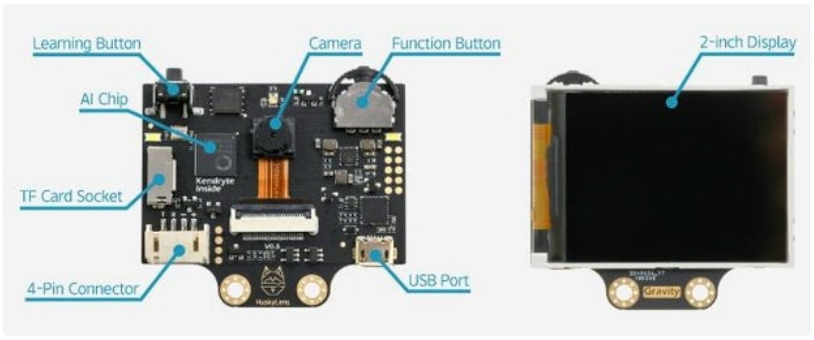
Its Technical specifications are:

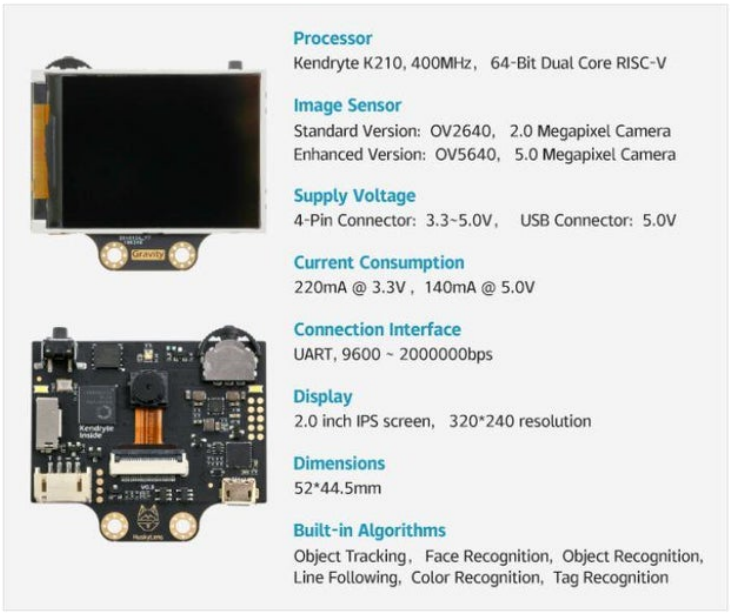
- Processor: Kendryte K210
- Image Sensor:
    - SEN0305 HuskyLens: OV2640 (2.0Megapixel Camera)
    - SEN0336 HuskyLens PRO: OV5640 (5.0MegaPixel Camera)

- Supply Voltage: 3.3~5.0V
- Current Consumption(TYP): 320mA@3.3V, 230mA@5.0V (face recognition mode; 80% backlight brightness; fill light off)
- Connection Interface: UART; I2C
- Display: 2.0-inch IPS screen with 320*240 resolution
- Built-in Algorithms: Face Recognition, Object Tracking, Object Recognition, Line Tracking, Color Recognition, Tag Recognition
- Dimension: 52mm44.5mm / 2.051.75"

