

# Severn Edge AI — Classroom Guide

**Duration:** 3 hours **Audience:** 5th graders (10-11 years old) **Goal:** Students train and deploy their own gesture recognition AI

---

## Materials Checklist

- [ ] Arduino Nano 33 BLE Sense (one per student) — firmware pre-flashed
  - [ ] USB cables (+ 3–4 spares — loose cables kill models)
  - [ ] Laptops with Chrome browser (BLE doesn't work in Firefox/Safari)
  - [ ] Web app running at <https://solkem.github.io/severn-edge-ai/> (or `npm run dev` locally)
  - [ ] Web app **pre-loaded** on all laptops before class (the page is ~1 MB, slow on weak WiFi)
  - [ ] Number stickers on each Arduino matching its BLE name (e.g. sticker "8" → SevernEdgeAI-8)
  - [ ] Visual slide deck on projector (4–5 slides — see Slide Deck section at end)
  - [ ] This guide (printed or on your own device — students don't see this)
- 

## Hour 1: "What Does AI Actually See?" (60 min)

### Hook (15 min)



#### SLIDE 1 — Title slide on projector

Ask the class: "**Is AI magic?**"

Hold up the Arduino. "This tiny board is going to learn from YOU today."

Open the app and show the **Preview page** — the 6 sensor bars jump around as the board moves. Key insight: **AI sees numbers, not motion.** It finds patterns in those numbers to figure out what you're doing.

### Live Demo (15 min)

Walk through the full flow end-to-end so students see the finish line before they start:

1. Connect to Arduino via Bluetooth
2. Explore the live sensor data on the **Preview page** (the 6 numbers)
3. Choose gestures on the Collect page
4. Record a few samples (wave, shake)
5. Train the model (watch the accuracy climb)
6. Deploy to Arduino
7. Test it live — wave and see the prediction appear

Keep it fast. The goal is excitement, not explanation.

## Connect (15 min)

- Each student plugs in their Arduino and opens the web app in Chrome
- Click Connect, find their board by matching the number sticker (e.g. "SevernEdgeAI-8")
- Celebrate each successful connection — green dot appears in the top bar

**Troubleshooting:** If a board doesn't appear, check USB is plugged in and try refreshing the page. If the board shows as "Arduino" instead of "SevernEdgeAI-N," unplug for 5 seconds and replug.

## Explore the Sensor — Preview Page (15 min)

### SLIDE 2 — "How the Sensor Captures Your Move" poster on projector

After connecting, students land on the **Preview page** — a dedicated sensor exploration screen. This is the most important teaching moment of the session: students see the raw numbers before touching any AI buttons.

#### What students see:

1. Click "**Start Seeing Numbers**" — the screen shows 6 live sensor bars updating in real time
2. **Red bars** ( $ax$ ,  $ay$ ,  $az$ ) — accelerometer. Measures movement and gravity.
3. **Blue bars** ( $gx$ ,  $gy$ ,  $gz$ ) — gyroscope. Measures rotation speed.
4. Bars grow **from the center** — right for positive, left for negative. Students can see direction, not just size.
5. A **raw numbers grid** at the bottom shows exact values.

Have students move the Arduino and watch the numbers react for a minute or two before starting the challenges.

#### Key things to say while they explore:

- "The AI has no camera, no ears — just these 6 numbers, 25 times per second"
- "Everything you're seeing right now is ALL the information the AI gets"

## Sensor Challenges — Student-Paced (10 min)

The Preview page has **4 built-in challenges** in a sidebar. Each challenge has:

- An instruction telling students what to try
- A "**Need a hint?**" toggle if they're stuck
- A "**Got it! I see what happens**" button they click when they've observed enough

**Students control the pace.** There is no timer and no auto-detection. They explore until they understand, then move on.

