SCRATCH LESSON PLAN

Exploring AI with Scratch Face Sensing Blocks

With Scratch Face Sensing blocks, you can create interactive projects that use the features of your face! Our AI-powered extension uses a machine learning model to detect if it sees a face and where a nose, eyes, ears, mouth, etc., are. Use the blocks to control game characters with a face tilt, draw with your nose, create face filters, interact with informational projects, and more! You can also try to fool the AI to test the limitations and potential for false negatives or false positives, and reflect on design considerations, like when these blocks can enhance the user experience. The blocks do not collect or store personal information, like recordings or images of your face, making it a safe, fun, and creative way to explore the possibilities of AI.

> **“With these new Scratch [blocks], young people can use AI in the process of creating their own projects — thus expanding the range of what they can create while also enabling them to learn about AI in [a] meaningful and motivating context.”**
>
> — Professor Mitch Resnick, co-founder of Scratch

As we explore the use of AI as a creative tool, the Scratch Foundation has developed “Guiding Stars” and “Lines in the Sand.” Our goal is to ensure that our use of AI is not just ethical and responsible, but also true to our creative learning principles. As we think about incorporating AI as one tool in the toolbelt, we believe in centering learner agency (allowing them to opt in or out of AI use) and exploring how AI could empower a user’s self-expression and add to, not replace, their creativity.

Guiding Stars	***Lines in the Sand***
● **Creativity**: Empower self-expression	● **Safe**: Prevent harmful content
● **Agency**: Keep power and choice in the user’s hands	● **Ethical**: Protect user’s data
● **Equity**: Make it free, fair, inclusive, and accessible	● **Transparent**: Share information about how it works
● **Community**: Emphasize human connection and collaboration	● **Human-Centered**: Never undermine human agency or relationships

Per Scratch's Terms of Service, students of any age may directly interact with/code with the Face Sensing blocks. **When you use Face Sensing blocks, only your computer can sense your face. None of your data is stored or sent to Scratch or any other site.** The extension can detect that a face exists, but cannot identify the person.

Audience: Classroom Teachers, Instructional Technology Specialists, Library Media Specialists, Informal Learning Environments

Time: Approx 1–2 hours total

- Part 1: What Is AI?
- Part 2: Create a Face Sensing Project - 30-60 min
- Part 3: Can You Fool the AI? - 30 min
- Part 4: Reflect and Share - 30 min

Resources for Learners:

- Learning Library: Face Sensing Resources (includes coding cards, a video tutorial, starter projects, and more)
- Interactive Tutorial | Face Sensing
- Coding Cards | Face Sensing (student-facing)
- Worksheet | Face Sensing Project Sharing
- Starter Project | Face Sensing - Face Filter
- Starter Project | Face Sensing Game
- Starter Project | Face Sensing - Sound Board
- Studio | Face Sensing Example Projects
- Video Tutorial | Exploring AI with Scratch Face Sensing Blocks
- Guide | Scratch Creative Learning Philosophy

Additional resources provided throughout the guide.

Objectives (Learners Will):

- Gain hands-on experience using predictive AI via Scratch Face Sensing blocks, exploring new perspectives and approaches that build upon their original ideas
- Decide whether to use AI-powered blocks based on the nature of the project's goals and their intended audience
- Reflect on ways AI can be integrated into activities in support of creative learning, to empower self-expression and add to, not replace, a learner's creativity
- Test, debug, and reflect on the possibilities and pitfalls of AI (including false negatives and false positives)
- Communicate and share their projects with their learning community

See page 12 for aligned standards.

Note: *At this time/at launch, Face Sensing blocks are only available in the Scratch web application and in English. However, our team is working to get the Face Sensing blocks translated, and we also hope to include this extension in the downloadable/offline version of Scratch in the future.*

Part 1: What Is AI?

Introductory Material for Educators

- "AI and Creative Learning: Concerns, Opportunities, and Choices" and "In the age of AI, we need a human-centered society more than ever" – blog posts by Mitch Resnick, Professor at MIT and co-founder of Scratch
- If you'd like to learn more about Google's AI Face Detection software that powers the Face Sensing extension, you can explore @tensorflow-models/face-detection and see details in their Short Range model card, such as the model's limitations, ethical considerations, and fairness evaluation metrics, including skin tone bias.
- Algorithmic Justice League – website that includes "Coded Bias: A Documentary" and more related to face detection

Key Concepts

Artificial intelligence (AI) is the umbrella term for computing systems that use data in order to analyze information, make predictions, and generate creative ideas.

