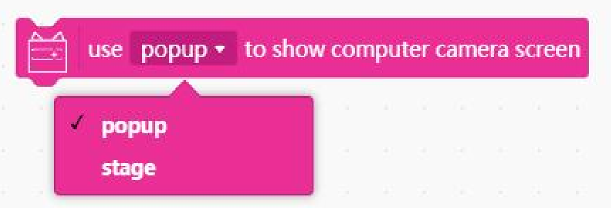
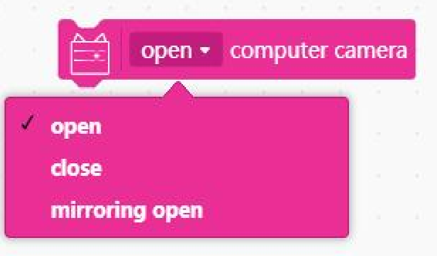
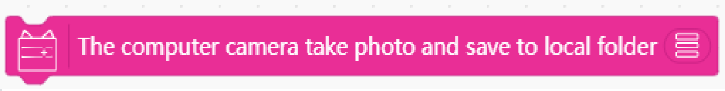
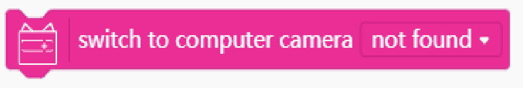
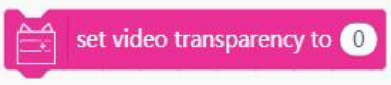
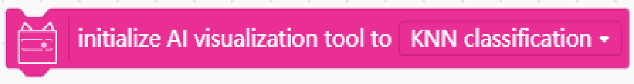
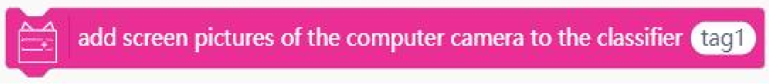
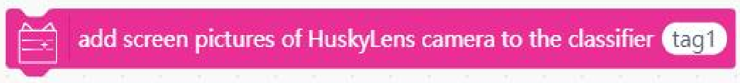
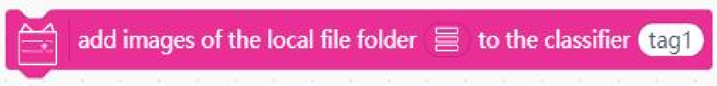
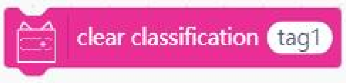
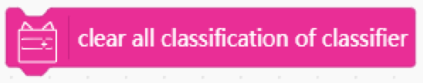
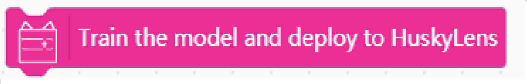
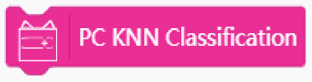
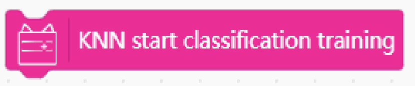
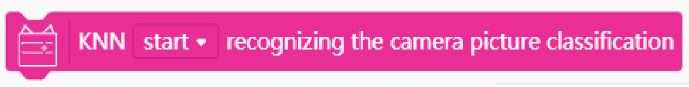
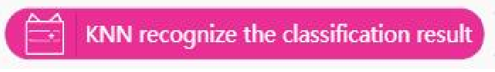
Block	Function Description
 use popup ▾ to show computer camera screen ✓ popup stage	After executing this command, the camera needs to be reopened to take effect
 open ▾ computer camera ✓ open close mirroring open	Turn on the camera. Select "mirroring open" when the screen is flipped. Some PC cameras may take a while to be activated, so it's recommended to add a "wait () seconds" behind.
 The computer camera take photo and save to local folder ☰	Call the computer camera to take a photo and save it on the computer. It can be imported into the KNN classifier for training later.
 switch to computer camera not found ▾	Switch to the computer camera. If there is no camera recognized, restart the software or use camera software on the computer to judge if the camera is recognized.
 set video transparency to 0	Set transparency to let stage background consist with camera view when displaying camera view on the stage.
 initialize AI visualization tool to KNN classification ▾	Initialize AI visual tool, set as KNN classification or facial recognition. Automatically deploy to the corresponding algorithm when deploying to HuskyLens.
 add screen pictures of the computer camera to the classifier tag1	Take a photo by the computer camera and add it to a classification named tag1. Place pictures of the same type into the same category. For example, for a model of rock, paper, and scissors, there should be four categories for all rocks, paper, scissors and backgrounds respectively.
 add screen pictures of HuskyLens camera to the classifier tag1	Get a photo from the HuskyLens camera and add it to a classification named tag1. The classification method is the same as the block above. Note that it takes a few seconds to transfer pictures and the HuskyLens should be connected with Mind+ this time.
 add images of the local file folder ☰ to the classifier tag1	Load several pictures from the local file folder of the computer to a classification named tag1 at once. The classification method is the same as the block above. The path can be delivered as a variable.
 clear classification tag1	Clear all pictures in classification tag1. Note that the result can be obtained by re-training after clearing.
 clear all classification of classifier	Clear all data in the whole classifier. Note that the result can be obtained by re-training after clearing.
 Train the model and deploy to HuskyLens	Train the classifier in visual tool and deploy it to HuskyLens. After deploying, HuskyLens can directly recognize. This block will take some time to run.
 PC KNN Classification	Initialize KNN classifier, load model, and clear the trained data. Run this command before training. There is no need to execute this block multiple times.
 KNN start classification training	Use KNN model to train the pictures in AI visual tool to generate model. Every time a picture is added, it needs to be trained to take effect. Note that the execution of this command requires a lot of computing power, which may cause the computer to freeze. (The higher performance a computer has, the smoother it runs)
 KNN start ▾ recognizing the camera picture classification	Use this block to recognize continuously when finished model training. Note that train first and recognize. Call this block to stop recognizing when adding pictures.
 KNN recognize the classification result	Get recognition results. In the KNN algorithm, for unlearned pictures, it will return the most similar result even if the similarity is only 1%. So it's recommended to learn the background first to remove interference.