

# AI FOR GOOD Summer Camp



INDIANA UNIVERSITY  
BLOOMINGTON



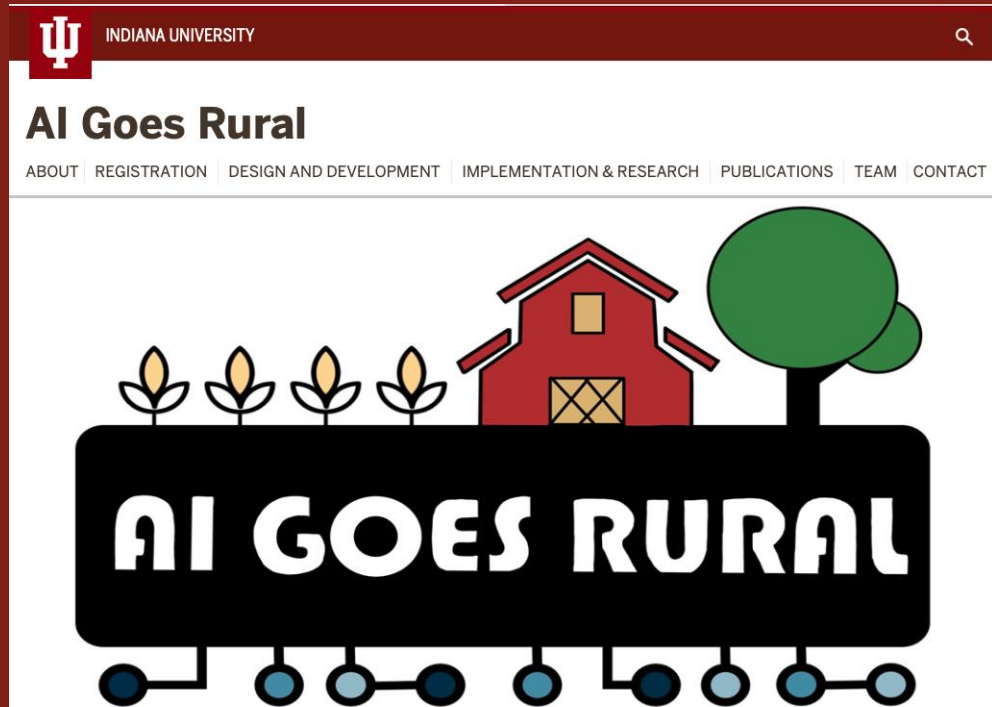
# Webpage link

<https://bit.ly/iuaigood>



# Day 1

# Introduction



**Check  
your understanding of AI**

# AI vs. Not AI



Toaster

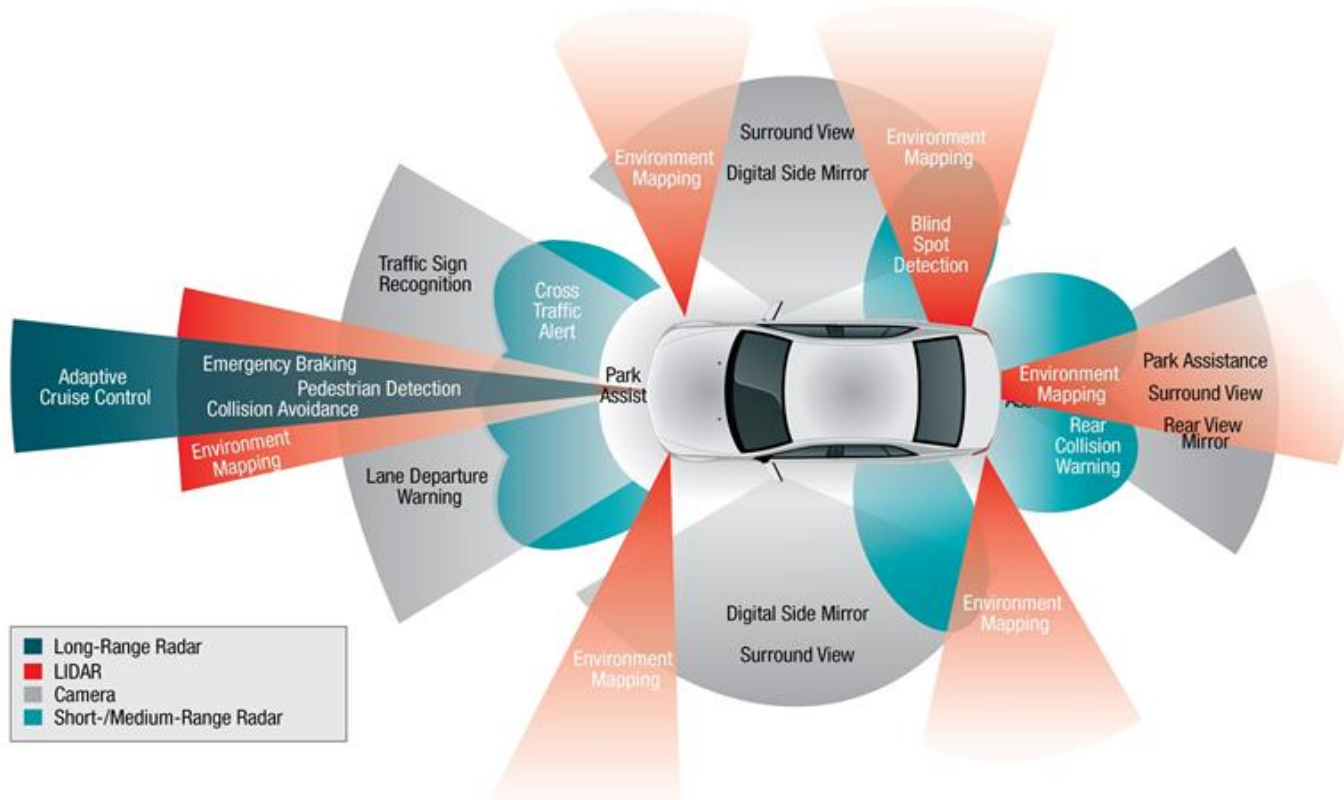


Siri on iPhone





Barcode scanner



# Smart Car



# Washing Machine

**NETFLIX**

Netflix



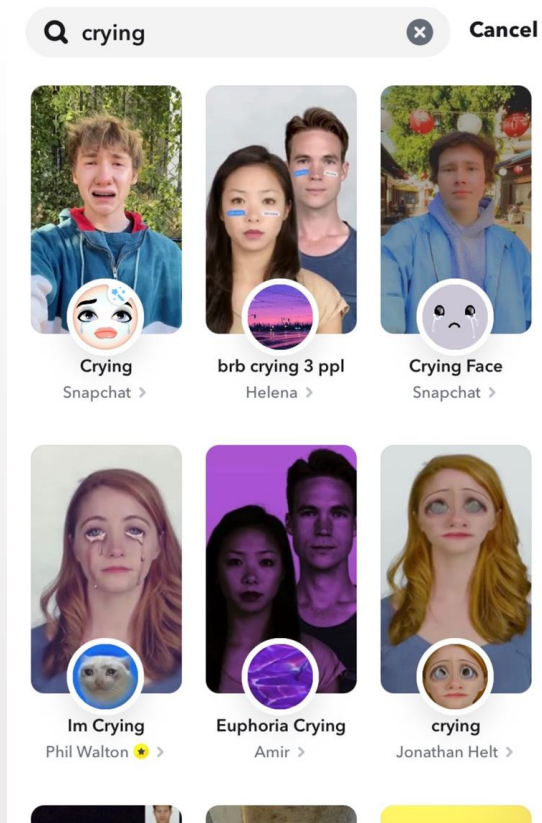
Industrial robot



Delivery Robot



Mixer



# Tiktok facefilter





Electronic toll collection

**What is AI?**

**What does AI Stand for?**



# What is Artificial?

- Made by **humans**; produced by humans (opposed to natural): artificial flowers

# What is Intelligence?

- An organism **uses data** to **improve** decision making





# What is AI?

- A way for a computer program to work “intelligently”
- The art of teaching computers how to "think."
- A discipline concerned with the designing of computers that make **predictions** and **decisions**.