Predictive AI is a branch of AI that uses machine learning to **classify data** it receives in order to **generate predictions.** The more sophisticated data we feed in, the more sophisticated and accurate the system will be. It **learns patterns and relationships in data.** Examples could include:

- Facial detection (Scratch's Face Sensing extension)
- Product recommendations (Facebook advertisements)
- Email automation and spam filtering (language classification)

Generative AI is a branch of AI that uses machine learning to **analyze user prompts and create new content** such as text or images based on its best predictions of what the user wants. Examples could include:

- ChatGPT – an AI chatbot that uses "natural language processing" to create human-like dialogue and can respond to questions and compose written content
- Dall-E – a system that enables users to create new images with text to graphics prompts, and develops visual predictions based on data it has been trained on

Part 2: Create a Face Sensing Project

Explore the Face Sensing Blocks

Step 1: Add the Face Sensing extension blocks by clicking on the extension menu in the lower-left corner of the project editor. Face Sensing blocks use the device's camera. If you haven't enabled the camera in your browser, you may be prompted to give permission.

Step 2: You can use the default sprite or choose another from the library.

Step 3: Click the "go to nose" block while your face is visible on the stage. Did the sprite go to your nose? What if you choose another feature?

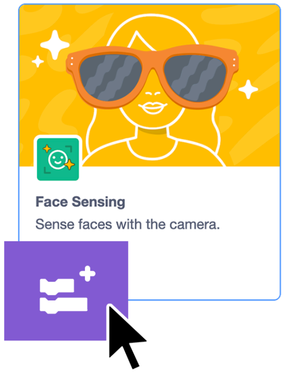
Step 4: How can you make the sprite constantly follow your nose? (Encourage them to find a block to help, like a forever loop.)

Step 5: How can you utilize additional blocks (like "turn clockwise 15 degrees" from the Motion category) to add more animation?

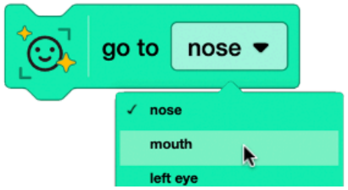
The machine learning model used for these blocks was trained on millions of photos of people's faces. This is known as a data set. The blocks detect things based on what the model has already seen (like parts of a face and how they are arranged), looking for patterns like: a nose is between the eyes and mouth, eyes are typically a certain size in relation to the face, mouths may show teeth or not, etc. **They can detect that a face exists, but cannot identify the person. And only your computer can sense your face. None of your data is stored or sent to Scratch or any other site.**

Resources:

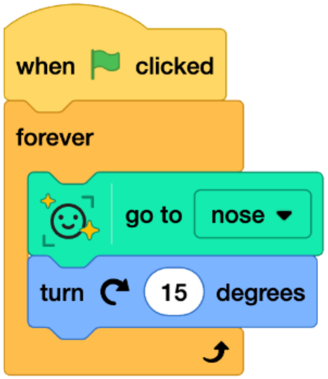
- [Guide | Scratch Creative Learning Philosophy]() (tips for designing creative learning experiences and facilitating creative coding)



Add the Face Sensing extension.



Try out a block, like "go to nose."



Try adding a forever loop and additional motion.

Option 1: Hat and Glasses Face Filter

Step 1: Select a “Fashion” accessory sprite from the sprite library, or draw your own. In the costume Paint Editor, try drawing a hat using brush or shape tools. Our resources on Sprite Creation (like coding cards and videos) contain helpful details on using the Paint Editor tools.

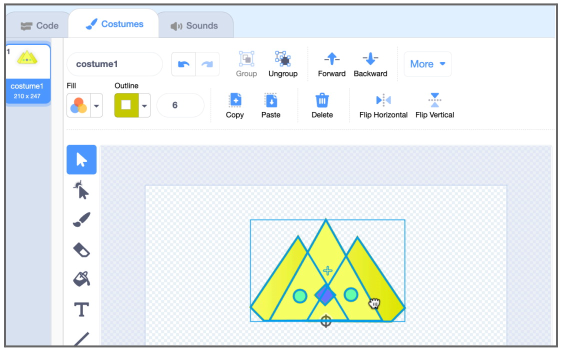
Step 2: Add Face Sensing blocks so the sprite sticks to the top of your head.

