

Lesson 9: Safety Helmet Reminder

As we all know, everyone must wear a safety helmet before entering the construction site. Don't look down upon the helmet, it can always save a life in the nick of time. Some people might question how much protection the helmet -- just a hard plastic shell -- can provide?

Exactly, the helmet may not protect you adequately when it comes to some large stones and mechanical components that fall from high. However, the probability is not that high. Usually are something like small stones, bricks, screws, etc. They are not big but have a large power when falling from high, which are easy to get workers injured.



If the worker wears a safety helmet, it can block and buffer most of the energy from the falling object, which can protect people.




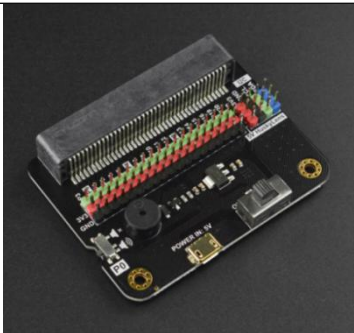
Although it's so important for workers to wear the helmet, there is still someone forgetting to wear it. So, how can we avoid this?

Next, let's use the **object classification** function of HuskyLens to make a safety helmet reminder! Only those workers wearing safety helmets can enter the construction site, while others without helmets cannot.

Learning Goals

1. Understand the principle of KNN algorithm for object classification
2. Learn the object classification function of HuskyLens
3. Use the object classification function of HuskyLens to make a safety helmet reminder

Preparation

	
Micro:bit x1	IO Extender for micro:bit V2.0 x1
	
HuskyLens -- AI Camera x1	DF9GMS 180° Micro Servo x1
	
RGB Full Color Light With x 1	Object Classification Card x 2

Note: The object classification card is the same as the face card.

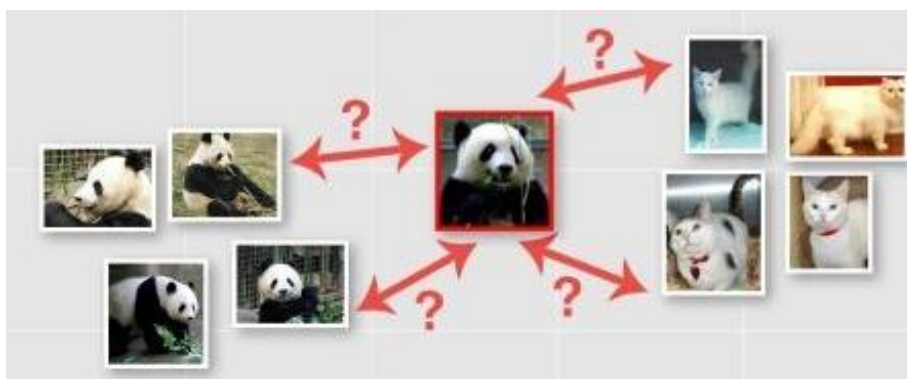
Learning Contents

Operating Principle of Object Classification

KNN can be said to be one of the simplest classification algorithms. At the same time, it is also one of the most commonly used classification algorithms.

1. What is KNN Algorithm

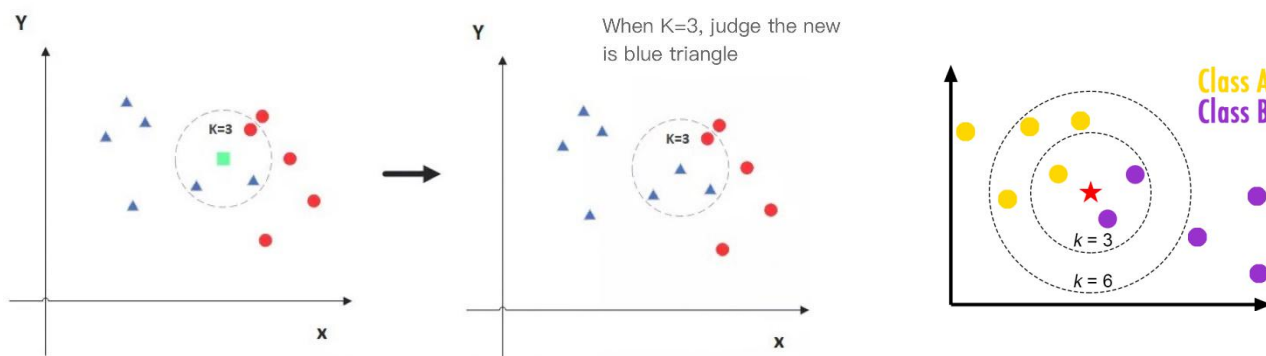
We can see KNN as a kind of rote-learning classifier. It will remember all the training data and will directly match the new data with them. If there are data with the same attributes, the classification of the training data will be used as the classification of the new.