# How AI works?



# What is ML(machine learning)?

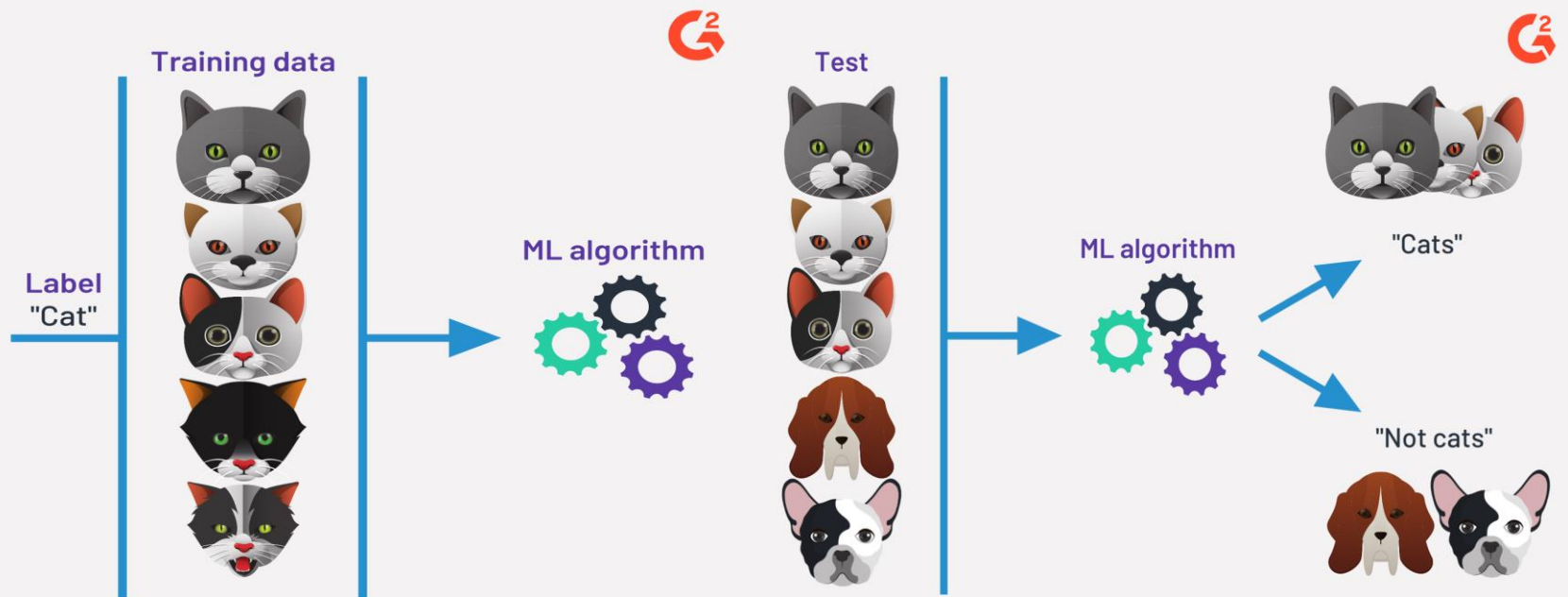
- Machine Learning (ML) is a process when AI learns for itself through data and experience
- Enable AI systems to come up with their own solutions
- Complete certain tasks at great speed and scale
- When you think of ML, who or what do you think of?





# What is ML (Supervised)?

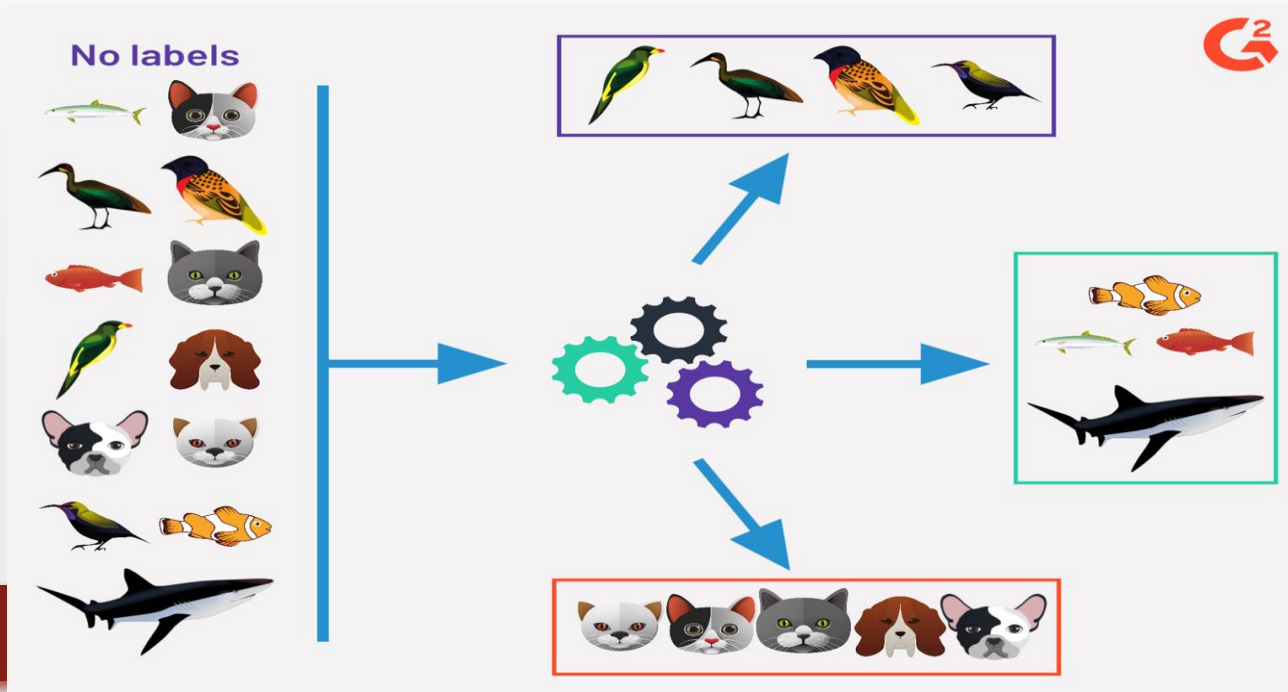
Train an AI model with labeled data





# What is ML (Unsupervised)?

Train an AI model without labeled data





# Introductory activity

# AI for Oceans

- [AI for Oceans](#)
  - How can we help protect the oceans using ML?



# AI for Oceans

These were the most important fish parts:

- body
- color
- dorsal fin
- tail
- mouth

Click individual fish to see their information.

Train More Continue

This screenshot shows the main interface of the 'AI for Oceans' application. A central panel displays a list of fish parts: 'body', 'color', 'dorsal fin', 'tail', and 'mouth'. The 'body' and 'color' items are highlighted in blue. Below the list, a small white robot icon labeled 'AI' is visible. The background features a dark blue ocean scene with various colorful cartoon fish. At the top right, there are icons for a checkmark, a lock, and an information symbol. At the bottom, there are two buttons: 'Train More' on the left and 'Continue' on the right.

These were the most important fish parts in determining whether this fish was "circular" or "not circular".

- body
- color
- tail
- mouth

Train More Continue

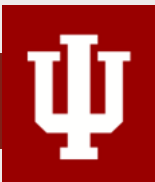
This screenshot shows the same interface as the previous one, but with a different selection. The 'body' item is highlighted in green, while 'color', 'tail', and 'mouth' are highlighted in red. The text above the list indicates that these parts were used to determine if a fish was 'circular' or 'not circular'. The 'AI' robot icon is now positioned to the right of the list. The 'Train More' and 'Continue' buttons remain at the bottom.

These were the most important fish parts in determining whether this fish was "circular" or "not circular".

- body
- color
- dorsal fin
- mouth

Train More Continue

This screenshot shows the interface with a different selection. The 'body' and 'dorsal fin' items are highlighted in red, while 'color' and 'mouth' are highlighted in green. The text above the list is the same as in the previous screenshot. The 'AI' robot icon is now positioned to the left of the list. The 'Train More' and 'Continue' buttons remain at the bottom.





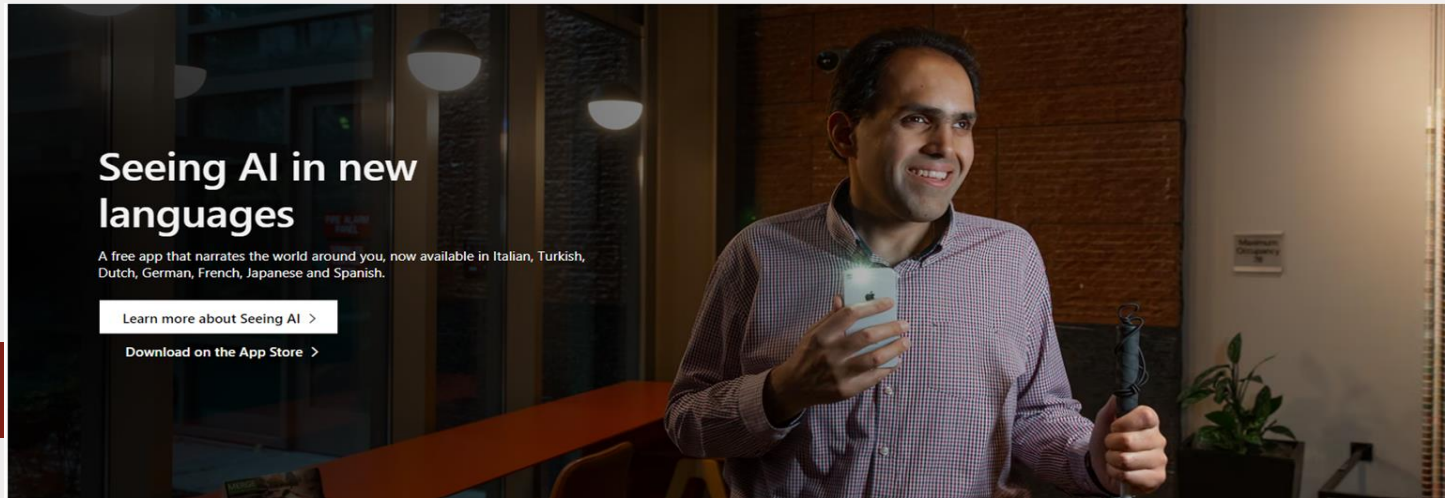
## Let's Think....