Step 3: Is the placement of the hat just right on your face? Experiment with the placement of the costume in the Paint Editor. For instance, can you see the difference if a hat costume is centered versus when only one edge is on the centerpoint? See our video tutorial “Scratching the Surface: Costume Center” for more information.

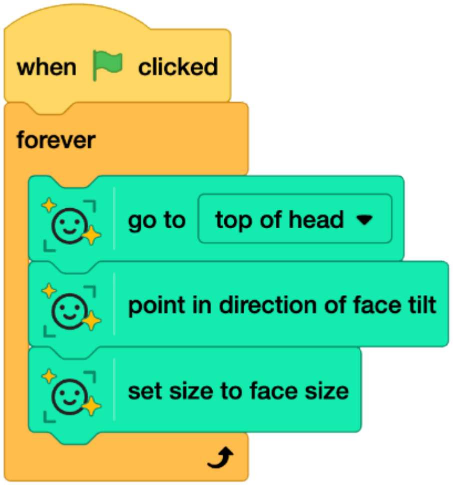
Step 4: How does the sprite appear on your face when you are close to the camera and further from the camera? Do you see any blocks you could use so it scales to match the size of your face?

Step 5: Does the sprite match the direction of your face tilt? Do you see blocks to match it?

Step 6: Add another sprite, like glasses or earrings. Stick it between your eyes or to your ear. What else can you try? Have you created multiple costumes for a sprite? Add code to switch.



Creating a sprite in the Scratch Paint Editor.



```
when ⚑ clicked
forever
    go to [top of head ▾]
    point in direction of face tilt
    set size to face size
```

Adding blocks to control direction and size.

Resources:

- Interactive Tutorial | Face Sensing
- Starter Project | Face Sensing - Face Filter
- Video Tutorial | Exploring AI with Scratch Face Sensing Blocks
- Learning Library: Sprite Creation Resources
- Learning Library: Face Sensing Resources (includes coding cards)

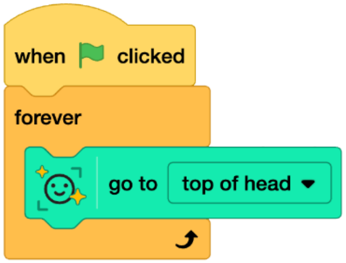
Option 2: Create a Face Sensing Game

Step 1: Select two sprites from the sprite library.

Step 2: Add code to the first sprite so you can control it with your face. This will be the player.

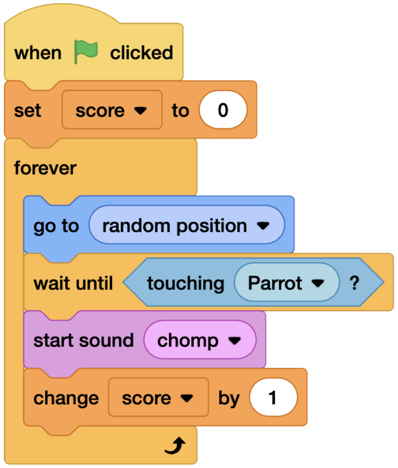
Step 3: Add code to the second sprite so it moves to a random position on the stage when touched by the player sprite.

Step 4: Create a score variable to track points. Give the player a point when the sprites touch Don't forget the program will need to reset the score each time a new game is started.



```
when 🏳 clicked
forever
    go to (top of head ▾)
↩
```

Example code for the player sprite.

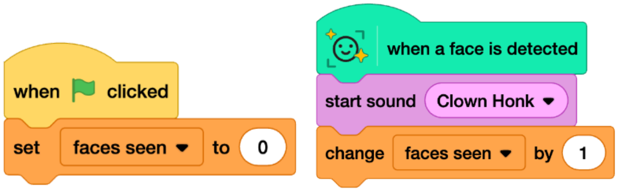


```
when 🏳 clicked
set (score ▾) to (0)
forever
    go to (random position ▾)
    wait until <touching (Parrot ▾) ?>
    start sound (chomp ▾)
    change (score ▾) by (1)
↩
```

Example code for the second sprite.

