USC University of Southern California

ECE 105: Introduction to Electrical Engineering

Lab 4.2

Camera classifier

Yasser Khan

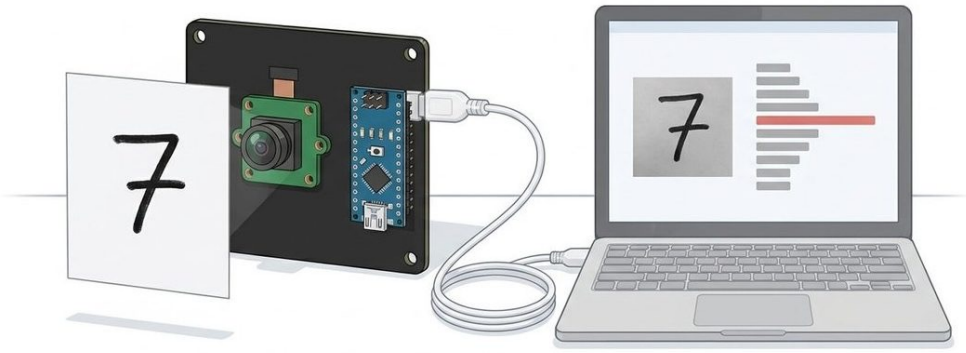
Rehan Kapadia

Today's mission

Show the board's camera a digit you wrote, and let your Lab 4.1 network read it.

By the end of this lab you can:

- Stream **live pictures** from the camera into the notebook
- Turn a picture into a **28 x 28 MNIST-style input**
- Classify your digits live with your **Lab 4.1 CNN**
- Measure the **accuracy on your own handwriting**

