Teaching Cozmo the wonderful world of Uno

Rebecca Manley & Nathan Glover

How do we recognize cards?

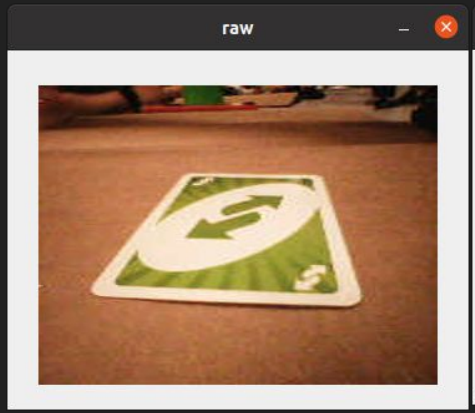
- Unfortunately, the internet was a bit lacking on cozmo height/angle training sets of uno cards
  - It was pretty lacking on uno cards in general, but we did find one birds-eye view sort of set. Not exactly the best for our situation
- So, we set out to train our own CNN, on our own card images

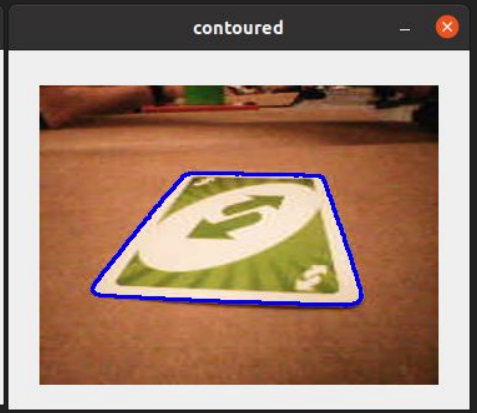
Simplifying the problem

- We wanted to minimize the complexity of the network, so we made some simplifications to what we wanted to identify
  - 1: don't consider color (we can determine that separately)
  - 2: give it only black and white, no grayscale
  - 3: 6's and 9's are too similar, so we're starting with no-9's uno
- So at the end of the day, we had 12 types of cards: numbers 1-8, skip, reverse, and +2
  - Technically 13 types, but we handle wildcards separately. They are the only black cards, so once we identify the color we do not have to identify the "number"

Making a nice image (example: green)

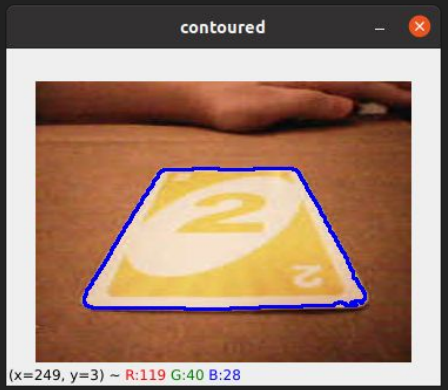
We get a raw color image from Cozmo. Threshold to separate the white parts of the card from everything else, then locate where the card is with contours (biggest contour). Use the contour bounding box to crop it square (“cropped”), and we’re good to go! We also, in an effort to force it to focus in the middle, tried cropping it even more (“smaller”).





The accursed yellow

That method worked *great* on red, blue, and green. Yellow… not so much.



(The card contour was still nice though, at least)

Our Solution: Look at the color

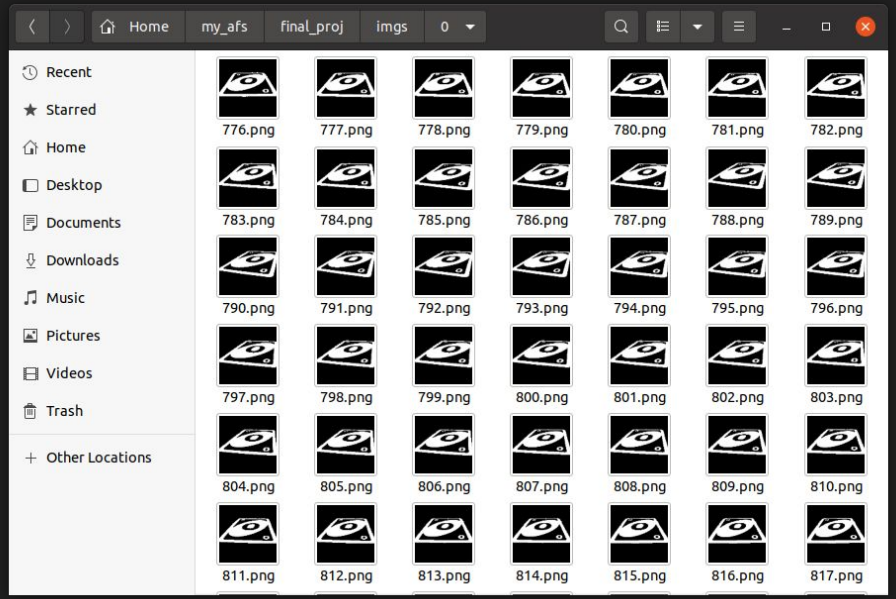
To get around the low contrast on yellow, we stopped looking for light/dark and started looking for yellow/not yellow. Pictured below (‘yellow mask’) is the result of our color thresholding which spits out a 255 if the color is in the range we called ‘yellow’, and a 0 otherwise (inverted to match the other pictures). Once we had this nice and crisp, we combined it with the threshold from before to remove the background white.