- How do AI know what a “fish” may look like?
  - Why did we need to label and distinguish different objects in “AI for Oceans” activity (e.g., “fish” versus “not fish” )? How do you call this step?
  - How important this step was for machine learning?
- Please leave your thought on the google jamboard ([AIFORGOOD Discussion board](#))



# Seeing AI

- [Seeing AI](#)
  - Explore it using the app with iPad
  - How does the Seeing AI help low vision people?
  - How does ML work in the Seeing AI?
  - Please leave your thought on the google jamboard on 4th slide ([AIFORGOOD Discussion board](#))



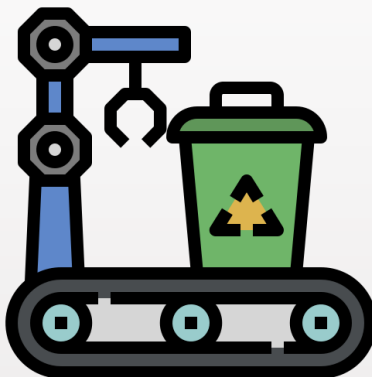
# Hands-on Activity



# Choose the activities interested most



AI emergency  
text message



AI garbage  
collector



AI Squat  
machine



# Hands-on Activity 1

## AI emergency text message





# AI emergency text message

- Problem to solve
  - Emily wants to send out the emergency text messages to her families but she often face difficulty typing on her phone due to a time-sensitive situation or if she has difficulty using her hands or fingers to type. Thus, she wants to send the text messages by using simple words. How can we help help her to send out the text message with using her voice?





# AI emergency text message making process

1. Train the AI model using ML with [Teachable Machine](https://teachablemachine.withgoogle.com/models/llrqwNrQ7/) (e.g., <https://teachablemachine.withgoogle.com/models/llrqwNrQ7/>)
2. Export the model to [BYOTM](https://byotm-282218.ue.r.appspot.com/) (<https://byotm-282218.ue.r.appspot.com/>)
3. Test the model and modify it



# AI emergency text message

- Train the AI model using ML with [Teachable Machine](https://teachablemachine.withgoogle.com/models/llrqwNrQ7/) (e.g., <https://teachablemachine.withgoogle.com/models/llrqwNrQ7/>)
  - 1: Keunjae → Text to Keunjae / 2: Yunha → Text to Yunha

The screenshot displays the Teachable Machine interface. On the left, there are three audio sample upload sections. The first section, labeled 'Background Noise', has 26 audio samples (20 minimum) and shows a waveform. The second section, labeled '1', has 12 audio samples (8 minimum) and shows a waveform. The third section, labeled '2', has 12 audio samples (8 minimum) and shows a waveform. Below these sections is a button labeled 'Add a class'. In the center, a 'Training' panel shows 'Model Trained' and 'Advanced' options. On the right, a 'Preview' panel shows 'Input' (ON), 'Switch Microphone', and 'Overlap Factor' (0.5). Below the preview, an 'Output' section shows a bar chart with three bars: 'Background Noise' at 50%, '1' at 59%, and '2' at 27%.





# AI emergency text message

- Click the Export Model and Upload my cloud model
- Copy the sharable link of your AI model

The screenshot shows the Teachable Machine interface with a modal dialog box open for exporting a model. The dialog has two tabs: 'Tensorflow.js' (selected) and 'Tensorflow Lite'. It offers three options for exporting: 'Upload (shareable link)' (selected), 'Download', and 'Update my cloud model'. A sharable link is displayed: `https://teachablemachine.withgoogle.com/models/11rqwNrQ7/`. Below the link, it states: 'When you upload your model, Teachable Machine hosts it at this link. (FAQ: Who can use my model?)' and 'Your cloud model is up to date.' There are also code snippets for 'Javascript' and 'p5.js' with a 'Contribute on Github' link. The background shows the 'Background Noise' model interface with '26 Audio Samples / 20 m' and '12 Audio Samples / 8 m' sections.





# AI emergency text message

- Go to [BYOTM](https://byotm-282218.ue.r.appspot.com/) (<https://byotm-282218.ue.r.appspot.com/>)

**BYOTM** (Bring Your Own Teachable Machine)

Your Model URL

→

**<Audio Input 1>**  
Your Audio Class name will be populated here when you enter your Model URL. When your model recognizes audio input from this class, it will trigger Output 1, as seen to the right.

→

**Output 1 (Text Message)**  
Recipient Phone Number  
8120000000  
Message  
Hi, Keunjae

**<Audio Input 2>**  
Your Audio Class name will be populated here when you enter your Model URL. When your model recognizes audio input from this class, it will trigger Output 2, as seen to the right.

→

**Output 2 (Text Message)**  
Recipient Phone Number  
8120000000  
Message  
Hi Yunha

START ▶ STOP ■





# AI emergency text message

- Paste the URL to Your Model URL

**BYOTM**

(Bring Your Own [Teachable Machine](#))

Your Model URL

`https://teachablemachin`



1

Your Audio Class name will be populated here when you enter your Model URL. When your model recognizes audio input from this class, it will trigger Output 1, as seen to the right.



2

Your Audio Class name will be populated here when you enter your Model URL. When your model recognizes audio input from this class, it will trigger Output 2, as seen to the right.



## Output 1 (Text Message)

Recipient Phone Number

📞 8120000000

Message

💬 Hi, Keunjae

## Output 2 (Text Message)

Recipient Phone Number

📞 8120000000

Message

💬 Hi Yunha

START ▶

STOP ■



# AI emergency text message

- Write the phone number and text message you want to send

**BYOTM** (Bring Your Own [Teachable Machine](#))

✓  
Your message has been sent!

Your Model URL  
https://teachablemachine

1  
Your Audio Class name will be populated here when you enter your Model URL.  
When your model recognizes audio input from this class, it will trigger Output 1, as seen to the right.

2  
Your Audio Class name will be populated here when you enter your Model URL.  
When your model recognizes audio input from this class, it will trigger Output 2, as seen to the right.

START ▶

STOP ■

**Output 1 (Text Message)**  
Recipient Phone Number  
8128023988  
Message  
Hi, Keunjae

**Output 2 (Text Message)**  
Recipient Phone Number  
8125584441  
Message  
Hi Yunha





# AI emergency text message

- Click the Start button and tell your word to AI emergency text message and check out the results

**BYOTM** (Bring Your Own [Teachable Machine](#))

✓  
Your message has been sent!

Your Model URL  
https://teachablemachine

1  
Your Audio Class name will be populated here when you enter your Model URL. When your model recognizes audio input from this class, it will trigger Output 1, as seen to the right.

2  
Your Audio Class name will be populated here when you enter your Model URL. When your model recognizes audio input from this class, it will trigger Output 2, as seen to the right.

### Output 1 (Text Message)

Recipient Phone Number

8128023988

Message

Hi, Keunjae

### Output 2 (Text Message)

Recipient Phone Number

8125584441

Message

Hi Yunha

START ▶

STOP ■







# Discussion

- Importance of data quality and data quantity
  - Why is it important to have accurate and reliable data?
  - How can we ensure that the data we collect is accurate and reliable?
  - How can poor quality data affect the AI-based app you made?
  - How can we ensure that the data we collect is of high quality? What steps can be taken to reduce errors and inaccuracies in data?
  - Have you ever come across an example where having more data would have helped solve a problem or make a decision? How might more data have been beneficial?
  - Do you think that collecting more data always leads to better decisions? Why or why not?

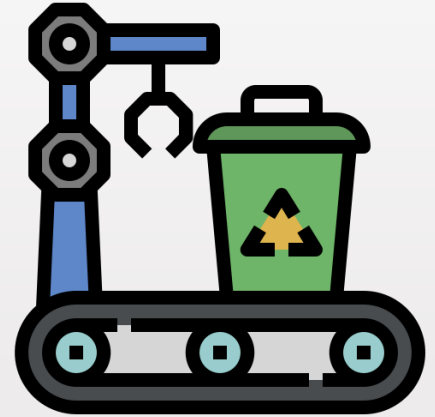


# Hands-on Activity 2

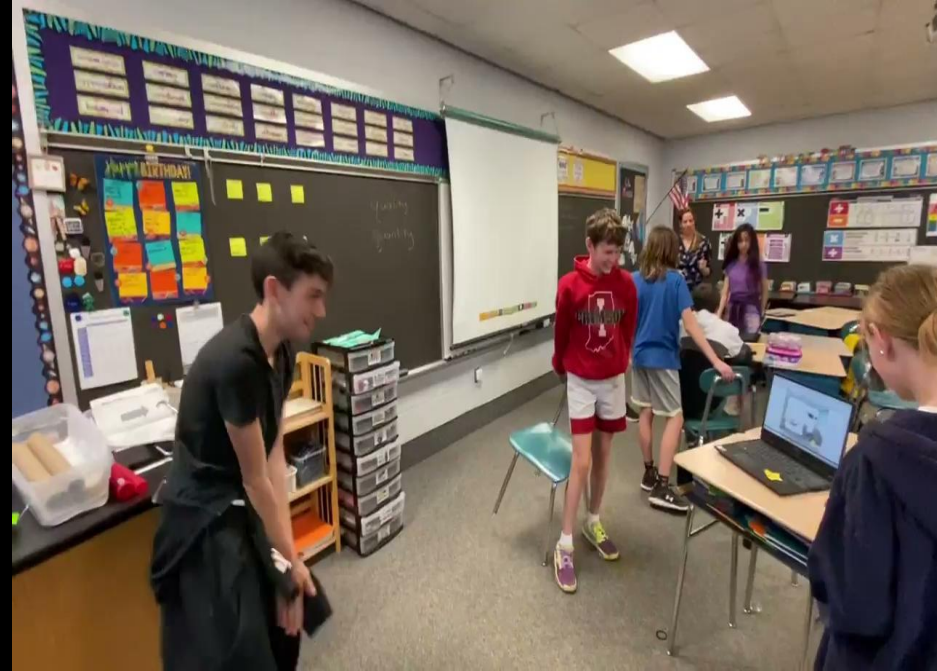
## AI garbage collector

# AI garbage collector

- Problem to solve
  - Jane lives in Bloomington, IN, and she saw that there was a lot of trash in her town but not enough people to clean it up. So, she asked Indiana University to create a robot to collect the garbage, called an "AI garbage collector."
  - However, the robot doesn't know the town's routes, so the university asked Jane to teach the robot by using her body movements to control it.