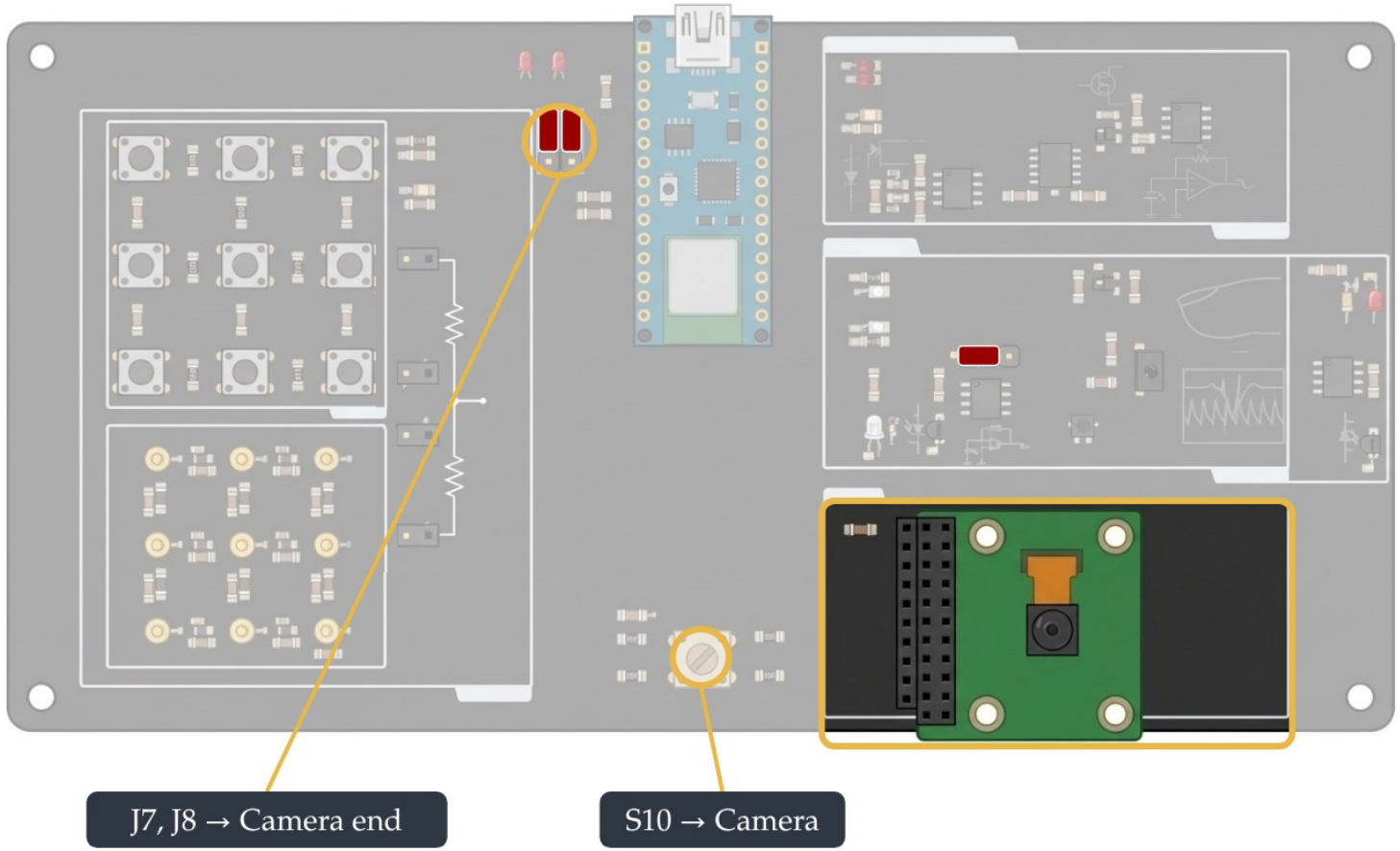


A paper digit, the board's camera, and the network's answer.

A	B	C	D	E	F
Camera **10 min**	Live view **15 min**	Preprocess **10 min**	Load CNN **5 min**	Predict **15 min**	Test set **10 min**

SETUP

The camera lab moves the jumper caps. Move them back to Module 1 when you are done.



1. **S10 → Camera**
   Turn the selector to Camera.

2. **J7, J8 → Camera end**
   Both caps on pins 1-2: A0 and A1 go to the camera.

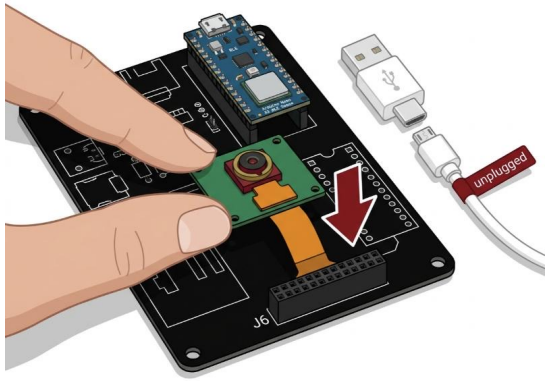
3. **Camera on J6**
   The OV7675, lens facing out. Seat it with the USB cable unplugged.

4. **Digits on paper**
   0 to 9 on one sheet, thick black marker, each about 4 cm tall.

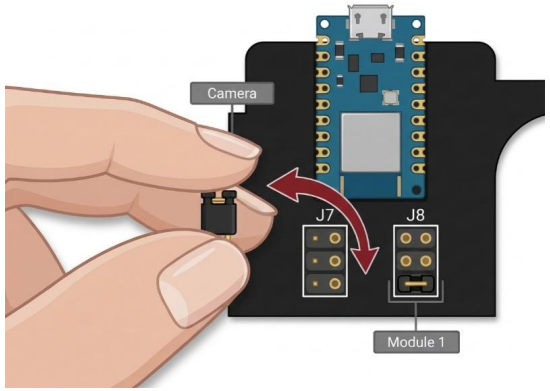
Part A: Connect the camera

Unplug, seat the camera, move the caps, plug back in, run the first cell.

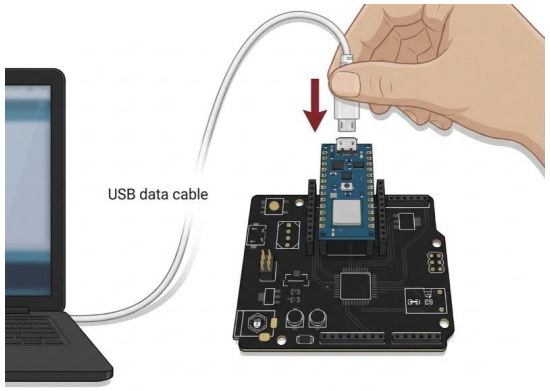
PART A **Goal: A1 · DO** connect the camera and upload its firmware.



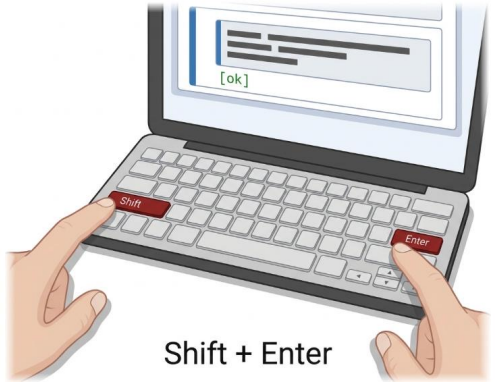
1 USB unplugged: seat the camera on J6



2 J7, J8 (3-pin headers): caps to pins 1-2, the Camera end



3 Plug the USB cable back in



4 Run the first cell with Shift+Enter

You should see: two **[ok]** lines. The first time, **connect_camera** compiles and uploads the camera firmware: about a minute.