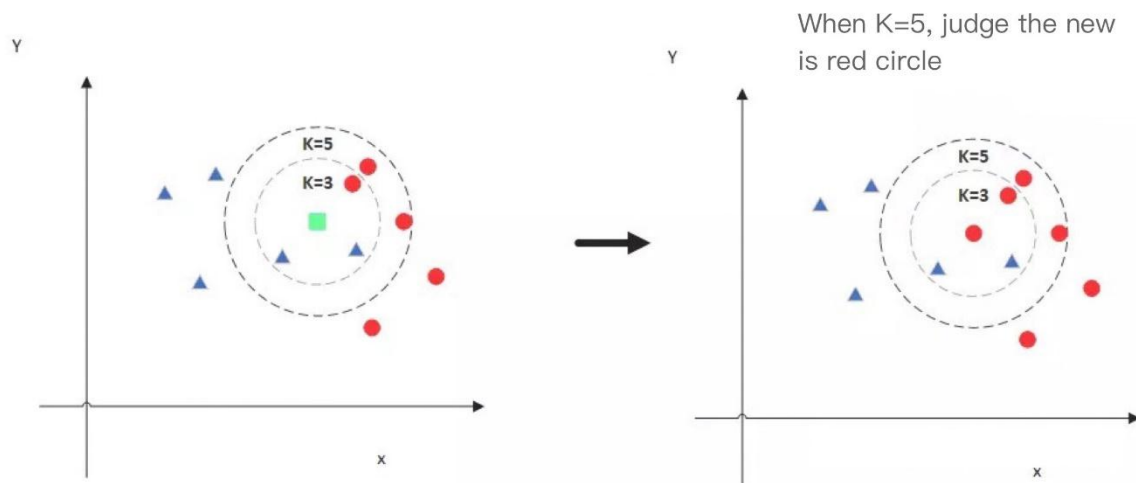
2. KNN Algorithm Principle

The full name of KNN is K Nearest Neighbors. There is no doubt that the value of K is definitely crucial. So what's the nearest neighbor? Let's take a look at the picture.

The green point in the figure below is the data to be predicted. Assuming $K=3$, that is to say the KNN algorithm will find the three points closest to it (here we circle it) to see which kind is more. For example, as there are more blue triangles in the picture, and the new green points will be classified into it.



However, when $K=5$, the judgment will be different. In this case, there are more red circles, so the new green dots are classified as red circles. From this example, we can see that the value of K is very important.



The KNN algorithm will first train multiple data and extracts their characteristics. When predicting a new value, it judges which category x belongs to according to the category of the nearest K points. In the figure on the right, the yellow circle and the purple circle are the training data, and the red five-pointed star is the value to be predicted. When $K=3$ and $K=6$, what the predicted results respectively will be?

3. KNN Arithmetic of HuskyLens

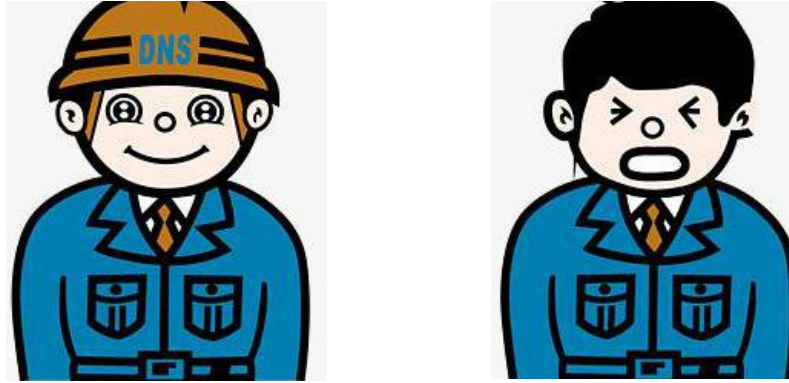
In the object classification algorithm of HuskyLens, the K value is 1. Therefore, when HuskyLens predicts new data, it will compare the data with all the training data, and select the classification of the one with the highest similarity as its output result.