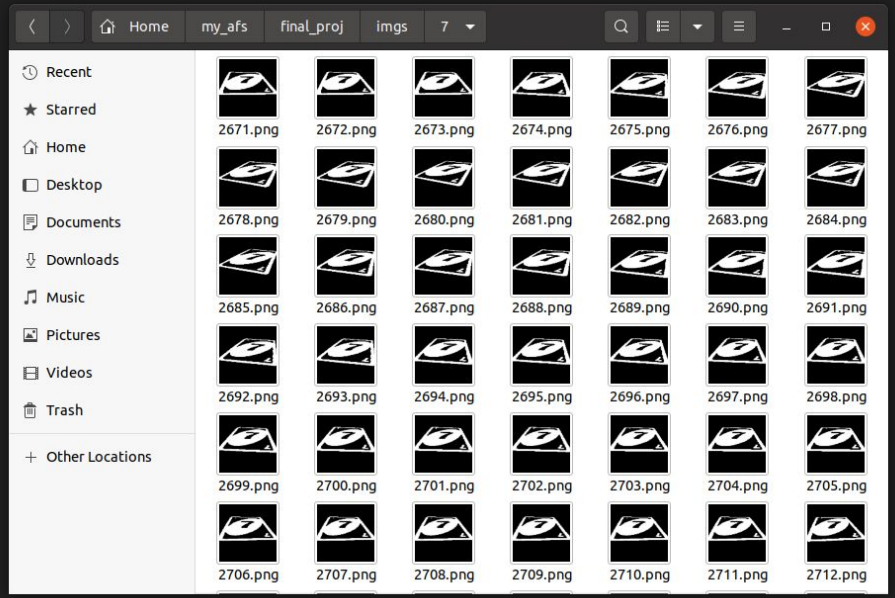
Cool, we can see cards… Now what? Collecting data

We made an FSM to take a continuous video (processed as we just discussed) and save each frame according to what the card is (which is supplied via TM by the user). To vary the images we were saving, we:

- Have the Cozmo move slowly forward and backward. This changes the viewing angle we see the card from, and we want to be a bit resilient to how we are positioned
- Continuously turn the card. We also want resilience to the angle of the card
    - (This was accomplished by taping it to the end of a chopstick shoved through the background, and turning the chopstick)

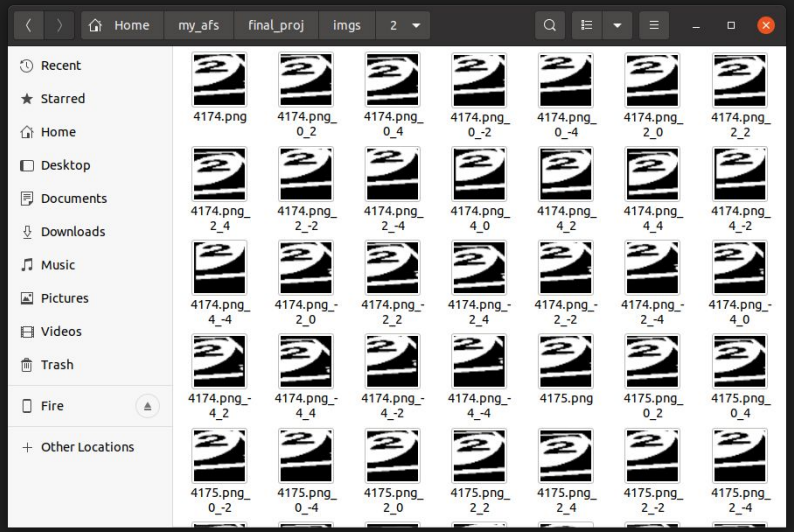
The Raw Data!