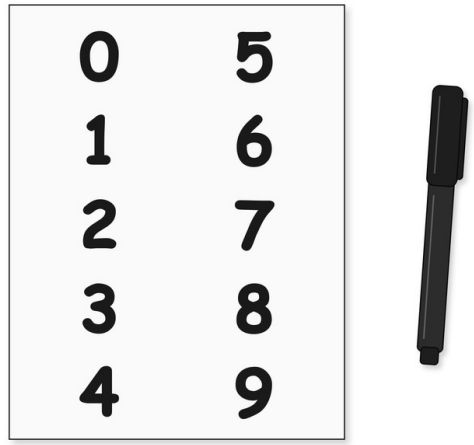
Part B: Look through the camera

Scan your ten digits once, here. Every later part reuses these scans.

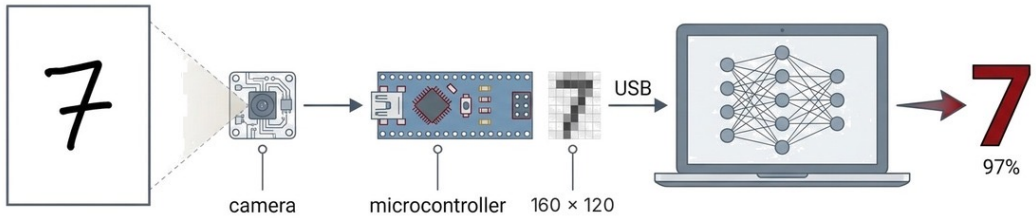
PART B

Goal: look through the camera, scan your ten digits, fix the orientation.

- **B1 · DO cam.live(15)**: a digit **in front of the lens**, 10 to 20 cm away
- **B2 · DO** Scan **0 to 9**: show the digit named in the red title, 15 s each, hold still at the end
- Missed one? The next cell asks for a digit and **scans it again**
- **B3 · ANSWER** Upside down or mirrored? **FLIP_UD**, **FLIP_LR**
- **B4 · RUN** One of your scans, the right way up



Write 0 to 9 with a thick black marker; B2 scans them one at a time.



Camera, Nano, USB, laptop: the network runs on the laptop.

You should see: ten panels that fill one after another, each with a **clear digit** that fills about half the picture; paper light gray, not black or burned out. Then **[ok] 10 pictures scanned**.

Parts C and D: Network input and your CNN

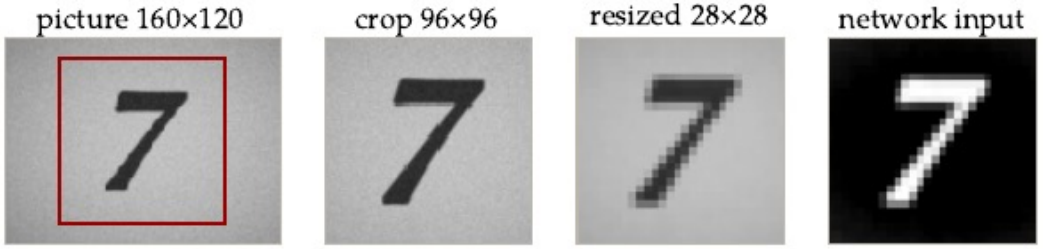
USC University of Southern California

to_digit: center crop, 28 x 28, invert, stretch; then the Lab 4.1 normalization.

PARTS C, D

Goal: turn one of your scans into a network input and load your CNN.

- **C1 · DO** Uses your scan of **DIGIT** (0 to 9): crop, resize, invert, stretch
- Then (x - 0.5)/0.5, exactly as in Lab 4.1
- **D1 · RUN** Load **lab4.1_cnn.pt** from the lab folder (your own from Lab 4.1 Part J also fits)



160 x 120 picture, center crop, 28 x 28, network input.

Must match Lab 4.1 training

- white digit on black, 28 x 28
- digit about 20 px tall, centered
- pixels scaled with (x - 0.5) / 0.5
- the same SmallCNN class

You should see: a **white digit on black** that fills the middle of the 28 x 28 input, then **[ok] loaded your Lab 4.1 CNN** and an MNIST accuracy near **98.5 %**.