Demonstration of HuskyLens Object classification function

If you want to judge whether a worker is wearing a helmet, you need to classify the image according to the characteristics of the helmet. One type is those wearing the helmet, the other is those not wearing. This project is to use the **object classification** function of HuskyLens to perform safety helmet recognition function.

1. Learn Objects

It can be tested by the following cards.



Firstly, align the big orange frame in the center of the HuskyLens screen with the object to be learned (the worker with a helmet), press and hold the "learning button" to learn from different angles and distances. Then release the button after completing learning the first object. At this point, the screen will prompt: "Press the button again to continue! Press other buttons to end". If you want to continue to learn the next object, you can continue to learn the next object by pressing the "learning button" before the countdown ends. If you no longer need to learn other objects, just press the "function button" before the end of the countdown, or do nothing and wait for its end.

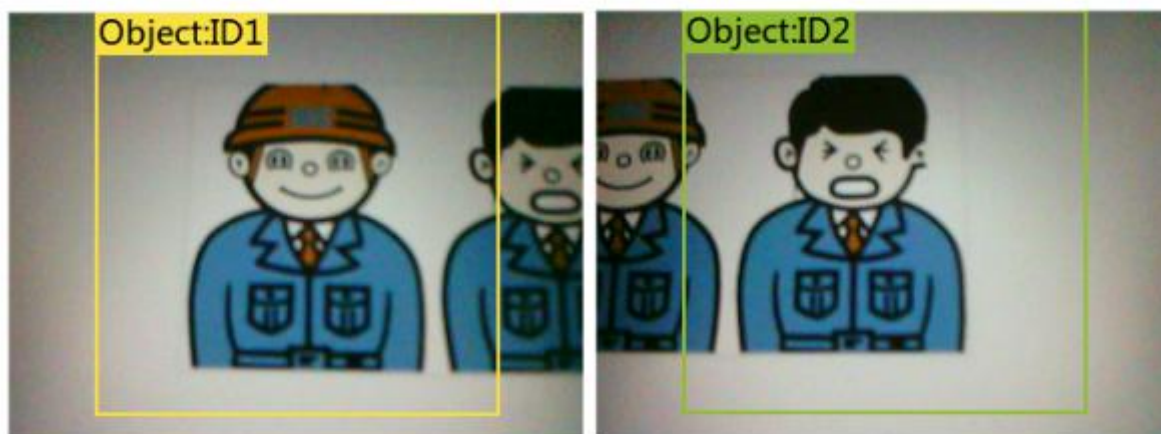


2. Recognize Objects

In the course, we need to continue to learn the next object (the worker without helmet), so press the "learning button" before the countdown ends, and then align the orange frame in the center of the screen with the object to be learned, press and hold "Learning button" to complete learning the second object. So on and so forth.

The object ID is consistent with its input order. That is, the learned objects will be marked as "object: ID1", "object: ID2", "object: ID3", and so on in order. Moreover, different objects have different border colors.

When HuskyLens detect the learned object again, its ID number will be displayed on the screen. As shown in the figure below, when HuskyLens recognizes that the worker is wearing a helmet, the screen displays ID1. If he does not wear it, it will display ID2.



3. FAQ

Q: Can the object classification algorithm get the relative position of the object on the HuskyLens screen?

A: No, it can't.

Under the object classification algorithm, the position of the output box is fixed, and its x and y center coordinates on the screen remain unchanged, so HuskyLens cannot get the relative position of the object on the screen. But it can learn different positions of the object as different IDs, and judge the position by them. For example, in an unmanned vehicle, learn ID 1, 2, 3 as the left, middle, and right sides of the road respectively. By judging the ID, the position of the unmanned vehicle relative to the road can be known.

Q: How to improve the accuracy of recognition under the object classification algorithm?

A: Press and hold the "learning button", you can record the target picture from multiple angles and distances to improve the accuracy.

Project Practice

This project will be divided into two tasks. First, we will learn to use the object classification function of HuskyLens to identify whether there is a helmet or not. Second, add LED strip lights and servos on this basis, and achieve the reminder effect according to their status.