# AI garbage collector





# AI garbage collector making process

1. Train the AI model using ML with [Teachable Machine](https://teachablemachine.withgoogle.com/models/zZ1ioJ-i3)  
(e.g., <https://teachablemachine.withgoogle.com/models/zZ1ioJ-i3>)
1. [Programming with Scratch](https://playground.raise.mit.edu/create/)  
(<https://playground.raise.mit.edu/create/>)
2. Test the model and modify it





# AI garbage collector (modeling)

- Go to [Teachable Machine](https://teachablemachine.withgoogle.com/) (https://teachablemachine.withgoogle.com/) and choose Pose project

The screenshot shows the 'New Project' page on the Teachable Machine website. At the top left, there is a hamburger menu icon followed by the text 'Teachable Machine'. Below this, the heading 'New Project' is centered. There are two buttons: 'Open an existing project from Drive.' and 'Open an existing project from a file.'. Three project cards are displayed in a row: 'Image Project' (with three small images of a dog), 'Audio Project' (with three small audio waveforms), and 'Pose Project' (with three small images of a person in a blue shirt). Each card has a title and a brief description. At the bottom, there is a dashed box containing the text: 'More coming soon. More models will appear here as they're developed.'





# AI garbage collector (modeling)

- Create 4 labels (up, down, left, and right ) and train the AI model by inputting the data using a webcam

The screenshot displays the Teachable Machine interface. On the left, there are four training cards for labels: 'up', 'down', 'left', and 'right'. Each card has a title, an edit icon, and a section titled 'Add Pose Samples:' with 'Webcam' and 'Upload' buttons. On the right, there is a 'Training' panel with a 'Train Model' button and an 'Advanced' dropdown menu. Next to it is a 'Preview' panel with an 'Export Model' button and a message: 'You must train a model on the left before you can preview it here.' The top left corner shows the 'Teachable Machine' logo. The bottom right corner has a language dropdown set to 'English' and a version number 'release-2.4-5 - 2.4.5.81827c1'.





# AI garbage collector (modeling)

- Click export model and choose upload (sharable link)
- Copy the sharable link (e.g., <https://teachablemachine.withgoogle.com/models/zZ1ioJ-i3/>)

Teachable Machine

up

Add Pose Samples:

Webcam Upload

down

Add Pose Samples:

Webcam Upload

left

Add Pose Samples:

Webcam Upload

right

Add Pose Samples:

Webcam Upload

Export your model to use it in projects.

Tensorflow.js

Export your model:

Upload (sharable link)  Download

Your sharable link:

<https://teachablemachine.withgoogle.com/models/...>

When you upload your model, Teachable Machine hosts it at this link. (FAQ: Who can use my model?)

Code snippets to use your model:

JavaScript [Contribute on Github](#)

Learn more about how to use the code snippet on [github](#).

```

<div>Teachable Machine Pose Model</div>
<button type="button" onclick="init()">Start</button>
<div><canvas id="canvas"></canvas></div>
<div id="label-container"></div>
<script src="https://cdn.jsdelivr.net/npm/@tensorflow/tfjs@1.3.1/dist/tf.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/@teachablemachine/pose@0.8/dist/teachablemachine-pose-main.js"></script>
<script type="text/javascript">
  // More API functions here:
  // https://github.com/google/creative-lab/teachablemachine-community/tree/master/libraries/pose

  // the link to your model provided by Teachable Machine export panel
  const url = "https://teachablemachine.withgoogle.com/models/zZ1ioJ-i3/";
  
```

Export Model

in a model on the left can preview it here.

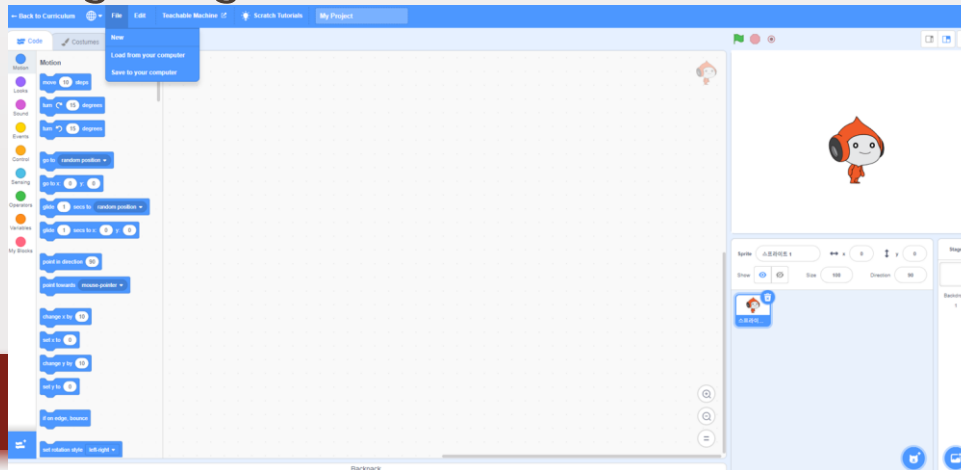






# AI garbage collector (programming)

- Go to [Scratch Playground](https://playground.raise.mit.edu/create/) (https://playground.raise.mit.edu/create/)
- Click the file and choose Load from your computer-  
“aigarbagecollectorrobot2.sb3”





# AI garbage collector (Test & Modify)

The screenshot shows the Scratch AI Lab interface for a project titled "ai garbage collector robot 2". The interface includes a top navigation bar with "Back to Curriculum", "File", "Edit", "Teachable Machine", "Scratch Tutorials", and the project name. Below the navigation bar are tabs for "Code", "Costumes", and "Sounds".

The left sidebar shows the "Motion" category selected, with various blocks like "move 10 steps", "turn 15 degrees", "go to random position", etc. The main workspace contains a script starting with a "when clicked" event block, followed by a "use model" block with the URL <https://teachablemachine.withgoogle.com/models/z21ioJ43/>. A red circle highlights this "use model" block and a yellow tooltip box above it that reads: "First, train either an Image or Pose model at Teachable Machine. Include the 4 classes: up, down, left, right. Then, insert your model here, and try the game!".

Below the "use model" block are four "when model detects" blocks, each with a "point in direction" block: "up" (90), "down" (180), "right" (90), and "left" (-90). The right sidebar shows the "Video Sensing" section with a "model prediction" dropdown set to "right". Below this is a video feed of a person with a green robot sprite overlaid on their face. The "Score" is 0. The "Sprite" section shows "airobot" selected, with "garbage" and "eraser" also visible. The "Sprite" properties are set to x: 0, y: -50, Size: 100, and Direction: 90.





# Discussion

- Importance of data quality and data quantity
  - Why is it important to have accurate and reliable data?
  - How can poor quality data affect the AI-based app you made?
  - How can we ensure that the data we collect is accurate and reliable?
  - How can we ensure that the data we collect is of high quality? What steps can be taken to reduce errors and inaccuracies in data?
  - Have you ever come across an example where having more data would have helped solve a problem or make a decision? How might more data have been beneficial?
  - Do you think that collecting more data always leads to better decisions? Why or why not?



