Actua's AI Activities Series

Activity 6

What Machines See: Digging Into Machine Vision

actüa
Youth · STEM · Innovation
Jeunesse · STIM · Innovation

What Machines See: Digging Into Machine Vision

If you're accessing this activity directly, did you know there are eight other activities in this series up on our website? If you find yourself unfamiliar with any of the AI concepts and terminology introduced in these activities, please refer to our [AI Glossary](#). These activities also follow a space exploration narrative when done in order. It is recommended to complete the activities in order but they can also be done on their own.

You and your group mates are astronauts and scientists aboard the Actua Orbital Station. Unfortunately, your station just got bombarded by magnetic rays and your electronics have begun to shut down! The only one who can save you is the orbital station's AI, DANN. DANN stands for Dedicated Actua Neural Network, and it's gone a little loopy. Brush up on your technical skills, learn about AI, and save yourself and your crewmates!

Our access to DANN's audio core is almost complete. In [“Hand Commands: Training Image Classification Models”](#), you trained a model that can recognize hand shapes so that you can give instructions to DANN to continue with repairs. Now we need to look deeper, though. How does that model work? How can the model tell if you're making one hand shape or another? The instructional model to give DANN instructions isn't working as well as it should. What can we do to fix it? Let's find out so we can access DANN's audio core and fix it in [“Voice Activated AI: Training Audio Recognition Models”](#)!

Activity Summary

In this activity, participants will learn about machine vision, comparing and contrasting how humans and computers see. Participants will also learn about how computers understand what they see. They will also learn about applying different strategies for data collection that will help make machine vision models more accurate.

Developed by Actua, 2022.

Delivery Environment	Activity Duration	Intended Audience
Classroom with computer access	60-75 minutes	Grades 9-12 (Ages 13-18)

Achievement Goals

Learning Goals

Learning goals are statements referring to the understanding, knowledge, skills or application participants acquire during the activity. **Following this activity, participants will:**

- **Identify** differences between human and machine vision
- **Implement** machine learning training strategies
- **Explore** the implications of computers with realistic human senses

Logistics (Timing, Group Size, Materials)

Section Title	Time	Group Size	Materials
Opening Hook: How and what does a machine really see?	10 minutes	*Entire group*	
Activity 1: Finding features	20 minutes	*Individually or in small groups*	**Group Size** - Print out of Appendix A.1 (Finding Features) - Pen, pencil, or highlighter - Computer with webcam - Internet access

Section Title	Time	Group Size	Materials
			• Interactive sketch: <u>2D Visualizer</u>
Activity 2: Training a better classifier	40 minutes	*Individually or in small groups*	**Group Size** • Computer with webcam and internet access • Teachable Machine
Reflection & Debrief	10 minutes	*Entire group*	

Safety Considerations

Safety considerations have been provided below to support safety during this activity, however they are not necessarily comprehensive. It is important that you review the activity and your delivery environment to determine any additional safety considerations that you should be implementing for the delivery of these activities.

Online Safety

Some components of this activity require the use of devices connected to the internet.

- Facilitators should review the provided videos and read/explore provided websites and materials to determine if they are suitable for their participants.
- Where applicable, facilitators should remind participants to stay on task and only use links provided within this activity.

Activity Procedure

The AI model for DANN's visual core was good enough to continue repairs, but it's not quite as good as it used to be. Mission Control has asked if you can improve it so that future teams will be able to use it. This means we need to take a closer look at some of Teachable Machine's features to see if we can use them to enhance the performance of the model. First though, you need to figure out exactly how your model is making its guesses.

Opening Hook: How does a machine really see?

To understand how computers see, we first need to think about how humans see, and how we make sense of what we see.

1. Write or draw what you think happens for humans to see. *Hint: Think about the organs and systems involved in human sight.*
   a. Light comes through the lens in our eyes and hits our retinas on the back wall of our eyes.
   b. Light sensitive cells (photoreceptors), called cones and rods, in our retinas respond to the light that reaches them, which generates a signal.
2. Write or draw how you think humans can **understand** what we see. *Hint: Think about what organs and systems might be involved in understanding visual information.*
   a. The signal from our retinas is carried along the optic nerve to the brain.
   b. The signal is interpreted by different parts of the brain for different purposes, including face and object recognition and understanding position and movement.
   c. We rely on our memory and experience to understand what we're seeing and figure out what new things are.