Task 1: Show whether the one wears a safety helmet

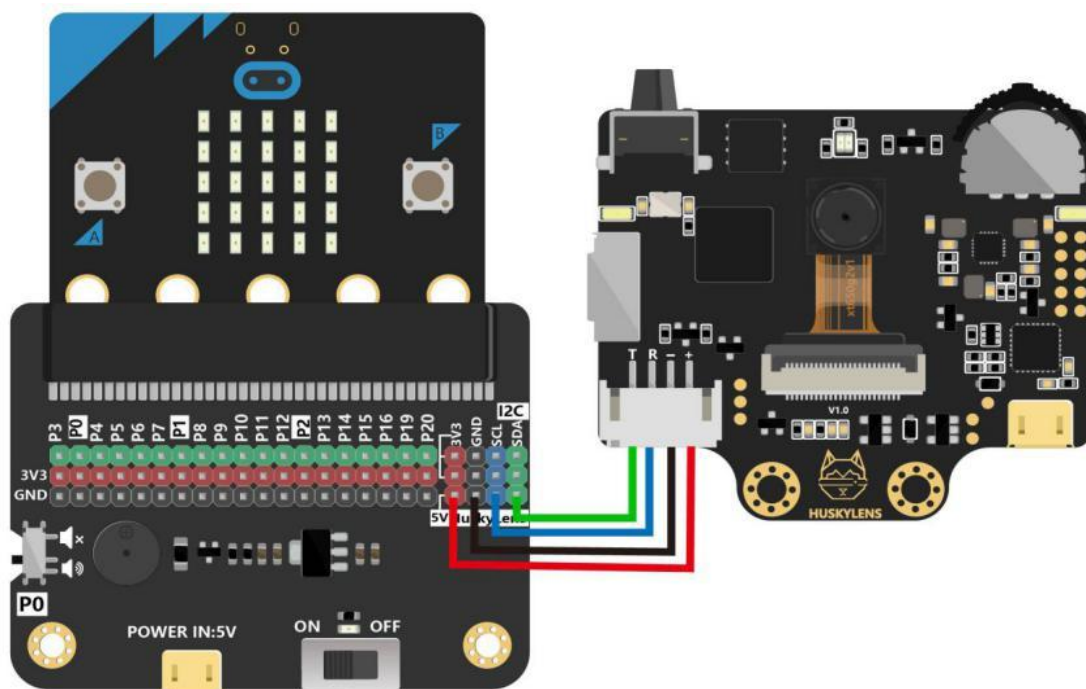
Use the HuskyLens sensor to learn three types of ID – the background, wearing a safety helmet, and not wearing a safety helmet. Write a program to rename the ID. In this way, we can identify whether the worker is wearing a safety helmet.

Task 2: Safety Helmet Reminder

On the basis of task 1, add LED strip lights and servos to judge whether workers entering the construction site with safety helmets, and perform corresponding operations and reminders.

Task 1: Show Whether One Wears a Safety Helmet

Hardware Connection



Program Design

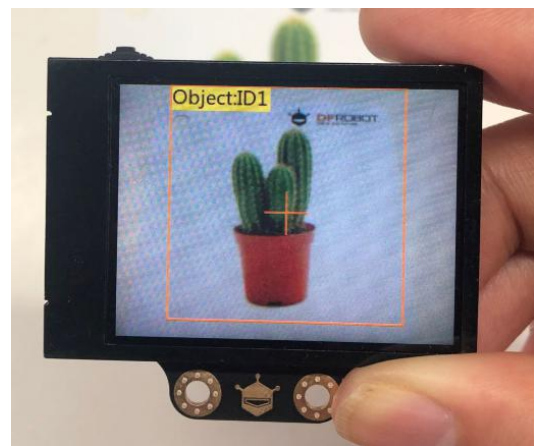
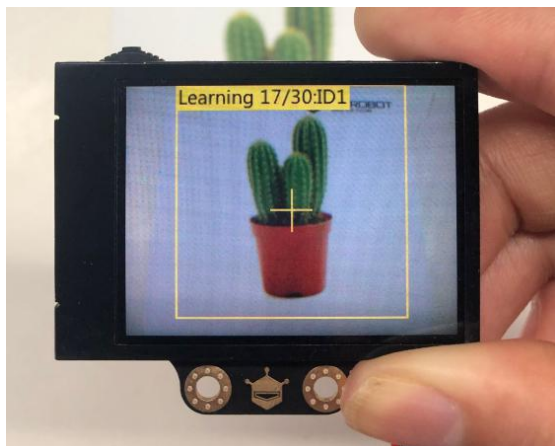
Function Description:

The object classification function can learn different photos of one object, and then use the built-in machine learning algorithm for training. After the training is completed, the algorithm will be named again. When the learned objects reappear on HuskyLens camera screen, HuskyLens can recognize them and display the algorithm name and its ID number.