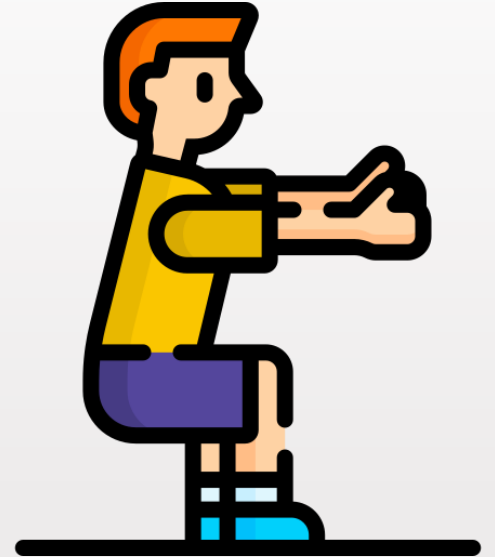
# Hands-on Activity 3

## AI Squat machine

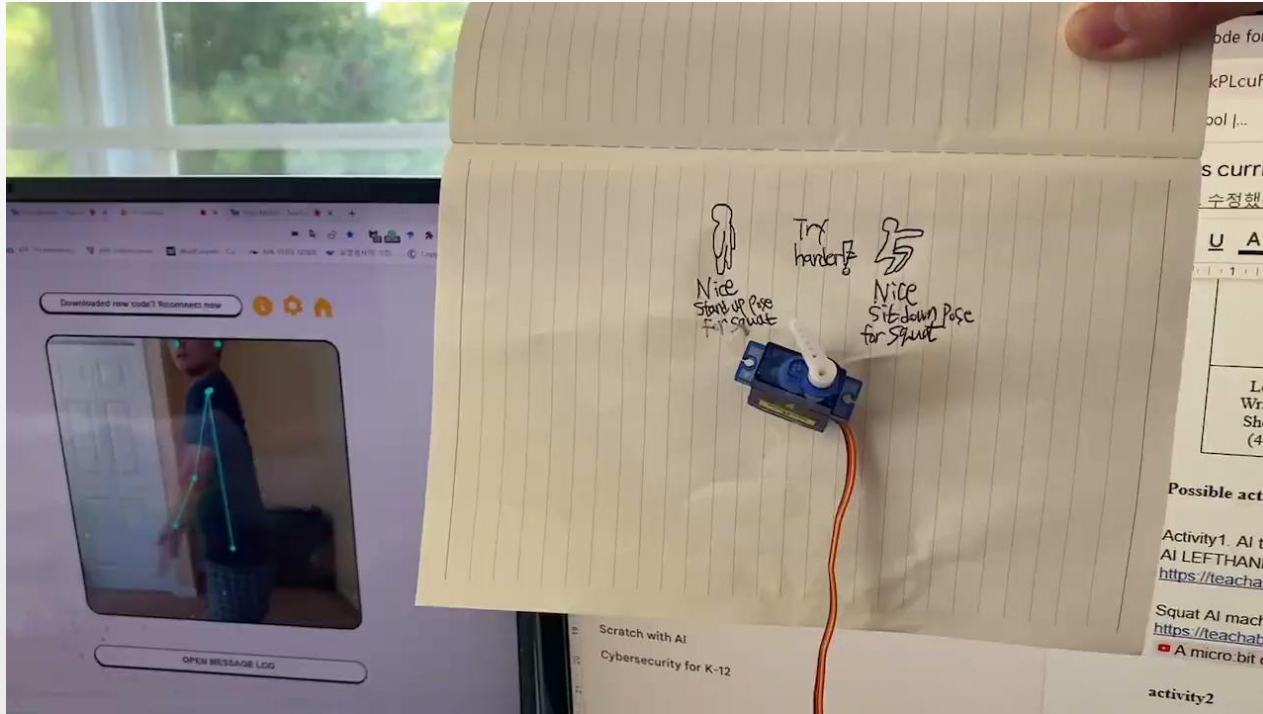
The logo features a stylized red circuit board with lines extending upwards and outwards, resembling a brain or neural network.

# AI Squat Machine

- Problem to solve
  - Due to COVID-19 restrictions, many students in Indiana are unable to exercise regularly. As a result, home workouts such as squats, push-ups, and sit-ups have become more popular. However, a significant number of students are not familiar with how to properly perform these exercises, particularly squats.
  - What measures can be taken to help students exercise correctly and with proper form?



# AI Squat Machine





# AI Squat Machine making process

1. Train the AI model using ML with Teachable Machine (<https://teachablemachine.withgoogle.com/models/ltiv7fcu3/>)
2. Create the Squat machine with Microbit and servomotor
3. Programming with Makecode (<https://makecode.microbit.org/>) and connect the Microbit
4. Upload the AI model to AI training (<https://ai-training.glitch.me/>) and sync with the Microbit
5. Test the model and modify it





# AI Squat Machine (Modeling)

- Go to Teachable Machine  
(<https://teachablemachine.withgoogle.com/>) and choose Image project

The screenshot shows the Teachable Machine web interface. At the top left, there is a blue button with a hamburger menu icon and the text "Teachable Machine". Below this is a "New Project" section. There are two buttons: "Open an existing project from Drive." and "Open an existing project from a file.". Below these are three project cards: "Image Project" (with three images of a dog), "Audio Project" (with three audio waveforms), and "Pose Project" (with three images of a person squatting). Each card has a brief description of how to teach the model. At the bottom, there is a dashed box containing the text "More coming soon" and "More models will appear here as they're developed."







# AI Squat Machine (Modeling)

- Create 2 labels (1: Sit-down pose, 2: Stand-up pose, or vice versa ) and train the AI model by inputting the data using a webcam. The label's name should be a number instead of using

The screenshot displays the Teachable Machine web interface. At the top left, there is a hamburger menu icon and the text "Teachable Machine". The main workspace contains two class creation cards, labeled "1" and "2". Each card has a title bar with a pencil icon and a vertical ellipsis, followed by the text "Add Image Samples:". Below this text are two buttons: "Webcam" (with a camera icon) and "Upload" (with an upload icon). At the bottom of the workspace is a dashed box labeled "Add a class". To the right of the class cards is a "Training" panel with a "Train Model" button and a dropdown menu currently set to "Advanced". Further right is a "Preview" panel with an "Export Model" button and a message: "You must train a model on the left before you can preview it here." At the bottom right of the interface, there is a language dropdown set to "English" and a version number "release-2.4.6 - 2.4.618d7c1".





# AI Squat Machine (Modeling)

- Click export model and choose upload (sharable link)
- Copy the sharable link

(e.g., <https://teachablemachine.withgoogle.com/models/ltiv7fcu3/>)

Teachable Machine

up

Add Pose Samples:

Webcam Upload

down

Add Pose Samples:

Webcam Upload

left

Add Pose Samples:

Webcam Upload

right

Add Pose Samples:

Webcam Upload

Export Model

Export your model to use it in projects.

Tensorflow.js

Export your model:

Upload (sharable link)  Download

Your sharable link:

<https://teachablemachine.withgoogle.com/models/ltiv7fcu3/>

When you upload your model, Teachable Machine hosts it at this link. (FAQ: Who can use my model?)

Code snippets to use your model:

[JavaScript](#) [Contribute on GitHub](#)

Learn more about how to use the code snippet on [GitHub](#).

```
<div>Teachable Machine Pose Model</div>
<button type="button" onclick="start()">Start</button>
<div>Canvas</div>
<div id="label-container"></div>
<script src="https://cdn.jsdelivr.net/npm/@tensorflow/tfjs@1.3.1/dist/tf.min.js"></script>
<script src="https://cdn.jsdelivr.net/npm/@teachablemachine/pose@0.8/dist/teachablemachine-pose.min.js"></script>
<script type="text/javascript">
// Here are TensorFlow.js docs:
// https://github.com/googlecreativelab/teachablemachine-community/tree/master/libraries/pose

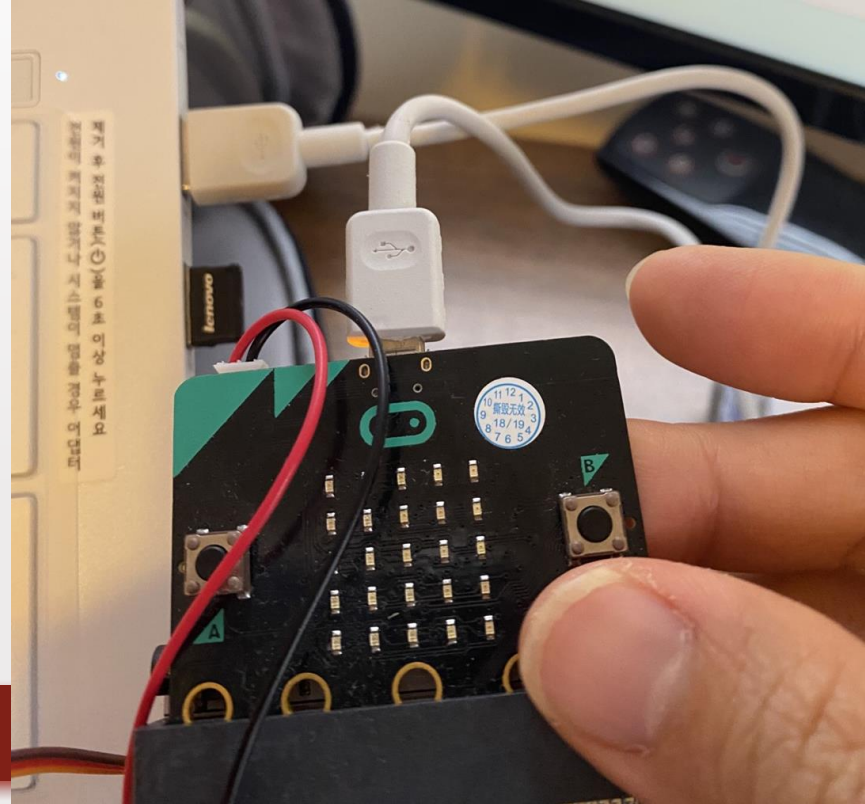
// the link to your model provided by Teachable Machine export panel
const URL = "https://teachablemachine.withgoogle.com/models/ltiv7fcu3/";
const modelURL = URL + "model.json";
const dataURL = URL + "data.json";
const model = tf.loadModel(modelURL);
const data = tf.loadDataset(dataURL);
const canvas = document.getElementById("canvas");
const startButton = document.getElementById("start");
const labelContainer = document.getElementById("label-container");
let modelReady = false;
let classes = [];
let classIndex = 0;
startButton.addEventListener("click", async () => {
  if (!modelReady) {
    model = await model.load();
    classes = await data.loadClasses();
    modelReady = true;
  }
  const prediction = model.predictFromImage(canvas);
  const classIndex = prediction.classIndex;
  labelContainer.textContent = classes[classIndex];
});
// End of TensorFlow.js code
</script>
```