Challenge	What Students Do	What They Learn	Your Follow-Up Question
Wave It	Wave the board side to side	ax and ay bars swing — motion shows up in the accelerometer	"Which bar moved the most — ax or ay? Why?"
Hold Still	Hold the board flat and still	Face-up, az stays near +1.0 even when "nothing" is happening — that's gravity!	"Why isn't it zero? What force is the sensor still feeling?"
Find gz	Spin the board like a steering wheel	When held flat, gz moves most — each axis measures a specific type of rotation	"What happens if you tilt instead of spin?"
Flip It	Slowly flip the board upside down	If you start face-up, az goes from +1.0 to -1.0 as gravity reverses direction	"Why did the bar swap sides?"

**Your role during challenges:** Circulate the room. Don't give answers — ask follow-up questions (right column above). The hints in the app give students a nudge if they're stuck. When most students have completed all 4, tell them to click "**I'm Ready! Let's Collect Data →**" to proceed.

#### Bonus challenge (verbal, if time allows):

"Before you move on — try two different gestures on the sensor panel. Do a big circle, then stir a pot. Watch the numbers. Are they the same to the sensor?"

Usually yes — two gestures that *look* different to a human can look *identical* to the sensor. This is the deepest lesson: **the AI doesn't see what you see. It sees what the sensor reports.** Students who grasp this will pick better gestures in the next phase.

## The 600 Numbers (2 min)

### SLIDE 3 — "What the AI Actually Sees" poster on projector

Before students leave the Preview page, point to the slide and say:

"When you record a gesture, the sensor takes 25 readings per second for 4 seconds. That's 100 readings. Each reading has 6 numbers. So your entire gesture — whether it's a wave, a shake, or a spin — is just **600 numbers** to the AI. That's it. Different gestures make different patterns of 600 numbers, and the AI learns to tell those patterns apart."

This is also the moment to point at the heatmap comparison at the bottom of the slide: **"See how Wave and Shake look totally different as colored grids? That's what the AI is looking at."**

Don't dwell — 2 minutes max. Tell students to click "**I'm Ready!**" to move to the Collect page.

## Hour 2: "Teach Your AI" (60 min)

### What Makes Good Training Data? (5 min)

Show two examples side by side:

1. **Good data:** Someone waves consistently 10 times, same speed, same motion

2. **Bad data:** Someone waves differently every time, sometimes barely moves

Train both. Show the accuracy difference.

Core lesson: **Garbage in, garbage out.** The AI can only learn patterns if the patterns are consistent.

## Choose Gestures (10 min)

Each student picks their gestures on the setup screen. Encourage creativity:

- "Cast a spell"
- "Stir a pot"
- "Swing a bat"
- "Draw a star"
- "Throw a ball"
- "Brush your teeth"

**Rule:** Gestures must be physically distinct. If two gestures feel similar (like "small wave" and "big wave"), the AI will struggle. This is a great teaching moment when it happens. Students who did the sensor challenges will already know to check the panel before committing.

Students can rename the defaults, remove them, or add new ones (up to 8, minimum 1).

 **Single-gesture note:** If a student only picks 1 gesture, the app automatically adds an "Idle" class. This is normal — the AI needs at least two things to compare. "Idle" means "not doing the gesture." The student doesn't need to record samples for Idle; the app generates them automatically.

## Collect Training Data (25 min)

Students record 10 samples per gesture:

1. Select a gesture card
2. Click "Record Sample"
3. Perform the gesture for 4 seconds — each recording captures 100 sensor snapshots (those 600 numbers from the poster!)
4. Wait for the ✓ or 💪 feedback — green confetti means it was accepted
5. Repeat until all gestures show "DONE"

**Tip:** Encourage students to keep the sensor panel open while recording. Watching the numbers helps them understand why consistent motion matters — if the bars look different every time, the AI will struggle.

Walk around the room helping. Common issues:

- **"It keeps rejecting my sample"** — Move bigger! The quality check needs to see real motion.
- **"My accuracy is bad"** — Are your gestures different enough? Check the sensor panel. Collect more samples.
- **"I only want to do one gesture"** — That's fine! The app auto-adds "Idle" so the AI can tell the difference between your gesture and doing nothing.