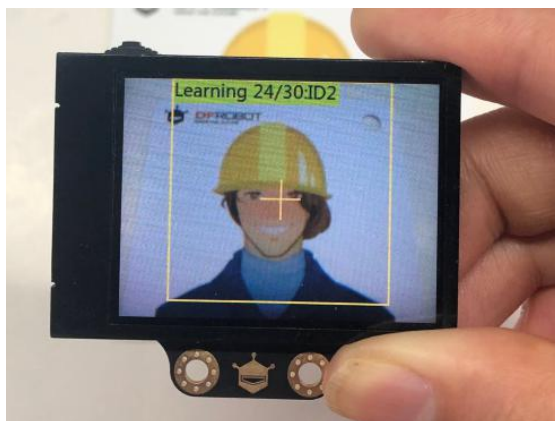
Note: The more photos of a object HuskyLens learns, the more accurate the recognition will be. The default setting is to learn more than one photo.

Learn and Recognize

Select the HuskyLens object classification function, press and hold the "learning button" to learn multiple background pictures to get ID1.



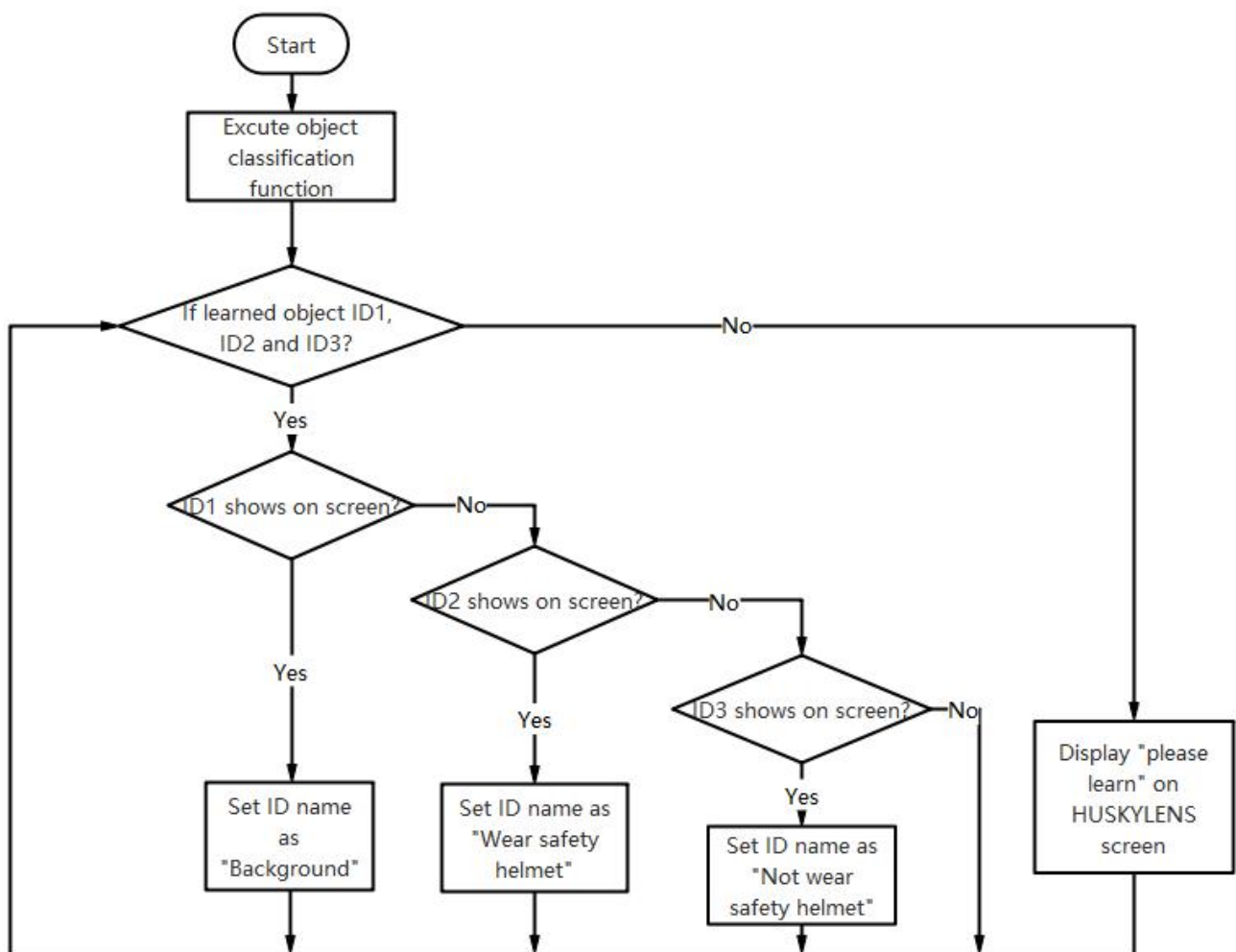
Then learn the character wearing a helmet, long press the "learning button" to learn multiple characters with helmets to get ID2.