3. **Computer vision used by AI happens in a similar way to human vision, though there are some key differences.** Write or draw what you think needs to happen for a computer to "see". *Hint: Think about tools, inventions, or technologies that can act in a similar way as the organs discussed above.*

   a. Computers need some sort of sensor, such as a webcam, to see the world around them.

   b. Light enters the webcam through its lens and lands on an image sensor.

   c. The image sensor interprets the light it receives into an image, and then the webcam sends the image to the computer.

   d. Computers can also load photos or videos from data files. Image files (such as a JPEG, GIF, or PNG) or video files (such as an MP4, WEBM, MOV, or AVI) contain image data (gathered by a sensor) and can be read by a computer.

4. Write or draw how you think an AI model can **understand** what it sees.

   a. The image data from a webcam or data file is fed into a trained AI model.

   b. Inside the AI model, the image data is processed in different ways to figure out what information it might contain. Those bits of information are called features and this process is called feature detection.

   c. Based on the presence or absence of certain features, the AI model will make a guess as to what it thinks it is looking at.

Compare human vision and computer vision: How are they the same? How do they differ?

5. What advantages do humans have over computers when it comes to visually understanding the world?

   a. Human visual understanding of our world is incredibly complex and built up over a lifetime of experience.

   b. We are able to recognize the same object in different lighting, in different locations in our field of vision, at different distances, sizes, and angles, and in different scenes or environments—and usually not just

one difference, but a combination of multiple differences. Any of these differences might fool an AI.

    c. Our brains are also very fast at object recognition. We are able to quickly glance around our environment and understand the different objects we see.

    d. We also have strategies for figuring out things we don't know: when we see an object we don't recognize, we can compare it to objects of similar shape, size, or colour. This feature comparison is something that AI can do, but we do it much faster.

    e. We could also gather more data on objects by tilting our heads to look at them from a different orientation or walking around them to see different angles. Computers are fixed to the perspective we give them of the object.

6. What advantages might computers have over humans when it comes to visually understanding the world?

    a. A computer can look at a large number of different images a lot faster than a human can.

    b. Computers can have access to different kinds of sensors outside the visible light spectrum. For example, some computers have ultraviolet or infrared sensors, allowing them to pick up light that humans can't detect.

    c. Computers might be able to detect smaller details or changes in an image than a human could.

Section 1: Finding features

Consider the above illustration of the American Sign Language letter A.

1. Imagine that you have to describe this letter to someone without them seeing it. How would you describe it? What are important words that you will need to use to accurately describe this hand shape? *Write down your answers.*
   a. When/If asked, share your answers. Did everyone come up with the same words? Did anyone describe the hand shape in terms of wrist, finger, or thumb position? Did anyone use terms like "closed fist" or "curled fingers"?
2. Now, imagine that you have to describe this letter to someone, **but you can only describe it in terms of lines and edges.** To help you visualize what this might look like, for each of the images below, think about what each would look like if only they only showed the feature described.
   a. If you have a printed version of this, highlight or shade (with a highlighter, marker, pen, or colour pencil) the described features.

Left edges	Right edges	Horizontal lines	Outside edges
Left and right edges	Outside and horizontal edges		

3. Can you think of other, similar features? Write down two of your ideas or fill in and highlight them in the spaces provided.

   a. When/if asked, share the features that you came up with. Did everyone come up with the same features?

4. The image that we've looked at is a line drawing with no background. How would this feature detection work if we added colour? How would this feature detection change if there was a background behind the image.

5. To get a sense of what machine vision looks like, explore the interactive sketch below.

   a. Interactive sketch: Machine vision vs human vision - 2D Visualizer

6. Once the sketch is loaded (the screen will show an image of your webcam), press the "Space" key to activate machine vision.