Experiment with different game options:

- Try making this game harder by moving the second sprite after a time, if not touched yet.
- Create a face detection counter using "a face is detected" blocks, and move your face on and off the stage. Does it detect you each time? Try covering parts of your face, moving far away or close up…or see what other face-like objects it detects.
- Remix another game like a clicker game into a Face Sensing project (for example, use "when this sprite touches a nose" versus "when this sprite clicked"). Move or change the direction of a sprite with face tilt.



```
when 🏳 clicked
set (faces seen ▾) to (0)

when a face is detected
start sound (Clown Honk ▾)
change (faces seen ▾) by (1)
```

Example using "when a face is detected."

Resources:

- Starter Project | Face Sensing Game
- Learning Library: Face Sensing Resources (includes coding cards)
- Learning Library: Clicker Game Resources

Option 3: Get Creative and Get Moving!

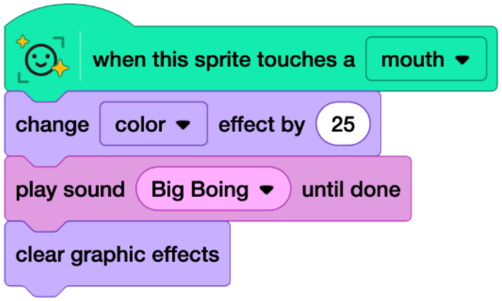
Look at the three Face Sensing hat blocks (a block with a rounded top that starts a script) or the Face Sensing reporter blocks (these can fit inside another block). Can you make an interactive project using some of these?

Options could include:

- a sound board, where sounds are triggered when a button touches a feature of your face, or a DJ game where face tilts change the beat and ears activate note sequences
- an informational project that plays a sound, changes a background, or displays information when you touch a sprite
- a quiz game where your mouth selects the right answer
- a drawing program that allows users to draw with their nose
- a sprite stamping project where you use your eye to place a sprite on the stage and your head tilt to stamp it or change the costume

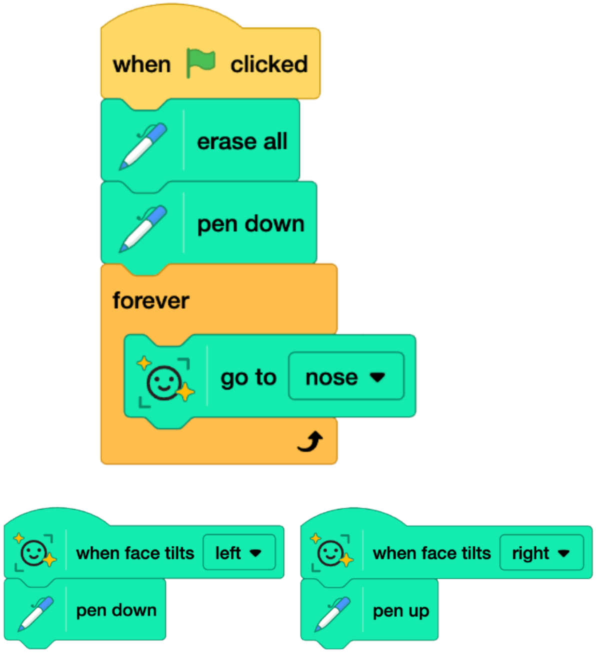
See our student-facing coding cards and studio with examples for additional ideas.

You can combine blocks from the Face Sensing extension with blocks from other extensions, like the Music, Pen, and Text to Speech extensions.



```
when this sprite touches a [mouth ▾]
change [color ▾] effect by (25)
play sound [Big Boing ▾] until done
clear graphic effects
```

Example of sound board code.



```
when ⚑ clicked
erase all
pen down
forever
    go to [nose ▾]
↻
```

```
when face tilts [left ▾]
pen down
```

```
when face tilts [right ▾]
pen up
```

Example using "when face tilts" in a drawing program.