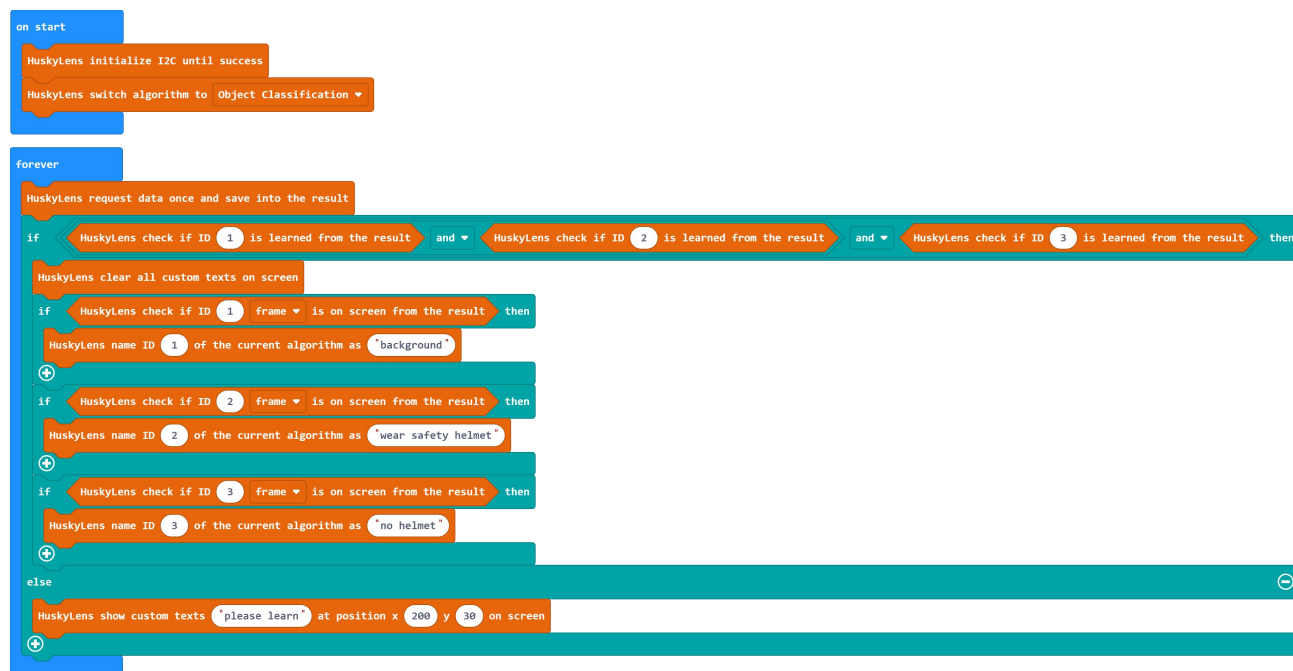
Finally, learn the character without a helmet to get ID3.



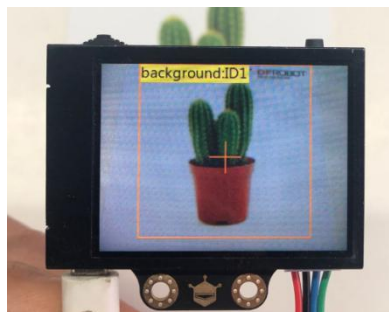
Program Flow Chart Analysis:



Sample Program:



Operating Effect



When the learned background is recognized, display "background: ID1" on the screen.