7. In this mode, your computer is using a filter to highlight certain features throughout the image.

8. Press the arrow keys to change to different filters. What features do you think are being highlighted by each filter? *Hint: Switch between the filters to see how the light areas and lines change*.
   a. These particular filters should be highlighting edges. Each filter should highlight the same edge as the direction of the arrow key pressed.
9. There are 4 filters in this sketch, but an image classification model uses many more (hundreds or more) filters (and thus, features) to understand an image.
10. Switch between human (non-filtered) and machine (filtered) vision. What observations can you make about how they're different? What about similarities?

Section 2: Training a Better Classifier

Because AI models use features like edges and lines to make their guesses, we can improve the accuracy of our models by making it clear what features are relevant and what aren't. To this end, you will be exploring two strategies to improve the accuracy of your image classification model:

1. Removing unnecessary information from your training data.
2. Improving the diversity of your training data.

Task 1: Removing unnecessary information from your training data

For this strategy, you will make use of Teachable Machine's crop tool when creating your training datasets. The crop tool lets you sample a smaller region of the webcam input instead of using the whole video. Before you begin, { individually / in small groups / as a whole class } consider the following questions:

1. In your training data, what might be considered unnecessary information? What might be considered necessary information?
2. How can unnecessary information affect your model's accuracy?
3. Now, create a new/empty Teachable Machine image classification project.

4. Follow the same instructions you used for setting up your classes in "Hand Commands", the previous activity. Use the same class names that were used in the training of your previous model (i.e. DISENGAGE, DISMOUNT, SHUTDOWN, REINITIALIZE, and UNKNOWN).

5. For creating your training data this time, the process will be changed a little:

    a. Click on "Webcam" to open the webcam capture tool.

    b. In the upper-left corner of the webcam image, you should see the crop tool button. When you hover your mouse over it, it should say "Crop". Click the crop tool button.

    c. You should now see a box drawn on top of the webcam image. You can click-and-drag on the corners of this box to resize it. Resize it such that it only captures your hand shape and nothing else. When you are done resizing it, click the "Done cropping" button.

    d. Once you click "Done cropping", the webcam image preview should only show what you included in the crop tool box.

    e. Resume the process to create new training data using the cropped output of your webcam. Do this for each of your classes.

6. Re-train your model by clicking the "Train Model" button in the middle "Training" box.

7. When training is complete for your model, the preview box on the right should become active.

8. For your model to work, you will have to crop the webcam image in your preview box as well, to match how you cropped the others.

    a. Use the crop tool button in the upper left of the webcam window to select a smaller region of the webcam image. Click the crop tool button once more to confirm your crop.

    b. The webcam image should update to show the region that you selected with the crop tool.

9. Proceed with testing your model like you did in "Hand Commands".

If you encounter any difficulties, make sure that you don't have any old training images in your set (i.e. you either created a new project or you deleted all of your previous training images).

Task 2: Improving the diversity of your training data

This strategy helps improve classification accuracy by providing more examples of your classes for your model to learn from. Before you begin, consider the following questions:

1. What do you think "diversity" means in terms of training data? What are some examples of diversity?
2. Why is it important to have lots of different examples of a class in your training data?

This strategy builds on the Teachable Machine project from the first strategy.

Remember: image classification models don't think about objects or images in the same terms that we do. Instead, they establish what an object "looks like" based on the examples you provide in your training data. This means that they won't recognize variations of the same object if they haven't already seen examples of those variations in their training data.

For this activity, you will create and then execute a plan for adding diversity to your training data.

1. First, think about the differences between environments that you're usually in. What are some important differences between your living room, your bedroom, and a classroom? What about between a classroom and the school library? What are possible differences that can happen within the same room? Think about light levels, background colours, objects in the area, and other differences like those.
2. How would these differences affect what a webcam sees? Write down your ideas about these differences.
3. Now, think about how the people that were part of creating the training data (Yourself, any groupmates) might also be able to provide variations on the classes. For example, are you wearing different coloured clothes than last time? Do you and your groupmates have slightly different ways of making the same hand shapes?

4. How would these differences show up in your training data, even after reducing unnecessary information? Write down your ideas about these differences.

