# 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.

	🕒 Attas//www.githabusercontent.com	rEOLab H	GRACSnapcon workshop/main/sna	g/snapcon_vile_final_base.x	ni - (hibisate)						
C A	D https://www.gthubusercontent.com/EDUs-HSRW/Soupcon					-19			InPrivate		
Chains (1): dige billow wonto (NEAK wonto (NEAK wonto (NEAK wonto (NEAK wonto (NEAK NEAK) Status (NEAK) Status (NEAK)	The subscription of Leff approximately provided the provided try or other subscription of the subscriptio		adazzorostan (Jar ser egoda (Y) Nie I Jaka AV, YVIIII (J Scher (J AV) Legel In (J Scher (J AV) Nie (Scher (J Scher (J AV))) Scher (J Scher (J Scher (J AV))) Scher (J Scher (J Scher (J Scher (J AV))) Scher (J Scher (J Scher (J Scher (J AV)))) Scher (J Scher (J Scher (J Scher (J Scher Back)) Ferniment Solve an Solve an	aktivity NICayler Hitgyrd 29(49/Acytyl Leff Yallo 29(4)/Acytyl Leff Yallo 29(4)/Acytyl Leff Yallo 20(4)/40(4)/20(4	2421 - Scholl That John M Glipsach (1964) (Sala / Sala 2713 - Official Alexandrollow) 2713 - Official Alexandrollow 2714	ycylani NYvani ECRIM ClQVic LQGL1r UlpG2x oMCl1p SkdSik tl0k5g ZkdSak tl0k5g ZkdSak tcrVfr 18Lov+ Calore2 TQLetp pyyAxV uSC shu 2AS 196 k10657 anE2ak b5cr/max	✓ 5400mb √550 fW1. 559,110m 599,110m 599,110m 599,110m 599,110m 599,110m 599,110m 599,110m 598,110m 799,110m 790,	221340 unger: exeWer; 1141aV p61212; what's 122900 122900 122900 p840000 p840000 p84000 p84000 p84000 p84000 p840000 p840000 p840000 p840000 p840000 p840000000000	LIMPTCFhty article/vision griburt/ciol Zysrokka/PP Systeks/Service/Ser	yeC315 WC4ksz ahck33 28+041 2005/ke 31huC4 31hUC4 31hUC4 31hUC4 31hUC4 31hUC4 31hUC4 30hUC4 3	144 1992 1941 1945 1940 1
3ktt.2f68CvOv cxfPANS6f4/ 0xfFRKy31230	adiv jivt jeviše + ježav (STP latbil) (Bivgigg (YYY) iš jebovici) is toogae ju/ zbit jisuvii Zeethi ASTV laciali z Bivi + at 90 vizreli IVI SwPDE ("Prvs and fili Parlas a NGTV Carlos NGNE ublige VIP Stitu i Grift pas PL at aPhilip 2ad z+ angle Thak Lav	a a	Read aroust	CDI-SHR+Q	Autopretoknyskefe Gravus 2004 NH37 - VEL 2010 L100001 V214	disjs30 78ny11 Maailie	ADDE:/0+ ad5/075 trv5gC+/	octiver[1, Skaliky Try813	/j7qai2te JTe6vo2F9 Gr1/LignQ	j/g3ks tQqOey FridaPr	8
Japte0#yq0p)	0 to Blog 26C FARGNEC INE on Left Hidd Force?/UC +An Filt PHOT7516+e3c 120800 10% 50y 125 x 1 431 e6F +Blockgy/124+0012 (bitweden / +2 25 556 (H380%) (50 2004 THEFT - All well Address - An Anna Anna Anna Anna Anna Anna Anna		Add page to Collections		-12E+SHScS720doE1 TAHLABET7PC80aCS+	lveagb	Que ive	A087:15	UMIdAzłow	PLANCIAN	ior

## 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 video

• Change value of **set video transparency** block to 0 for clear image.

set video transparency to 🕕

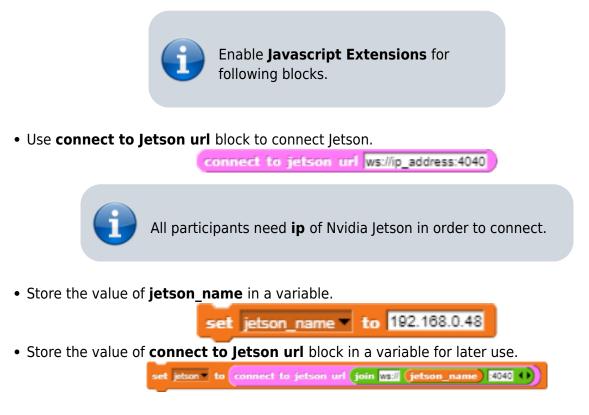
• video snap on stage block reports picture from stage.

video snap v on Stage v

Last update: 2022/08/06 snapcon2022:image-classification-game https://wiki.eolab.de/doku.php?id=snapcon2022:image-classification-game&rev=1659796371 16:32

## **Connection to Jetson from Snap!**

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



## **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**.

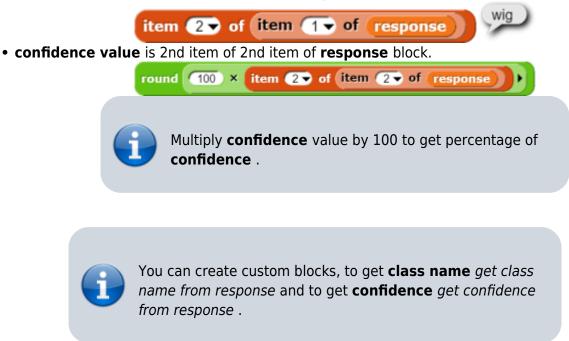
- Use get response from Jetson block to send image , and get class name and confidence value.
  - $\circ$  First input slot is for **jetson** variable that stores websocket data.
  - $\circ$  Second input slot is for **costume** you want to be classified by Nvidia Jetson.



## **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.



## **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.



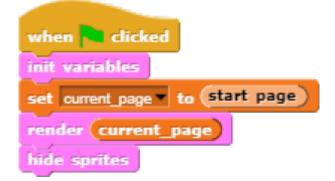
## **Connection to Jetson**

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  4
wait until not mouse down?

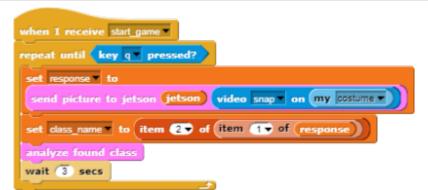
Also we added additional blocks to resume the game after it was stopped.



# 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

Last update: 2022/08/06 snapcon2022:image-classification-game https://wiki.eolab.de/doku.php?id=snapcon2022:image-classification-game&rev=1659796371 16:32



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 v to Mouse v 4
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 (1) rate (1)

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://wiki.eolab.de/ - HSRW EOLab Wiki

Permanent link: https://wiki.eolab.de/doku.php?id=snapcon2022:image-classification-game&rev=1659796371



Last update: 2022/08/06 16:32