Resources:

- Learning Library: Face Sensing Resources (includes coding cards)
- Starter Project | Face Sensing - Sound Board
- Example Project | Face Sensing: Pen Drawing with Your Nose
- Example Project | Face Sensing: Vietnamese Food Quiz
- Studio | Face Sensing Example Projects
- Video Tutorial | Exploring AI with Scratch Face Sensing Blocks

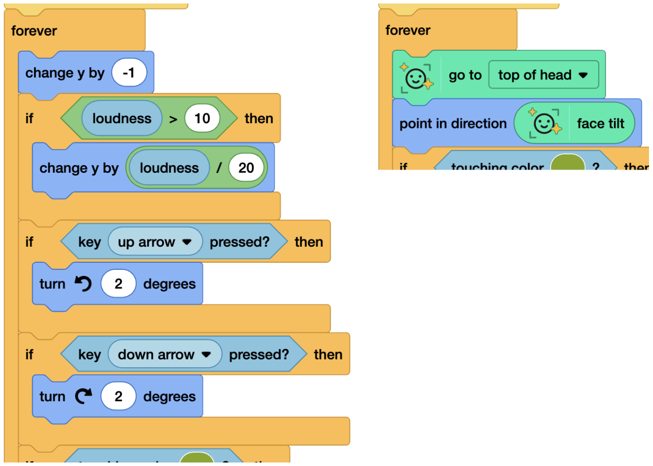
Design Considerations

Many projects could work using a variety of inputs, like:

- keyboard
- mouse
- sound/loudness
- external button (like a Makey Makey or micro: bit)
- Face Sensing blocks
- video motion

In pairs, as a class, or individually in a design journal, reflect on the potential pros and cons of using different inputs for your project.

- Who is your audience/who are you designing for? Do they have any accessibility concerns (like limited hand dexterity)?
- Are there benefits to having multiple input options or choosing one option over another? Does one input give the user a different level of control over sprites than another? What is your project's goal?
- How can adding movement benefit your audience, engaging both their brain and body?
- What additional challenges or debugging opportunities arise while working with different inputs? (For example, Face Sensing and video motion could be tricky when using a Sensing block like "touching color," since you don't know what color may be in the camera background.)



```
forever
    change y by -1
    if loudness > 10 then
        change y by loudness / 20
    if key up arrow pressed? then
        turn ↺ 2 degrees
    if key down arrow pressed? then
        turn ↻ 2 degrees
```

```
forever
    go to top of head
    point in direction face tilt
    if touching color ? ther
```

Example of using "loudness" and arrows versus "top of head" and face tilt to control a sprite.

Example Projects:

- *Fishy Fishy 217332929 vs 554771225 (using go to mouse pointer instead of go to nose)*
- *Vietnamese Boat Float 1216389964 vs 1060227699 (using loudness to control the movement of the boat and up and down arrows instead of face tilt)*
- *Vietnamese Food Quiz 1216390004 vs 1060254080 (using left arrow key to select the food costume and the mouse to drag the food to the cat)*
- *Vietnamese Drum Bounce 1216390066 vs 1060254615 (using left and right arrow keys to make the drum move; or see inside for notes about attempts at using video sensing instead)*

Part 3: Can You Fool the AI?

Identifying False Positives and False Negatives

You can learn more about the model used for the Face Sensing extension, by exploring their model card (model limitations, ethical considerations, and fairness evaluation metrics).