5. Now, your goal is to create a table that organizes this information so that you can track the different ways that you've diversified your training data. Then, capture training data for as many of these differences as possible. Use the template provided below as a starting point.

Personal/Group member conditions	Environmental conditions			
	Classroom, lights on	Classroom, lights off	Outside, overcast day	...
Sam, red sweater				
Ali, t-shirt				
Chris, bracelet				
...				

6. As you gather training data in each combination of conditions, place a checkmark in the corresponding box of your table so that you know that you've covered it.

7. Once you've collected as much training data as you can, click on the "Train Model" (or "Model Trained", if you've already trained at least once) to start the training process again.

8. After your model finishes training, test your model just like before. Don't forget to crop the webcam image in the preview window if you need to.

Reflection & Debrief

Creating good training data can be a time-consuming and resource-intensive process. You might notice that as you add more data, model training also starts to take longer. After completing this activity, use the following questions as a starting point for discussing the experience:

1. What are some of the biggest challenges for creating diverse training data? Think about your experience with this image classification model, but also broader applications of AI like self-driving cars, virtual assistants, or voice recognition interfaces.

2. Did your model(s) from this activity perform better than your previously trained models? What are some ways that you can compare models?

3. Thinking about the evaluation questions from the "Hand Commands" activity (printed below), did your model(s) perform better at any specific evaluation task? How do these evaluation questions help to address problems with data?

   a. Does your model accurately classify [...] hand shapes...

      i. ...when made by you or other group members used in the training data?

      ii. ...when made by other people who were not part of the training data?

      iii. ...when applied in a different environment from where the training data was generated (e.g. a different part of the room, a different background for the webcam, different levels of light)?

Delivery Adaptations

How might you adapt the time, space, materials, group sizes, or instructions to make this activity more approachable or more challenging? **Modifications** are ways to make the activity more accessible, **extensions** are ways to make the activity last longer or more challenging.

Modifications

- Activity 1 (Finding Features) can be done as a large group to save time. In this case, you would work collaboratively to complete each step and collect answers on the board or on chart paper. You would need one computer with a webcam to use the "Human vs. Machine" interactive sketch.
- The images in Activity 1 could be projected onto a white or blackboard and then coloured on top of, so that when you turn off the projector, the features would be left displayed on the board.

Extensions

- Between activities, you can provide more information about how AI models use features to make their guesses, using the appendix "Making sense of data: artificial neural networks" (Appendix A.1).
- The process in Activity 1 of describing an image conceptually then using only specific vocabulary can be repeated with a more complex image of your choosing.

References & Gratitude

Google (n.d.). Crop [Icon]. Material design.
   https://material.io/resources/icons/?icon=crop&style=baseline

TensorFlow (n.d.). Anatomy of a Neural Network [Image].
   https://www.tensorflow.org/about

Terms of Use

Prior to using this activity or parts thereof, you agree and understand that:

- It is your responsibility to review all aspects of this activity and ensure safety measures are in place for the protection of all involved parties.
- Any safety precautions contained in the "Safety Considerations" section of this write-up are not intended as a complete list or to replace your own safety review process.
- Actua shall not be responsible or liable for any damage that may occur due to your use of this content.
- This work is licensed under a Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International License. For more information, please see https://creativecommons.org/licenses/by-nc-sa/4.0/.
- You may adapt the content for your program (remix, transform, and build upon the material), providing appropriate credit to Actua and indicating if changes were made. No sharing of content with third parties without written permission from Actua.

About Actua

Actua is Canada's leading science, technology, engineering and mathematics (STEM) youth outreach network, representing a growing network of over 40 universities and colleges across the country. Each year 350,000 young Canadians in over 500 communities nationwide are inspired through hands-on educational workshops, camps and community outreach initiatives. Actua focuses on the engagement of underrepresented youth through specialized programs for Indigenous youth, girls and young women, at-risk youth and youth living in Northern and remote communities. For more information, please visit us online at www.actua.ca and on social media: Twitter, Facebook, Instagram and YouTube!