Check (Checkpoint 1): your scan is the right way up (B4), and the preprocessing ends in a centered white digit on black.

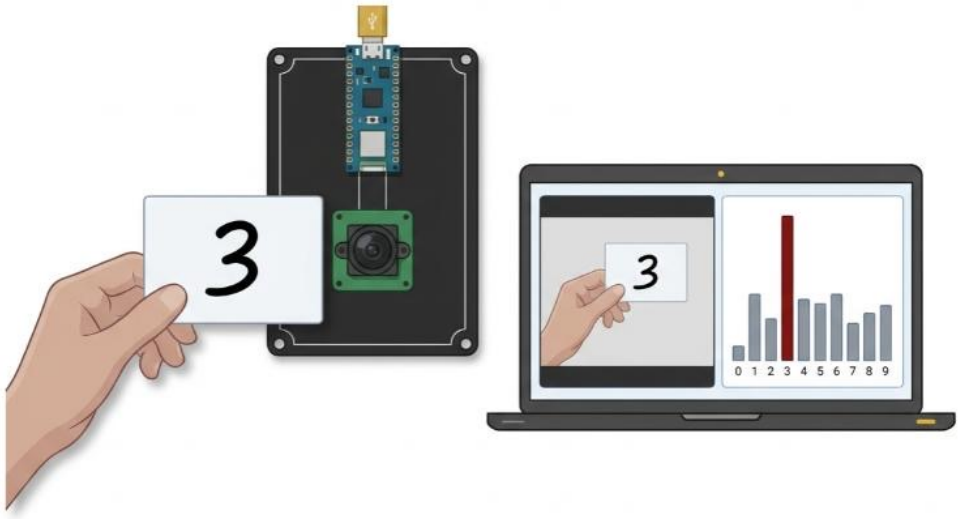
Part E: Live prediction

The largest of the ten softmax probabilities is the network's confidence.

PART E

Goal: predict your digits live, then record three predictions from your scans.

- **E1 · DO** Live view with predictions: hold up different digits (change the 15 in **cam.live** for more time)
- **E2 · DO** Choose three scanned digits in **DIGITS**, for example [7, 2, 3]
- **E3 · EDIT** Softmax by hand from the ten logits z of one scan; compare with PyTorch



Hold the digit facing the lens; the tallest bar is the prediction.

You should see: a thick, clear digit at **95 % or more**; an ambiguous one (an open 3) near **40 to 50 %**.

Check (Checkpoint 2): the live view predicts your digits, and you have three prediction figures.

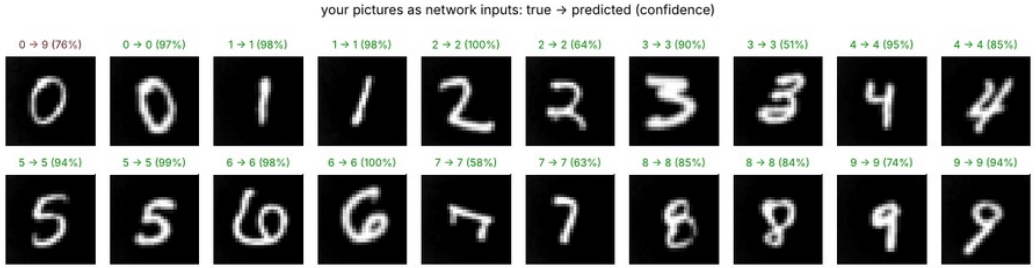
Part F: Your own test set

The ten digits you scanned in Part B are the test set of your handwriting.

PART F

Goal: measure the accuracy on your handwriting and see the confusion matrix.

- **F1 · RUN** Your ten B2 scans become the test set (**my_digits.npz**); nothing is scanned again
- **F2 · RUN** The contact sheet of your 10 inputs
- **F3 · EDIT** The accuracy **cam_acc** in percent; the confusion matrix is drawn for you



Typical contact sheet: a red title marks a wrong prediction.

You should see: typically **60 to 90 %** on your 10 pictures, below the **98.5 %** on MNIST. Saved: my_digits.npz and my_digits_results.csv.

Check (Checkpoint 3): the contact sheet, your accuracy next to MNIST's, and the confusion matrix.

Stuck?

Check these first. If the fix does not help, raise your hand.

Camera not found

Unplug USB, reseat the camera on J6 (lens out), plug in, press RESET.

The picture is black or burned out

Change the light or the distance: paper light gray, ink dark.

