Image Classification Game: Part 1

This Snap! game uses Nvidia Jetson capability to classify images.

Offline Snap! downloading

Please download and open Offline version of Snap! for our project. Go to https://snap.berkeley.edu/offline and follow the steps.

Snap! files' downloading

Please open the link Classification Game to download our project on your computer. Probably you would see the xml in raw format. Click the right button of your mouse and save it on the disk.



Web camera Image in Snap!

You can get picture from your web camera in Snap!.

- video capture block to enable video capturing.
- Set video capture to
 Change value of set video transparency block to 0 for clear image.
 set video transparency to
 video snap on stage block reports picture from stage.
 video snap on Stage

Connection to Jetson from Snap!

If you have not imported it yet, please download jetson blocks and import it to your Snap! project.



• Use connect to Jetson url block to connect Jetson.

connect to jetson url ws://ip_address:4040



You need **ip address** of Nvidia Jetson. You can use *ifconfig* command in a terminal to get **ip address**.

• Store the value of jetson_name in a variable.



• Store the value of **connect to Jetson url** block in a variable for later use.

set jetson to connect to jetson url join well (jetson_name) 4040 🕠

Response from classification

Here we will send **video snap on stage** to Jetson for processing. Jetson will respond back class name, confidence value and class ID.



Only **class name** and **confidence value** will be used in this example. This project does not use **class ID**.

- Use get response from Jetson block to send image , and get class name and confidence value.
 - $\circ\,$ First input slot is for jetson variable that stores websocket data.
 - \circ Second input slot is for **costume** you want to be classified by Nvidia Jetson.



Class name and confidence value

This section will demonstrate how to handle **response** variable to access **class name** and **confidence value.**

• class name is the 2nd item of 1st item of response block.

from response.

item 2 of item 1 of response
 confidence value is 2nd item of 2nd item of response block.
 round 100 × item 2 of item 2 of response
 Multiply confidence value by 100 to get percentage of confidence .

Speech functionality

Speech functionality is available as a library in Snap!. Select *export libraries* from settings then choose *speech module* .

• Use join block to create text like I am confidence percent sure, it is class name .



Repeat block for game

Last step is adding loop for the game.

• Use repeat block and put script inside of it.



Image Classification Game: Part 2

Please open the link Classification Game: Extended to download the extended version of our project on your computer.

Start Camera

Drag and drop the **start video** custom block to start using web camera.

start video

Connection to Jetson

Repeat the steps from Part 1 to connect to the Jetson Computer.



Game Initialization

Use When flag clicked , variable setting and our custom blocks for initializing the game



Game process control blocks

In the game process we use **when** block to catch the click event from the stage and change the current page and broadcasting to tell the other blocks that game is starting.

when start label is clicked
set current_page to (game page)
render current_page
broadcast start_game to all • •
wait until not mouse down?

Also we added additional blocks to resume the game after it was stopped.



Main part: sending images to Jetson and having fun with our pets! :-)

We use familiar **when** block to listen to the start event. Then the block for receiving classification data is used in connection with **analyze** block



Inside **analyze** block we compare the class name with preset class names of food, that our pets consume, and broadcast to them. In case if the detected object is not suitable for any of them, we run **speak** block to pronounce the name of the object.

+ analyze + found + class +
if (class_name) = (an1_food)
broadcast rabbit_choice - to Rabbit
else
if class_name = an2_food
broadcast monkey_choice to Monkey <
else
if (class_name) = (an3_food)
broadcast mouse_choice - to Mouse - (
else
if (class_name) = (an4_food)
broadcast squirrel_choice v to Squirrel v <
else
if an5_food_list) contains class_name
broadcast worm_choice v to Worm v 4
else
speak join Iguess, it is class_name () with en-GB accent
and pitch 🚺 rate 🚺

In the script part of each sprite there are already blocks, responsible for handling their choice events. We use **speak** , **play sound** , **animate** blocks to animate the sprites.



From: https://student-wiki.eolab.de/ - HSRW EOLab Students Wiki

Permanent link:

https://student-wiki.eolab.de/doku.php?id=snapcon2022:image-classification-game&rev=1659790298

Last update: 2023/01/05 14:38