Product Link: https://www.dfrobot.com/product-1922.html





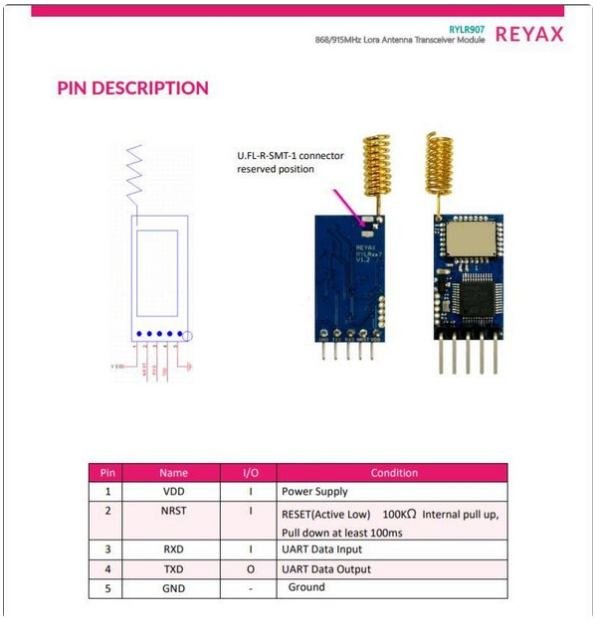


Step 3: About RYLR907 LoRa Module

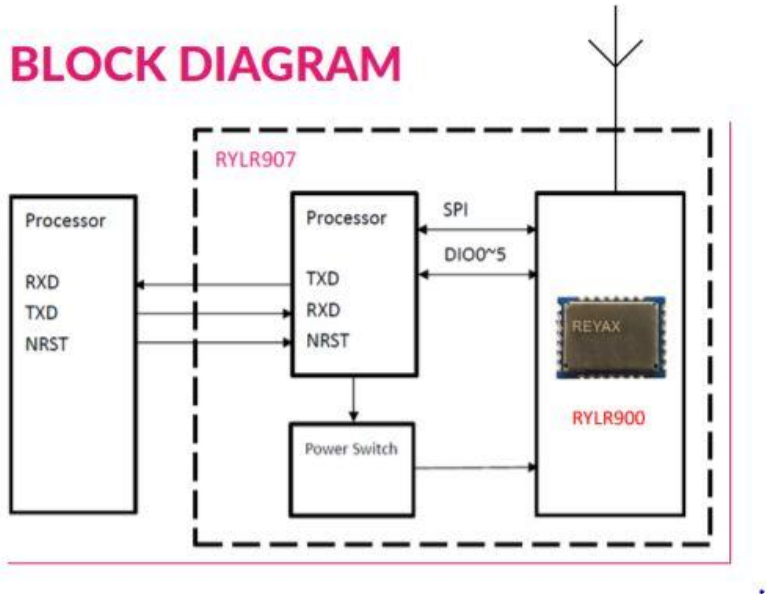
The RYLR907 transceiver module features the Lora long range modem that provides ultra-long range spread spectrum communication and high interference immunity whilst minimizing current consumption. It comes with a Semtech SX1262 Engine which is a powerful one and has an excellent blocking immunity. The RYLR907 has Low receive current and can detect channel motion to set power-saving CAD reception mode on. It is highly sensitive and can be easily controlled by AT commands. Apart from all the above mentioned features, it has a built-in antenna and uses AES128 Data encryption. All these features make it suitable for IoT Applications, Mobile Equipment, Home security, etc.

It can be used to transmit data to a distance in the order of km that to without any internet or other thing. So we will use this LoRa module to transfer the data collected by the HuskyLens from the transmitter end to the receiver end. To get a detailed read about the technical specifications of the RYLR907 module you can head over to its datasheet from here.

Product Link: https://www.amazon.com/....



Pin	Name	I/O	Condition
1	VDD	I	Power Supply
2	NRST	I	RESET(Active Low) 100KΩ Internal pull up, Pull down at least 100ms
3	RXD	I	UART Data Input
4	TXD	O	UART Data Output
5	GND	-	Ground



Step 4: Setting Up the Transmitter and Reciever Sections.

In this step, we are going to do the connections part of the project. First, we will connect the HuskyLens with the RYLR907 LoRa module this will make the transmitter side and after that, we will connect the LoRa module with an ESP8266 to make the receiver end which will receive the data sent by the transmitter and will display it on the Serial Monitor of the Arduino IDE.

The Steps to connect HuskyLens with the LoRa module are as follows:

- Connect the Vcc and GND Pin of the HuskyLens to the 5V and GND of the Arduino respectively.
- Connect the pins R and T of the HuskyLens to the Pin No. 11 and 10 of the Arduino respectively.
- Now take the LoRa module and connect its Vcc pin to the 3.3V output of the Arduino and GND pin to the GND of the Arduino.
- Connect the Rx pin of the RYLR907 to the Tx pin of the Arduino through a resistor as shown in the circuit diagram above. The resistor network is required because the Arduino works on a 5V logic level whereas the RYLR907 works on a 3.3V logic level so to bring down 5V to 3.3V these resistors are used.

In this way, the Transmitter section i.e the HuskyLens connections are completed.

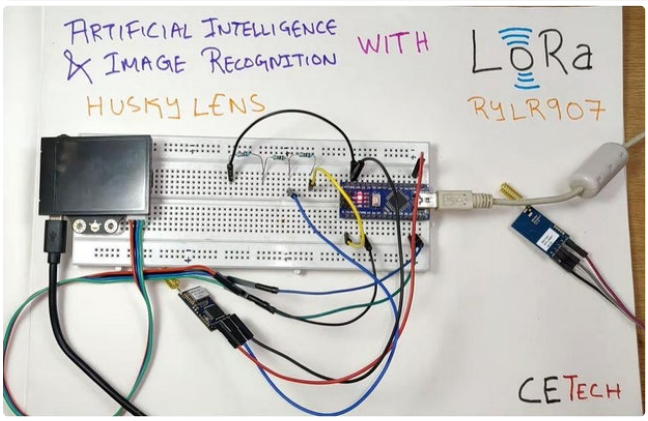
Now for the receiver section, we need an ESP8266 to control the LoRa module for receiving the transmitted data. Connections to be done on this end are as follows:

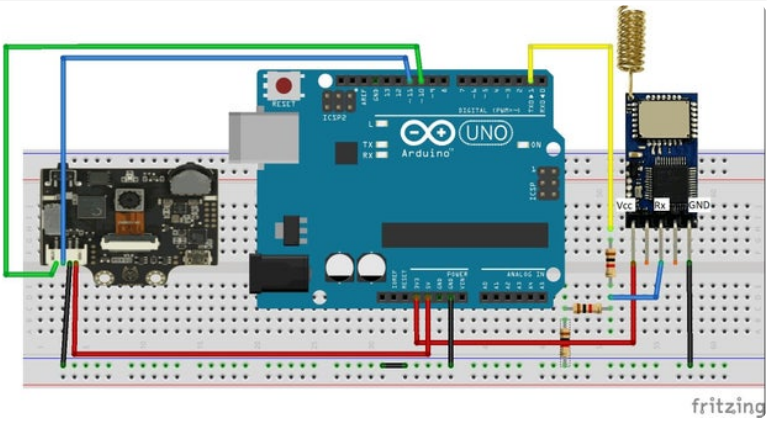
- Connect the Vcc and GND Pins of the LoRa module to the 3.3V and GND pin of the ESP8266.
- Connect the GPIO 15 pin to the Rx pin of the LoRa and GPIO 13 pin to the Tx pin of the RYLR907 module.

In this way, the connections of the receiver side are completed we now just need to connect the modules to our PC and upload the codes of the project. For a detailed description of the LoRa module used here and the connections

to be done at the receiver end, you can check the video above.

https://www.youtube.com/watch?v=jnvik7sUosw





Step 5: Coding the Modules

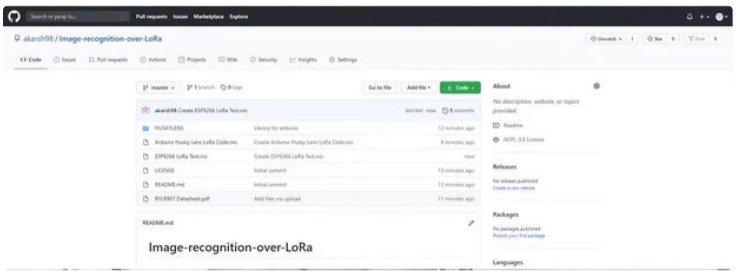
As the Connections for both the sections are done. Now the only thing left is to connect the Arduino and ESP to the PC and upload codes for the project one by one. You can get the codes for the project by heading over to the Github page from here.

- Download the HuskyLens library available on the GitHub page and install it to your Arduino IDE.
- Now open the file named "Arduino Husky Lens Lora Code.ino" this is the code that needs to be uploaded in the Arduino for obtaining data from HuskyLens and send it to the receiver. Copy this code and paste it in your Arduino IDE.
- Connect the Arduino to your PC, select the correct board and COM port, and hit the upload button as soon as the code gets uploaded you can disconnect your Arduino.

In this way, the coding part for the transmitter end is completed. Now you can connect the ESP module which combined with LoRa is going to be used as the receiver.

- After connecting the ESP to your PC open the Github page again and copy the code in the file named "ESP8266 LoRa Text.ino" this is the which needs to be uploaded in the ESP8266.
- Paste the code into the IDE. Select the correct COM Port and board and after that hit the upload button.

As the code gets uploaded you are ready to use the setup.



Step 6: Testing the Link

As soon as the code gets uploaded to both the modules we can check the link by opening the serial monitor initially it will show the message like "No block or arrow appears on the screen". This means that the HuskyLens has not learned about the object it is shown. The object is seen for the first time and is not recognized by the Lens. So to make it recognize the object or face shown to it. We need to show the HuskyLens the object and as soon as it acknowledges the object shown to it press the learning button(push button) this will make the HuskyLens learn about the object and make it recognize the object when anything similar to the learned object is shown. Now as the HuskyLens has learned about the object it will send the data about the object it sees and that data received by the LoRa at the receiver end is displayed on the Serial Monitor.

In this way, we can use AI-powered HuskyLens to recognize objects, collect data about them, and with the help of the LoRa module transmit the gathered data to another LoRa module placed several km away.

So that's it for the tutorial hope you liked it.