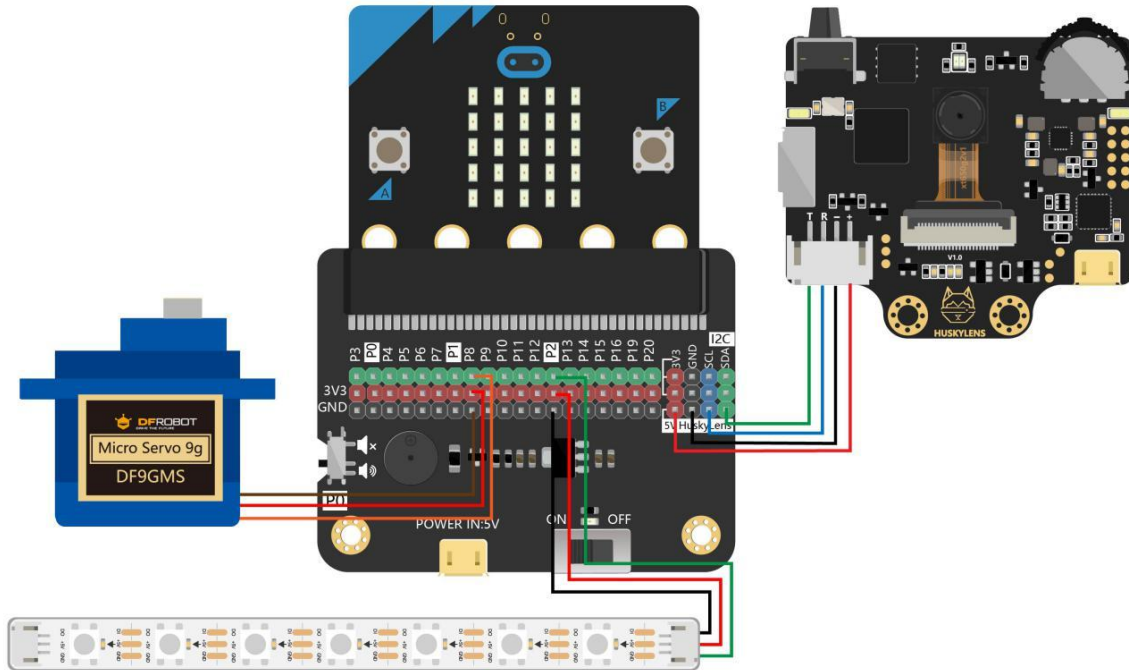
When a person wearing a safety helmet is recognized, display "wear safely helmet: ID2" on the screen.



When a person without a helmet is recognized, display "no Helmet: ID3" on the screen.

Task 2: Safety Helmet Reminder

Hardware Connection



Program Design

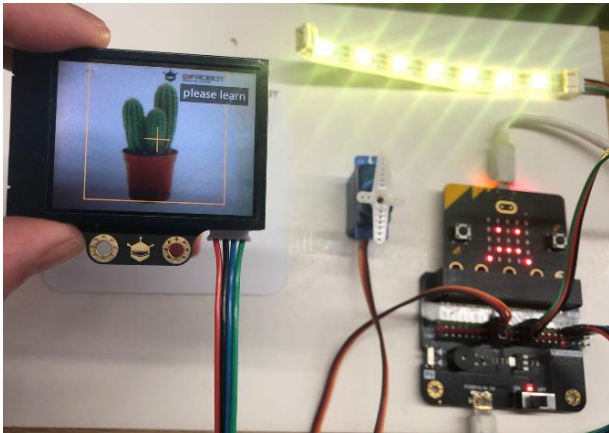
Function Explain:

As the recognition of whether there is a helmet has been realized in task 1, we only need to add and set the state of the servos and the LED strip lights on its basis.