The digit is upside down or mirrored

Set FLIP_UD or FLIP_LR to True in B3 and rerun it. No need to scan again.

[!!] lab4.1_cnn.pt not found

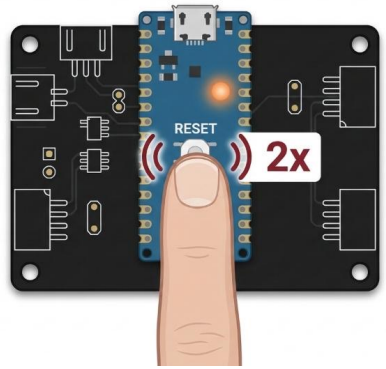
Put lab4.1_cnn.pt next to the notebook, or let D1 train a CNN now (15 s).

Poor accuracy on your digits

Thick marker, digit centered and half the height, no glare. Scan a poor digit again with the cell after B2.

A [!!] line about the selector

Only a warning. Check S10 on Camera and both caps on the Camera end.



Upload stuck? Double-tap RESET, then rerun.

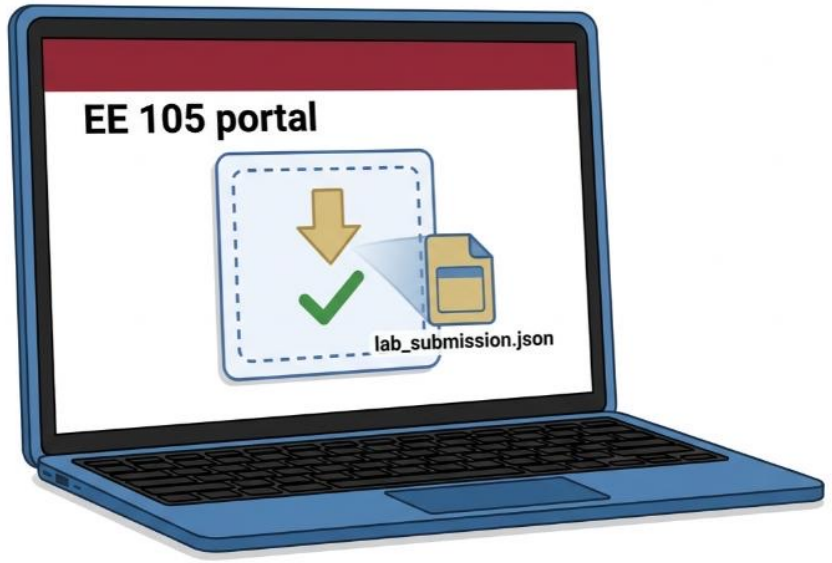
Submit your results

USC University of Southern California

The last cell saves your results in one file. Upload it on the course portal.

```
lab4.2_submission.json
```

1. **Save · RUN** writes **lab4.2_submission.json** next to the notebook
2. It holds your numbers, the labels and predictions of your 10 pictures, and their 28 x 28 inputs
3. Log in at **ee105.org/portal** with your student ID and portal PIN; open Lab 4.2
4. Upload the file. It is graded automatically; you can upload again after a fix



ee105.org/portal

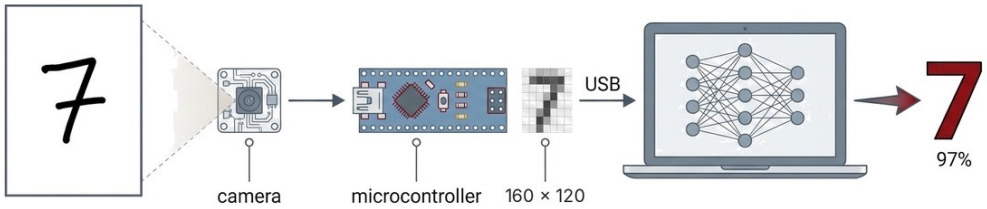
Check: before you leave, move both caps of J7 and J8 back to the **Module 1** end (pins 2-3).

Lab complete

You built a camera that reads handwriting with your own neural network.

You built:

- A live camera view inside the notebook
- One preprocessing for photos and camera pictures
- A test set of your ten digits and its confusion matrix
- The network's confidence, computed by hand with the softmax



Camera, microcontroller, laptop: 97 % on a 7.

Did you know?

Running machine learning on a microcontroller is called TinyML. The Nano has 256 KB of RAM; your CNN needs 808 kB in float32, so it would first be cut to 8-bit numbers (Chapter 4).

Next

Lab 5.1, LiFi: send text from one board to another with light. Caps of J7 and J8 back on the Module 1 end now.