Appendices

Appendix A: Career & Mentor Connections

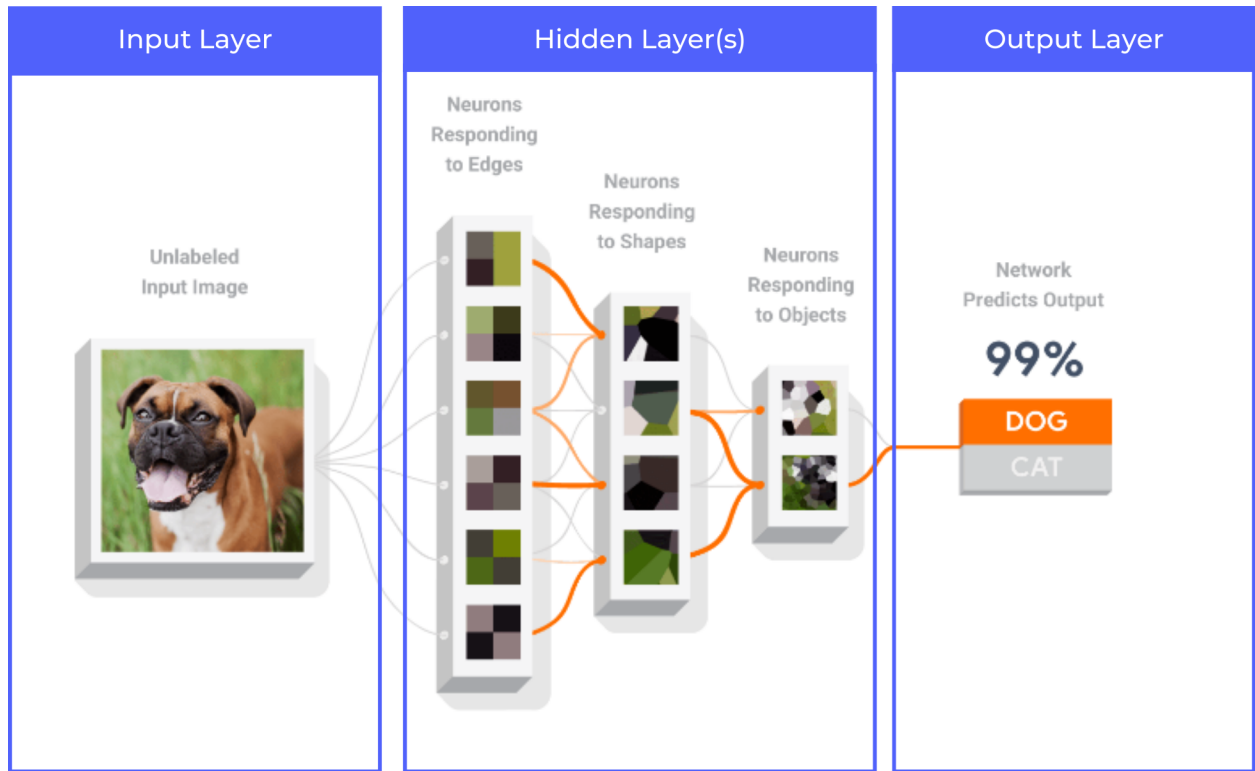- Machine learning researcher
- Programmer
- Software Engineer

Appendix B: Background Information

Appendix A.1: Making sense of data: artificial neural networks

The AI model that you trained in Teachable Machine uses an artificial neural network, sometimes just called a "neural network", to analyse and classify the images that it sees. A neural network is composed of:

1. An input layer, where the input data is introduced into the neural network. In the case of an image classification model such as Teachable Machine, the input data is made up of the data from each of the pixels of the image being analysed.
2. A number of hidden layers, where the data is manipulated to look for certain features and groups of features.
3. An output layer, where the results of the last hidden layer are used to make a guess, in the case of a classification model, as to what class is likely to have been represented by the input data.
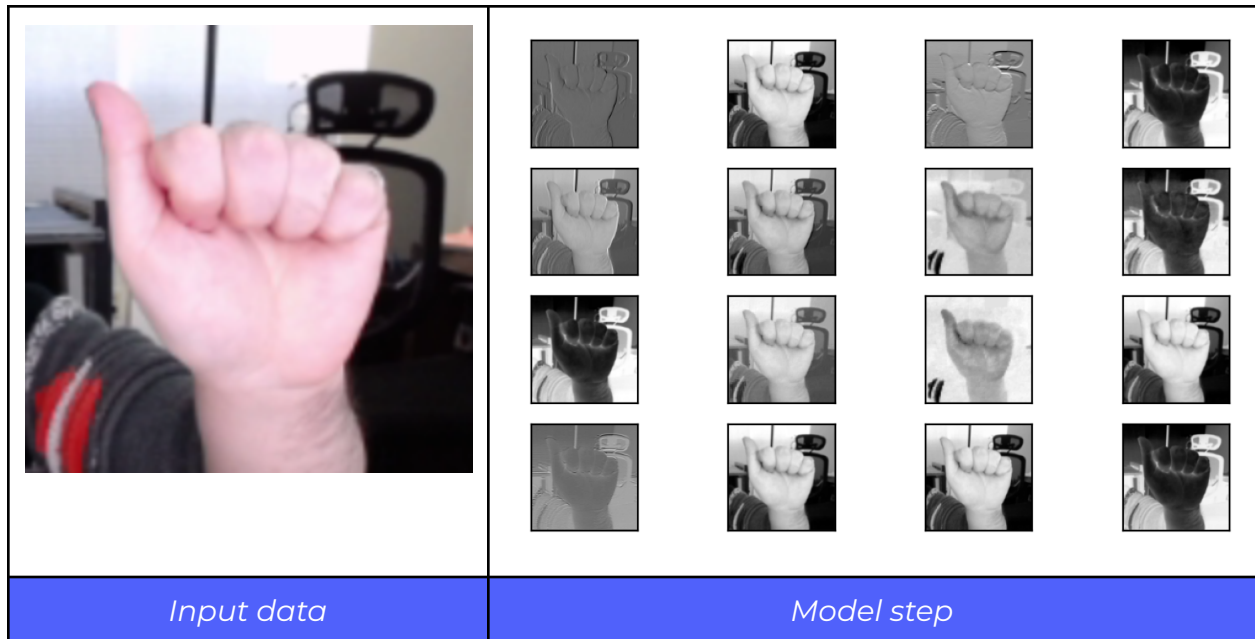
| Input Layer | Hidden Layer(s) | Output Layer |

Neurons Responding to Edges

Neurons Responding to Shapes

Neurons Responding to Objects

Unlabeled Input Image

Network Predicts Output

99%

DOG

CAT

Adapted from TensorFlow. (n.d.). Anatomy of a Neural Network.
https://www.tensorflow.org/about

The challenge is that AI models don't see "hands" in terms of fingers, thumbs, wrists, or arms. Instead, they learn by looking for patterns in your training images. These patterns likely aren't meaningful to humans. They don't necessarily correspond to a specific feature or set of features. For example, consider the output of the first hidden layer, below:

Input data	Model step

Most of the output images look different, but it's hard to describe what, exactly, each one is showing, or how it relates to the eventual classification of this input image as American Sign Language "A". This is just the first of many hidden layers: Teachable Machine's model uses more than 28 hidden layers to arrive at its guess.

Appendix C: Supporting Resources

Finding features printout *(see below)*

Left edges	*Right edges*	*Horizontal lines*	*Outside edges*
Left and right edges	*Outside and horizontal edges*		

Model evaluation questions *(see below)*

Does your model accurately classify, for each of your command classes, your specified hand shapes...

1. ...when made by you or other group members used in the training data?
2. ...when made by other people who were not part of the training data?
3. ...when applied in a different environment from where the training data was generated (e.g. a different part of the room, a different background for the webcam, different levels of light)?

If yes to all of the above, your model is ready for use. If not, consult the section on troubleshooting, below:

1. Does your training data include any images that are not good representations of the hand shape for the class that they are in?
2. Does your model work on some command classes but have difficulty recognizing specific command classes? Looking at the hand shape specification and training data, could you hypothesize why this might be?

Training data planning table printout *(see below)*

Personal/Group member conditions	Environmental conditions			