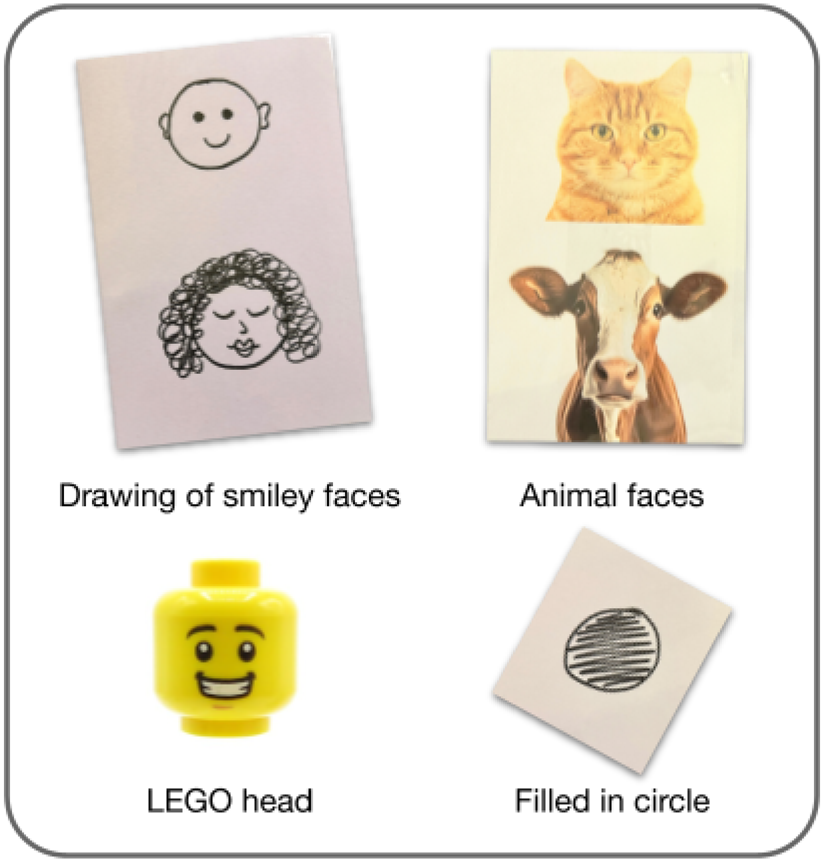
False Positives

The AI that we're using is trained to detect human faces. **False positives are things that are not actually a human face (false) that it detects as a face (positive).** What are some false positives you can find?

Run the Face Filter starter project, make sure your own face is out of frame, and hold up something with a face (like a drawing of a face, a picture of an animal face, or a toy). See if the hat and glasses track smoothly (a false positive!) or randomly jump around (indicating it does not recognize the object as a face).

What variables can you change to try to fool it into thinking it sees a face? Do all elements of a face (eyes, ears, nose, and mouth) need to be present for the AI to detect a face?

People can easily see that an animal has a face, but why might some animal faces (like cats) be recognized while others (like cows or dogs) may not be seen as a face? Think about the proportions of a human face and how those are different from a cow's face with a longer snout. Those measurements don't match the human faces in the photos the AI trained on, so it doesn't see a cow's face as a face!



Drawing of smiley faces

Animal faces

LEGO head

Filled in circle

Examples of possible false positives: drawings of faces, animal faces (cat may be seen as a face, while cow not), Lego head, a filled in circle (depending on the shading and direction of the lines, etc.).

False Negatives

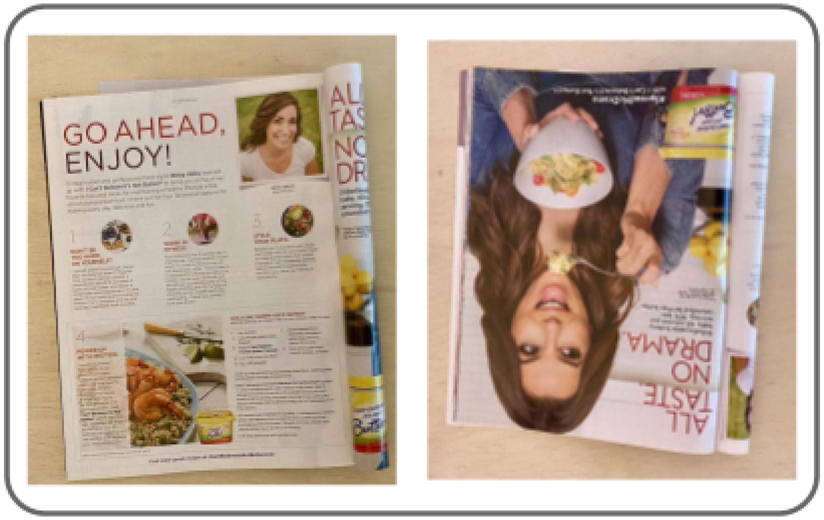
A false negative when working with the Face Sensing blocks is when a human face is in the frame, but the AI does not detect it. What are some false negatives you can find?

Using the [Face Filter starter project](), try to fool the AI into not seeing a face. See if the hat and glasses track smoothly, or if they randomly jump around (a false negative!).