# AI Squat Machine (making)

- Connect the Microbit with the laptop using USB cable





# AI Squat Machine (making)

- Go to Makecode (<https://makecode.microbit.org/>)
- Click the setting-connect device button and connect with Microbit

```
on start
  serial
  redirect to
  TX USB_TX
  RX USB_RX
  at baud rate 9600

Click on the...

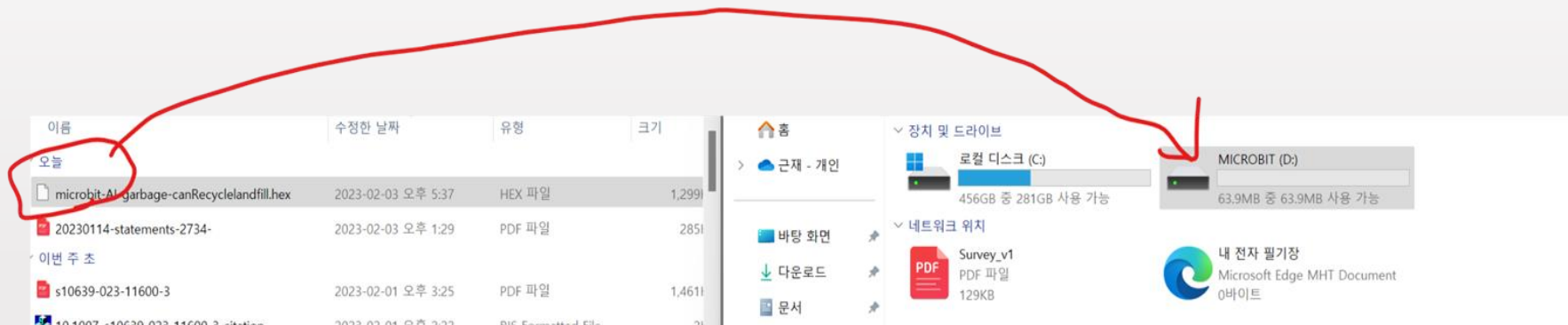
forever
  set SerialData to serial read string
  if SerialData = 1 then
    continuous servo P1 run at 100 %
    pause (ms) 1000
    stop servo P1
  else if SerialData = 2 then
    continuous servo P1 run at -100 %
    pause (ms) 1000
    stop servo P1
  else
```





# AI Squat Machine (making)

- Make sure to download the hex file. Drag and drop the hex file into Microbit (D:) folder.





# AI Squat Machine (Test & Modify)

- Go to AI training (<https://ai-training.glitch.me/>) and click “Pair Microbit” to sync with the Microbit
- Paste the trained AI model from the Teachable Machine and click “Ready!”

**Not sure what to do?**

Visit the Teachable Machine to train an AI project:

[Google Teachable Machine](#)

New to Google Teachable Machine? Follow these guidelines to learn more: [Guide to Teachable Machine](#)

**When you have finished training your model, follow these 4 simple steps:**

1. Click “Export Model”

Paste your Google Teachable machine model link here:

Choose Camera:

Choose Audio:

[Ready!](#)





# Discussion

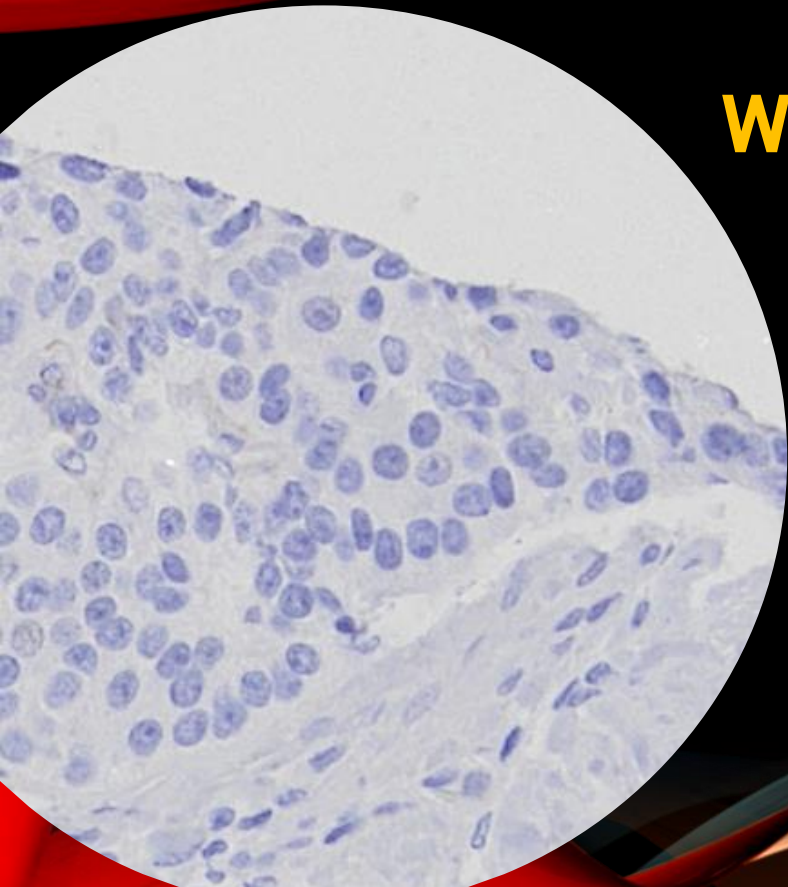
- Importance of data quality and data quantity
  - How did you fix the problems encountered while developing AI-based artifacts?
  - Why is it important to have accurate and reliable data?
  - How can poor quality data affect the AI-based app you made?
  - Please leave your thought on the google jamboard (<https://jamboard.google.com/d/1gu2qSoTgSPZVc3JN6vm7E9xPodCx1tZ7zG08dmnSqbm/edit?usp=sharing>)



# Invited Talk

**Dr. Xuhong Zhang**





# When AI Meets Healthcare

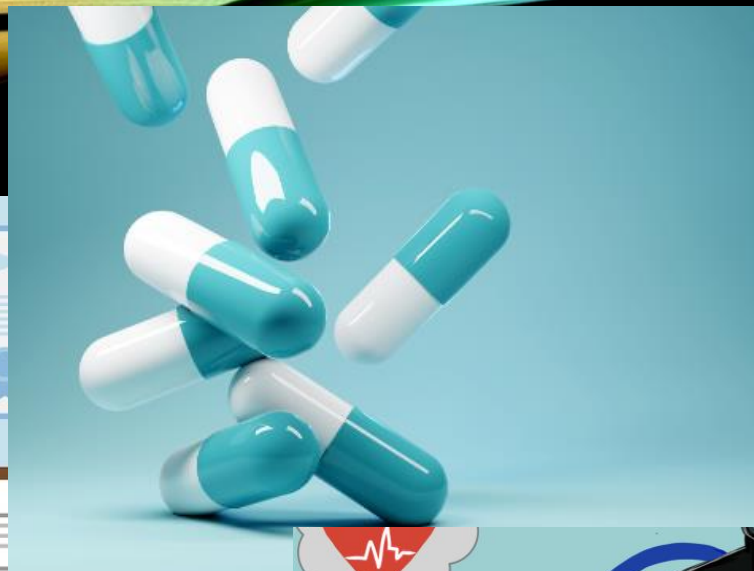
*AI Goes Rural*  
*AI4Me Summer Camp*  
June, 5th, 2023



INDIANA UNIVERSITY

## AI & HEALTH CARE

- When we talk about health care, what comes to your mind?





## **AI & HEALTH CARE**

- **How the AI and Health care can be related?**

# AI & HEALTH CARE

- Basically speaking, AI can help us making decisions.
- In Health Care, what decision we need to make?



## **AI & HEALTH CARE**

- **Biomedical Imaging Computing: Make Sense from Biomedical Image Pixels**

# AI HISTORY FOR BIOMEDICAL IMAGING



~201

Early methods<sup>3</sup> mainly use basic image processing techniques, but they usually produce inferior performance compared to machine-learning based approaches<sup>1</sup>.



2013~201

Other supervised machine learning methods<sup>7</sup> have been applied: SVM, linear classifier, Bayes classifier, etc.