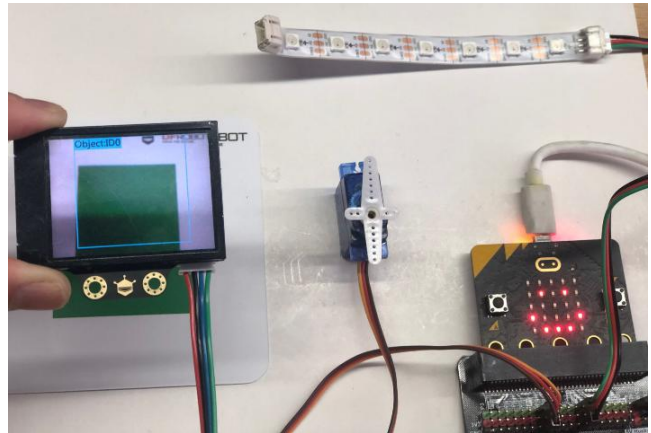

```
on start
  HuskyLens initialize I2C until success
  HuskyLens switch algorithms to Object Classification
  set strip to NeoPixel at pin P2 with 7 leds as RGB (GRB format)
  strip clear
  HuskyLens clear all custom texts on screen
  servo write pin PB (write only) to 90

forever
  HuskyLens request data once and save into the result
  if HuskyLens check if ID 1 is learned from the result and HuskyLens check if ID 2 is learned from the result and HuskyLens check if ID 3 is learned from the result then
    HuskyLens clear all custom texts on screen
    HuskyLens request data once and save into the result
    if HuskyLens check if ID 1 frame is on screen from the result then
      HuskyLens name ID 1 of the current algorithms as "background"
      show icon
      strip show color black
      servo write pin PB (write only) to 90
    else if HuskyLens check if ID 2 frame is on screen from the result then
      HuskyLens name ID 2 of the current algorithms as "wear safety helmet"
      show icon
      strip show color green
      servo write pin PB (write only) to 10
    else if HuskyLens check if ID 3 frame is on screen from the result then
      HuskyLens name ID 3 of the current algorithms as "no helmet"
      show icon
      strip show color red
      servo write pin PB (write only) to 90
    else
      HuskyLens show custom texts "please learn" at position x 200 y 30 on screen
      show icon
      strip show color yellow
      servo write pin PB (write only) to 90
```

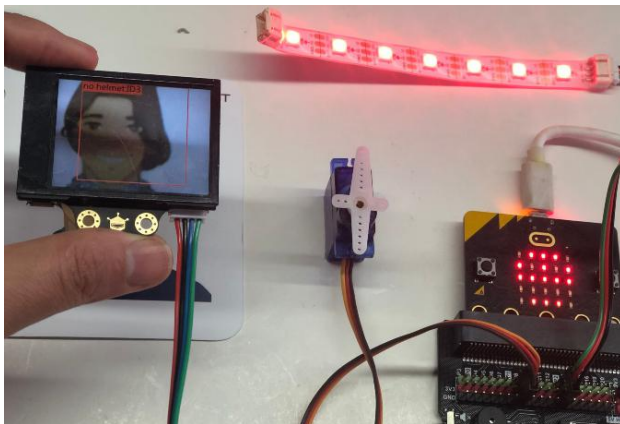

Operating Effect



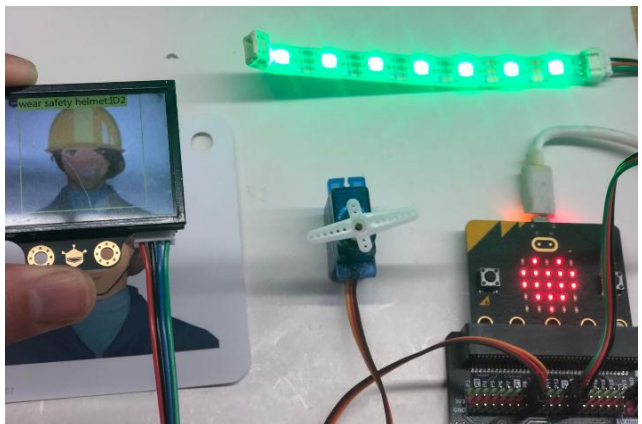
1. Unlearned or incomplete ID, closed, LED strip lights up in yellow.



2. No one at the door, closed, light off.



3. Detected a person without a helmet, closed, LED strip lights up in red.



4. Detected a person with a helmet, open, LED strip lights up in green.

Project Review

In this project, we learned how to use the object classification function of HuskyLens to make a safety helmet reminder with the servo and LED strip. It not only allows workers to pay attention to wearing helmets, but also prevents workers who are not wearing helmets from entering the construction site.

Project Expansion

In this project, we mainly used the object classification function to classify people wearing safety helmets into one category, and those who do not wear into the other category. In fact, there is a

wide range of applications of object classification. Can you come up with other interesting projects by using the object classification function of HuskyLens?

Of course, you can continue to improve the helmet reminder and use the resources at hand to build a real model. The picture below is to use discarded corrugated paper and fix the hardware on it to make a safety helmet detection door on the construction site. Come and have a try!