Can the AI find the parts of a face if:

- you are in disguise or your face is covered?
- your face is tilted or upside down or far away? (to make this easier, try holding up a photo of a face)
- the lighting in the room is very bright or very dark?

People can see an upside-down face is a face, but AI may not be able to. Why? Likely, the data set the model was trained on contained photos that were all rightside up, so it expects that.



Examples of possible false negatives: too small or upside down.

Classroom Reflection Prompts

Face Sensing blocks provide a meaningful opportunity to open up conversations and reflect with your learners about AI ethics and how these concepts will shape our future:

- What does it recognize as a face? What were you surprised it recognized/did not recognize? Does it appear to treat people of different races, genders, and ages fairly?
- What do you think face detection software (like the one that powers the Face Sensing extension) was developed for?
- What if you don't want to be seen?

Recommended Reading:

- [Exploring a Creative, Safe Introduction to Machine Learning]() (Scratch Team) - This blog post introduced the Face Sensing blocks when they initially launched on our Scratch Lab site for experimental blocks. [Learn more about Scratch Lab here]().

Part 4: Reflect and Share

Reflect

Learners can reflect on their project creation and process as they complete the Face Sensing Project Sharing Sheet. Next, their peers are encouraged to leave feedback or comments on the sheet for the creator as they view the projects in a studio or participate in the gallery walk.

Resources:

- Worksheet | Face Sensing Project Sharing

Share Option #1: Create a Class Studio to Gather Shared Projects

Studios are a space on Scratch where users can come together to make, share, and collect projects related to a particular theme, idea, or prompt. Set up a class studio* for your learners and add their original asset projects. Learners are encouraged to take time to look at projects and read/listen/interact with them to learn more about their peers.

Resources:

- Teacher Accounts - Information on setting up teacher accounts and student accounts, managing classes, and class studios.
- Scratch Studios Guide - General information on setting up and managing.

**Note: Learners need a Scratch account and access to the online editor to participate in this option.*

Share Option #2: Gallery Walk

Have each participant's project open on their computer or other device. Participants can walk around a room, or take turns sharing their screen in a virtual space, to experience each other's creations. Or display one project at a time on a large screen. Learners are encouraged to take time to look at projects and read/listen/interact with them to learn more about their peers.

Standards Aligned

AILit Framework (a joint initiative of the European Commission and the Organization for Economic Cooperation and Development)
Based on the review draft of May 2025, our lesson touches on the following competences:

Engaging with AI Competences
1. Recognize AI's role and influence in different contexts.
3. Examine how predictive AI systems provide recommendations that can inform and limit perspectives.
4. Explain how AI could be used to amplify societal biases.
6. Analyze how well the use of an AI system aligns with ethical principles and human values.
7. Connect AI's social and ethical impacts to its technical capabilities and limitations.

Creating with AI Competences
1. Use AI systems to explore new perspectives and approaches that build upon original ideas.
2. Visualize, prototype, and combine ideas using different types of AI systems.

Managing AI Competences
1. Decide whether to use AI systems based on the nature of the task.

Designing AI Competences
2. Compare the capabilities and limitations of AI systems that follow algorithms created by humans with those that make predictions based on data.
5. Describe an AI model's purpose, intended users, and its limitations

CSTA Standards

Link to full standards

- 1B-AP-09 - Create programs
- 1B-AP-10 - Create programs
- 1B-AP-12 - Modify, remix, or incorporate
- 1B-AP-15 - Test & debug a program
- 1B-IC-18 - Discuss technologies
- 1B-IC-19 - Accessibility & usability

ISTE Standards

Link to full standards

- 1.1d Technology Operations
- 1.3.b Evaluate Information
- 1.4.b Design Constraints
- 1.6.a Choose Platforms or Tools
- 1.6.b Original and Remixed Works

CASEL Framework

Link to full standards

- Self-Awareness
- Social Awareness
- Responsible Decision-Making

RITEC Indicators

Link to full standards

- Autonomy
- Competence
- Creativity
- Diversity, equity & inclusion
- Safety and security

This lesson also fulfills all three of the ISB Indicators of Playful Learning (Choice, Delight, Wonder), developed by the Pedagogy of Play (PoP) research project at Harvard University.

Tip: If you'd like to translate this guide, **click here to make a copy** of this Google doc.