## Train (10 min)

1. Click "Next: Train Model"
2. Watch the training progress — accuracy should climb above 70–80%
3. Training takes about 30 seconds for 3 gestures

 **SLIDE 3 on projector during training** — While students wait, point to the poster: "Right now, the AI is looking at all your 600-number patterns and learning the difference between them. That's what training is — finding the pattern."

**If accuracy is low (below 60%):** Students can click "**Train More**" to run additional training rounds. Each round refines the model. They can also go back and collect more samples for weak gestures.

## Deploy to Arduino (10 min)

This is the moment the AI goes from the laptop to the tiny board.

1. Click "**Upload via Bluetooth**"
2. A progress bar shows the upload (~10–20 seconds for a typical model)
3. When it says "**Model deployed! ✓**" — the model is now running on the Arduino

### What to expect:

- The upload sends ~78 KB of data over Bluetooth in small chunks
- Progress bar should move steadily from 0% to 100%
- If it stalls or shows an error, click the upload button again — it will retry

 **Important — RAM warning:** The model lives in the Arduino's memory, not saved to permanent storage. **If the USB cable wiggles loose or the board loses power, the model is gone** and students must re-upload. Tell students: "**Don't unplug your Arduino!**"

## Hour 3: "The Gesture Games" (60 min)

### The 10-Turn Challenge — Warm-Up (15 min)

 **SLIDE 4 — "Challenge Time! Can your AI score 7/10?" on projector**

Before the big contest, students test their own models using the **built-in 10-Turn Challenge** on the Test page:

1. Click "**Start Testing**" — the app verifies the Arduino has a model loaded
2. The sidebar shows "**Do this gesture now: [name]**" — a target gesture to perform
3. The student performs the gesture and watches the big prediction label
4. When the prediction matches, tap "**Score Attempt**"
5. ✓ or ✗ appears — the app scores whether the majority of recent predictions matched
6. Target rotates to the next gesture
7. After 10 turns, they get a final score (e.g. 7/10 = 70%)

**Key teaching moment:** The 10-Turn Challenge accuracy is usually **lower** than the training accuracy. That's normal! Training accuracy is "how well did it learn the examples." Challenge accuracy is "how

well does it work in the real world." This gap is called **overfitting** — but you don't need to use that word with 5th graders. Just say: "**Training is like studying. The challenge is the real test.**"

**If a student's score is low:** They can go back, collect more samples, and re-train. The "Train More" button lets them add rounds without starting over.

## Contest: The Swap Challenge (30 min)

This is the main event. It tests whether students' models actually learned real motion patterns.

### Setup (5 min):

- Each student writes down their gesture names on a piece of paper (face down)
- Students pair up with a neighbor

### Round 1 — Guess the Gestures (10 min):

- Partners swap Arduinos (keep them plugged in!)
- You must figure out the other student's gestures by trying different motions and reading the predictions
- You get the prediction label (e.g. "Gesture 1," "Gesture 2") but NOT the name
- First student to correctly name all gestures wins the round

### Round 2 — The Stranger Test (10 min):

- Each student demonstrates their gestures live
- But their partner performs them on the Arduino
- Use the 10-Turn Challenge to score — how many out of 10 does the model get right?
- Highest score wins

### Why this works:

- Proves the model learned real patterns, not just memorized one person's style
- Physical, loud, competitive — kids love it
- Teaches generalization: did your model learn "the gesture" or just "your hand"?

### Scoring:

Round	Points
Swap Challenge: Guess each gesture	3 points per correct guess
Stranger Test: 10-Turn Challenge score	1 point per correct prediction

## Reflection (10 min)

Bring the class back together. Ask:

- "**What made your model better?**" (More data, distinct gestures, consistent recording)
- "**What made it worse?**" (Similar gestures, sloppy recording, not enough samples)
- "**What surprised you?**" (How fast it trained, how it sometimes gets confused, how it works without seeing you)
- "**Could the AI learn any gesture?**" (What are the limits?)