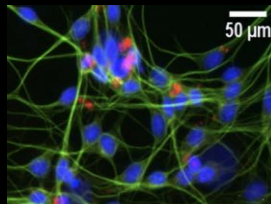
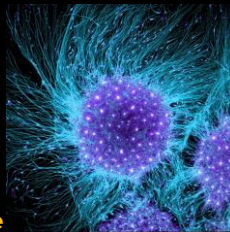


2017~presen

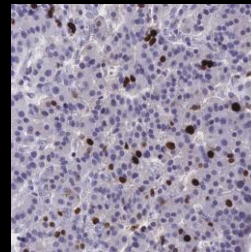
Convolutional Neural Network<sup>†</sup>

# Biomedical Imaging

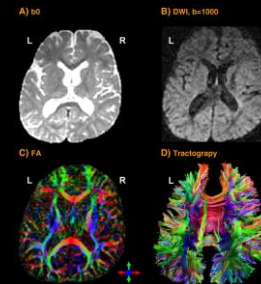
Overlay of betaIII-tubulin neuron stained image and DAPI stained cell nuclei image (1)



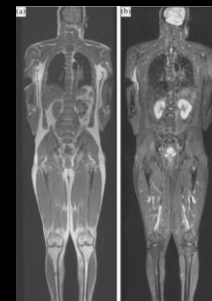
Immunohistochemical (IHC) Stained tissue microarray image



Human neural rosette primordial brain cells (2)

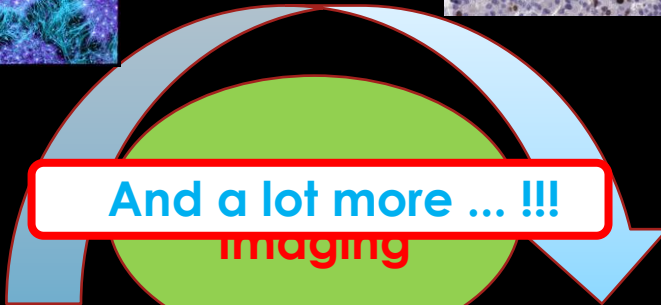
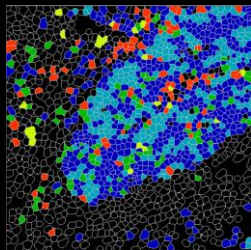


Primary data containing a T2 (baseline b0) image, diffusion weighted MRI (3)



Whole body MRI on T1SE(a) and STIR(b) (4)

Images analyzed with HALO<sup>®</sup> software using dsDNA for nuclear segmentation



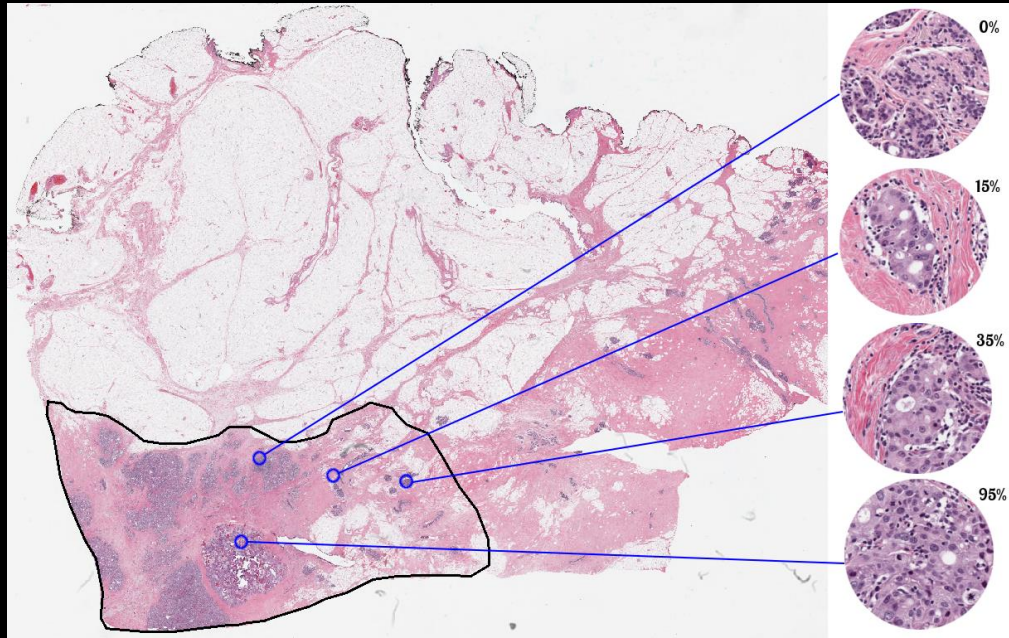
**Micro-level**

**Macro-level**

Image Credit:  
(1) J Zhang, E Moradi, MG Somekh, ML Mather. Label-Free, High Resolution, Multi-Modal Light Microscopy for Discrimination of Live Stem Cell Differentiation Status. *Scientific Reports*. (2018)8:697  
(2) GF. Croft, L Pietila, S Tse, S Galgoczi, M Fenner, AH, Brivanlou. *A small world*. The Rockefeller University, Brivanlou Lab.  
(3) ZA Englander and et al., Diffuse reduction of white matter connectivity in cerebral palsy with specific vulnerability of long Range fiber tracts. *NeuroImage: Clinical* 2 (2013) 440-447



# RECOGNIZE CELLS



Breast cancer: find regions with many tumor cells

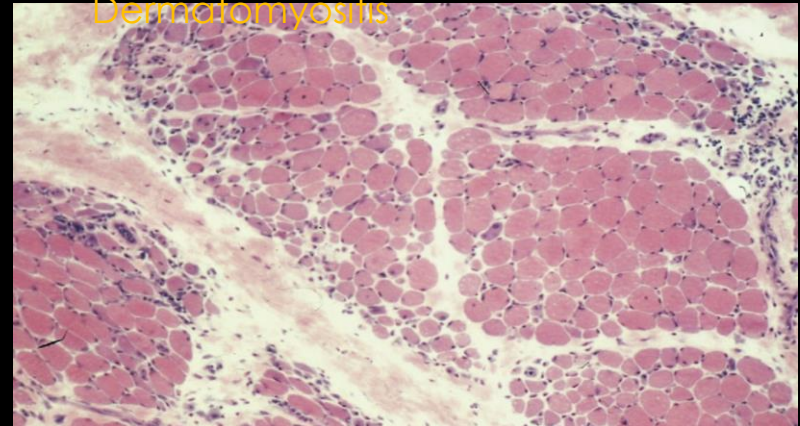
# Disease Classification

- Myositis: 5 ~ 10 people per million.
- Symptoms:
  - Proximal muscle weakness
  - Inflammation
  - Gottron's rash (pictured, A)
  - Calcification (B & C)
- H&E stained biopsies
- Types of myositis:
  - Dermatomyositis (DM)
  - Polymyositis (PM)



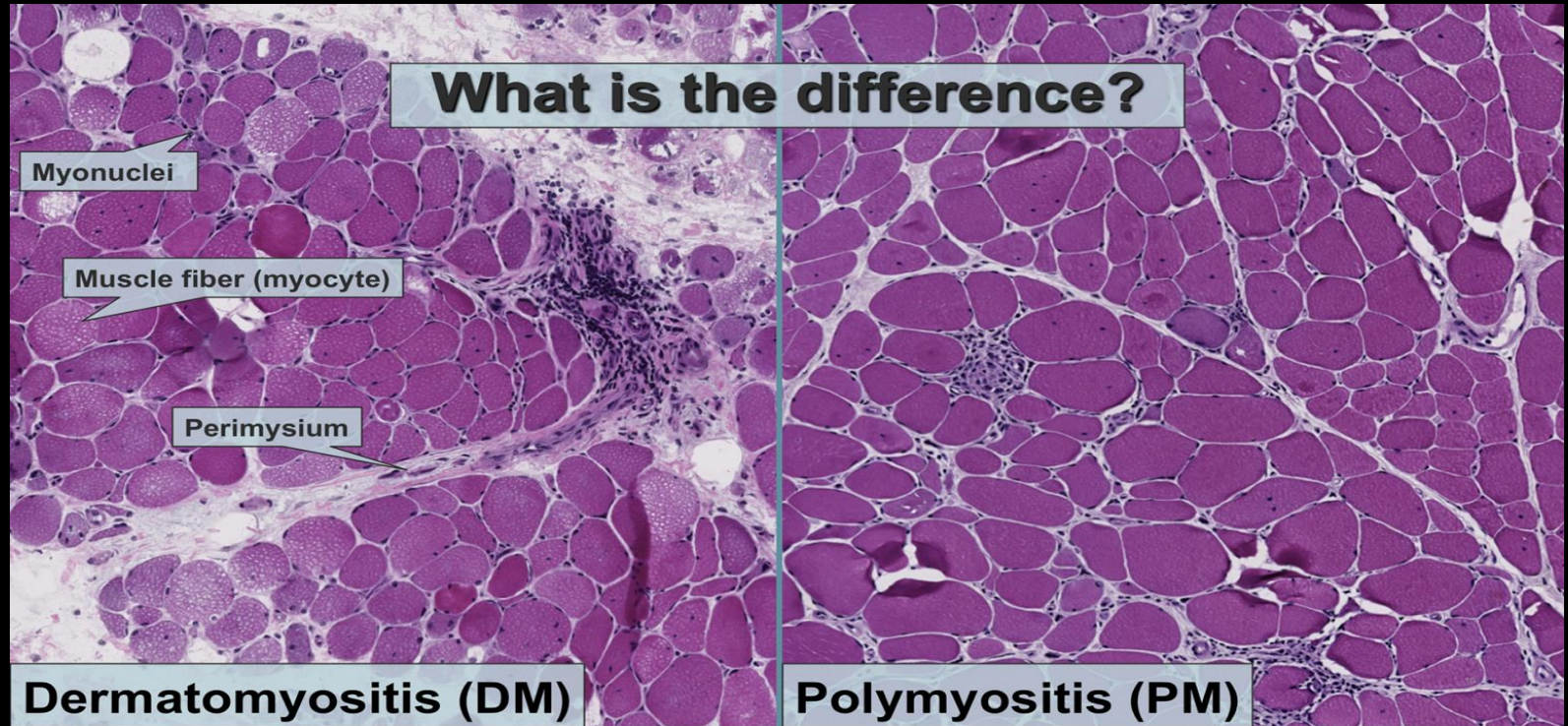
Figure 1: **Rash and calcifications in dermatomyositis**  
A: Gottron's rash. B: Skin effects of calcification. C: Radiographic evidence of calcification.