- **"Why was the challenge score lower than the training score?"** (Training is studying, the challenge is the real test)

Draw out the core lessons:

1. **Data quality matters** — consistent, clean data makes better models
2. **Distinct patterns help** — the AI needs to tell things apart
3. **More data helps** — but only if it's good data
4. **AI isn't magic** — it's pattern matching on 600 numbers

## Wrap-Up (5 min)



### SLIDE 5 — Closing slide on projector

"You just trained and deployed a real neural network. That's what AI engineers do — they collect data, train models, and put them on devices. The only difference is scale. You used 30 samples and 6 sensors. A self-driving car uses millions of samples and hundreds of sensors. But the idea is exactly the same."

## Alternative/Bonus Contest Ideas

### The Fourth Gesture Challenge

After the main contest, students add a 4th gesture and retrain:

- Can you find a gesture that doesn't hurt accuracy?
- Students compete for highest 4-class accuracy
- Teaches: more classes = harder classification, class boundaries matter

### Fool the AI

Students try to find motions that trick another student's model into wrong predictions:

- Most successful "fooling" wins
- Teaches adversarial thinking: what confuses ML?
- Example: a gesture that's halfway between two trained gestures

## Teacher Notes

### Before Class

- Flash firmware to all Arduinos and verify each connects via BLE
- Note each board's number (e.g. SevernEdgeAI-8) and label the physical boards with stickers to match
- Test the web app on the classroom laptops (**Chrome only** — BLE doesn't work in Firefox/Safari)
- Pre-load the web app URL on all laptops (avoids 20 simultaneous 1 MB downloads on class WiFi)
- Have 3–4 spare USB cables and a backup laptop in case of Bluetooth issues

- Print this guide for yourself — students should never see this document

## Common Failure Modes

Problem	What the Student Sees	Fix
Board not appearing in Bluetooth dialog	Empty device list	Unplug USB, wait 5 seconds, replug, refresh page
Board shows as "Arduino" not "SevernEdgeAI-N"	Wrong name in list	Unplug for 5 seconds, replug. The name should appear correctly.
Training accuracy stays low (<60%)	Low percentage after training	Collect more samples, or make gestures more distinct. Use "Train More" for extra rounds.
Model upload fails mid-way	Progress bar stalls or error message	Check USB cable is firm (Bluetooth depends on power). Click upload again — it retries.
Upload shows "CRC error"	Error message on screen	The data got corrupted during transfer. Just click upload again. Works on retry.
"No trained model on the Arduino" on Test page	Error when starting test	Student skipped upload. Go back to Train page and click "Upload via Bluetooth."
Model disappeared / predictions stopped	Arduino stopped responding	USB cable wiggled loose — model lost from memory. Replug and re-upload.
Predictions always the same class	One gesture name stuck on screen	Model didn't learn well. Retrain with more distinct gestures and more samples.
Different predictions every time	Predictions flickering randomly	Gestures are too similar, or recording was inconsistent. Retrain with clearer motions.

## Key Vocabulary for Students

- **Training data** — Examples you show the AI so it can learn
- **Model** — The AI's "brain" after training — really just a pattern of numbers (called weights)
- **Inference** — When the AI makes a prediction on new data
- **Accuracy** — How often the AI gets the right answer
- **Sensor** — The chip that measures motion (accelerometer + gyroscope). Students can see live sensor values in the "See what the AI sees" panel
- **Deploy** — Sending the trained model to the Arduino over Bluetooth
- **Window** — The 4-second, 600-number snapshot the AI looks at each time it makes a prediction

## Slide Deck Reference

The student-facing slide deck should have these slides (generate as images, minimal text):

Slide	Content	When to Show
1	Title: "Severn Edge AI" + Arduino photo + "Today you teach a robot"	Opening hook
2	Visual poster: "How the Sensor Captures Your Move" (6 graphs, timeline)	Explore the Sensor section
3	Visual poster: "What the AI Actually Sees" (600-number table + heatmaps)	The 600 Numbers section + during training wait
4	"Challenge Time! Can your AI score 7/10?" + challenge icon	Start of Hour 3
5	"You just trained a real neural network" + recap of 4 lessons	Wrap-up