## Perifascicular Atrophy in Dermatomyositis



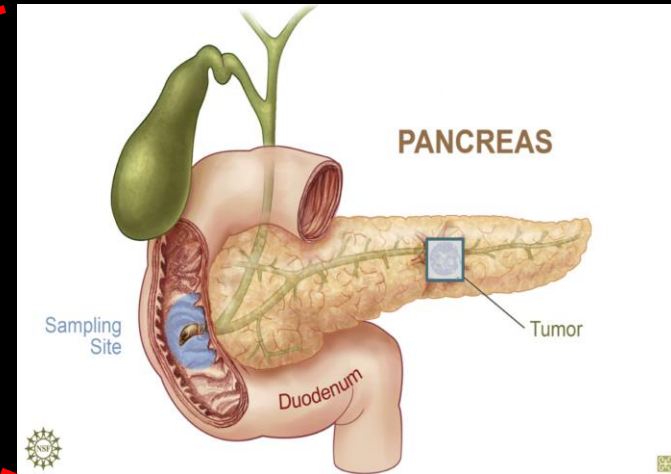
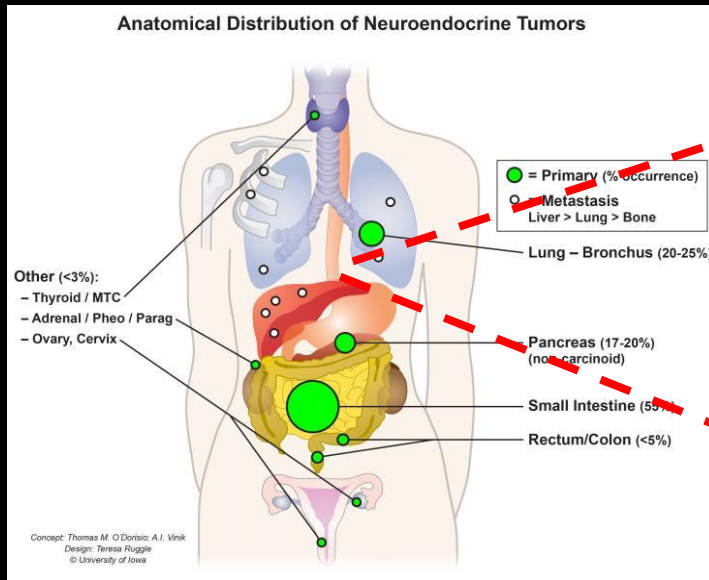
Marinos C. Dalakas and Reinhard Hohlfeld, "Polymyositis and dermatomyositis",  
*The Lancet*, vol. 362, no. 9388, pp. 971-982, 2003.

# DISEASE CLASSIFICATION



# DISEASE GRADING

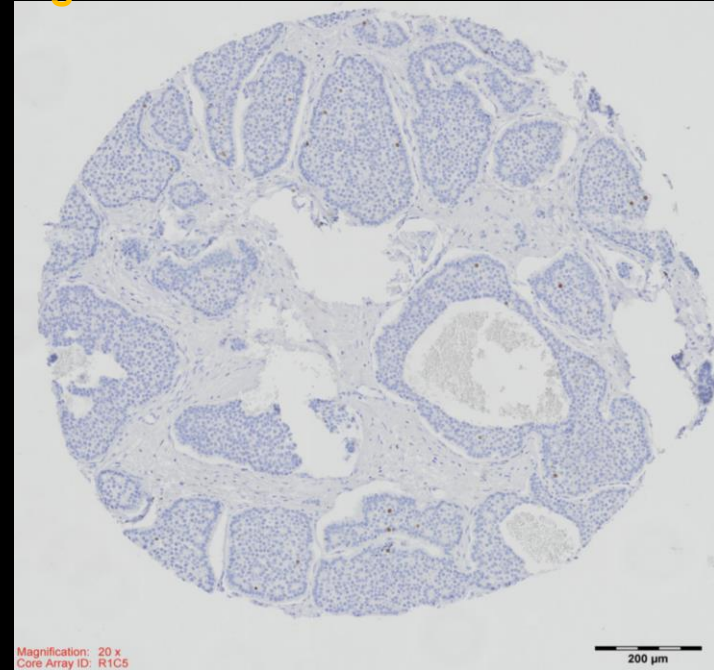
## Neuroendocrine Tumors



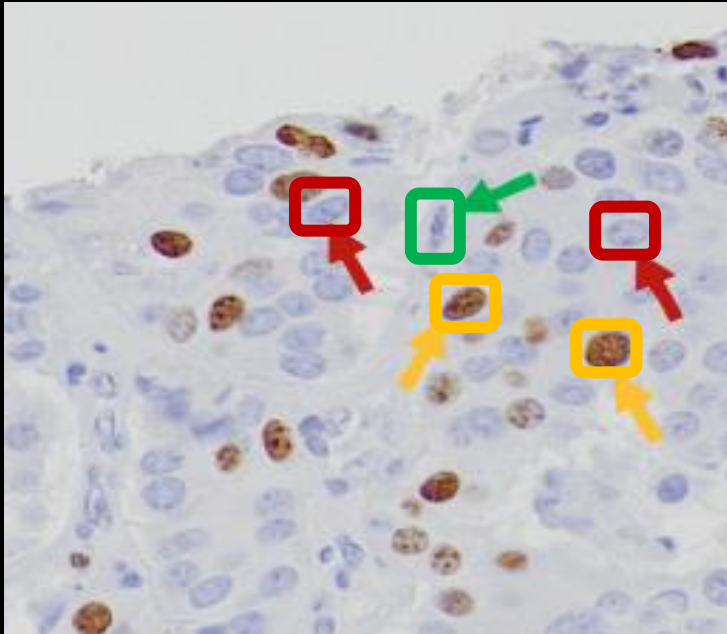
# DISEASE GRADING




- Neuroendocrine Tumors (NETs)
  - 12,000 new diagnoses in the US per year.<sup>1</sup>
- Tumor progress
  - Low: <3%
  - Intermediate: 3%~20%
  - High: >20%

Pancreatic tissue microarray (TMA) image

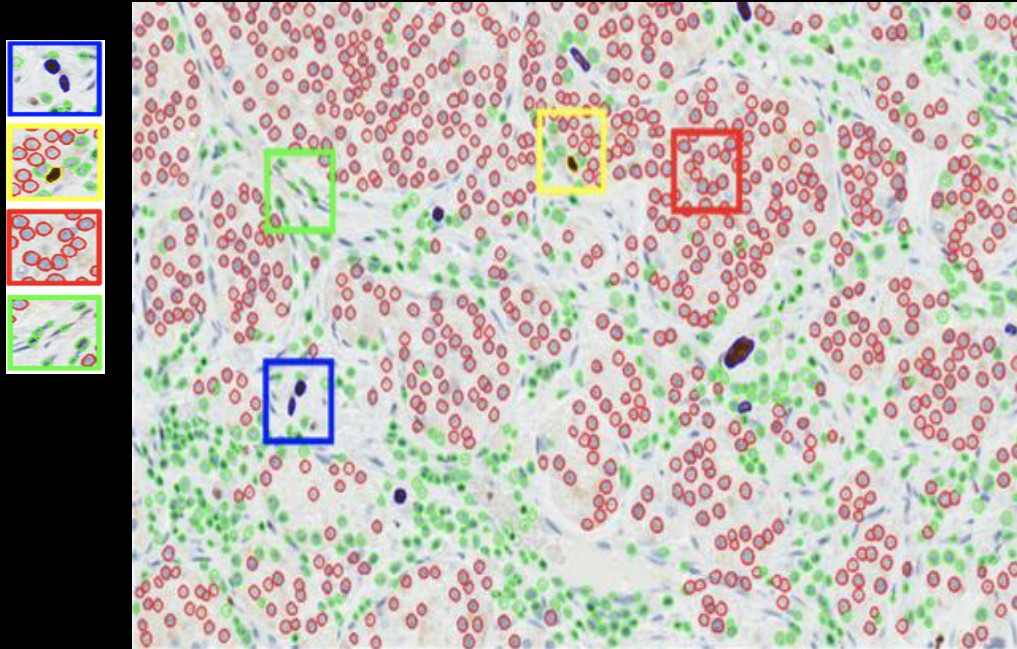


# DISEASE GRADING



-  Immunopositive tumor nuclei
-  Immunonegative tumor nuclei
-  Non-tumor nuclei

# DISEASE GRADING



Non- tumor cells (non-lymphocytes),  
Immunopositive tumor,  
Immunonegative tumor,  
lymphocytes



# Collecting Data

- [https://docs.google.com/forms/d/1f6Mtw\\_AEeidFahMWrZCnZpY90D0H1uNp8mOjfAfR-44/edit](https://docs.google.com/forms/d/1f6Mtw_AEeidFahMWrZCnZpY90D0H1uNp8mOjfAfR-44/edit)