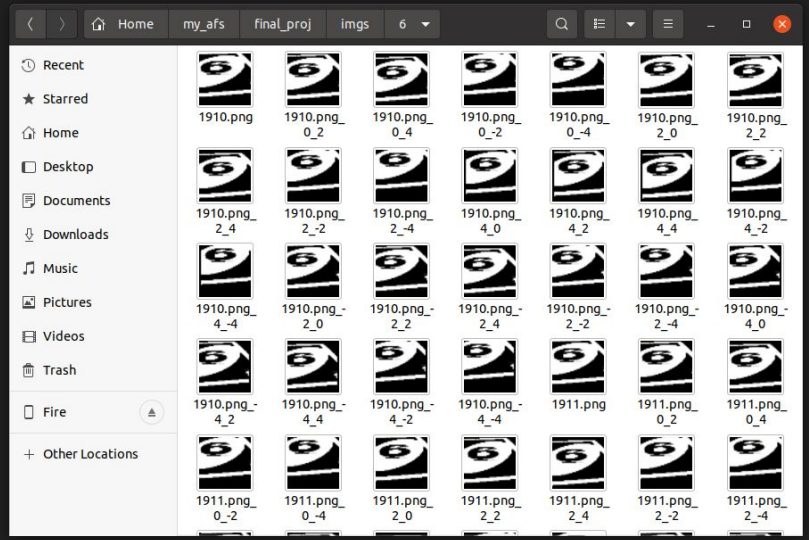

Home
my_afs
final_proj
imgs
0
Recent
Starred
Home
Desktop
Documents
Downloads
Music
Pictures
Videos
Trash
Other Locations
776.png 777.png 778.png 779.png 780.png 781.png 782.png
783.png 784.png 785.png 786.png 787.png 788.png 789.png
790.png 791.png 792.png 793.png 794.png 795.png 796.png
797.png 798.png 799.png 800.png 801.png 802.png 803.png
804.png 805.png 806.png 807.png 808.png 809.png 810.png
811.png 812.png 813.png 814.png 815.png 816.png 817.png


Home
my_afs
final_proj
imgs
7
Recent
Starred
Home
Desktop
Documents
Downloads
Music
Pictures
Videos
Trash
Other Locations
2671.png 2672.png 2673.png 2674.png 2675.png 2676.png 2677.png
2678.png 2679.png 2680.png 2681.png 2682.png 2683.png 2684.png
2685.png 2686.png 2687.png 2688.png 2689.png 2690.png 2691.png
2692.png 2693.png 2694.png 2695.png 2696.png 2697.png 2698.png
2699.png 2700.png 2701.png 2702.png 2703.png 2704.png 2705.png
2706.png 2707.png 2708.png 2709.png 2710.png 2711.png 2712.png

Then we decided that wasn’t enough data, so we made more!

We needed to crop them to match the crop we were testing on the camera view. We also shifted each picture by up to 4 pixels in each direction, to make 24 new slightly shifted images for each card photo we had.

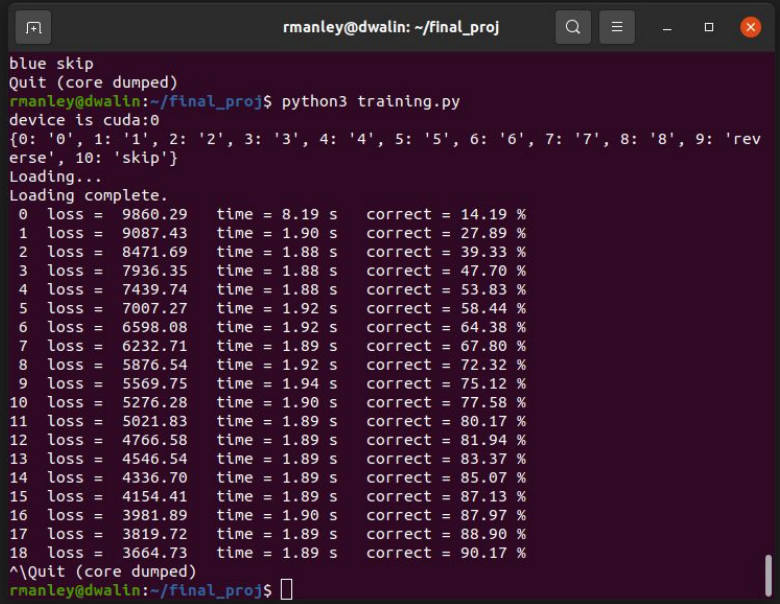




Time for the machine to learn

The general network we've been trying out for this is a two layer convolutional network. We also tried a couple of different datasets, with varieties such as

- Spin the cards a full 360
- Assume the card is facing the Cozmo, and only wiggle by about +/- 15 degrees
- Include +2 cards
- Don't include +2 (since unfortunately the deck we have is a bit quirky)
- Include the whole card
- Crop to focus on the center



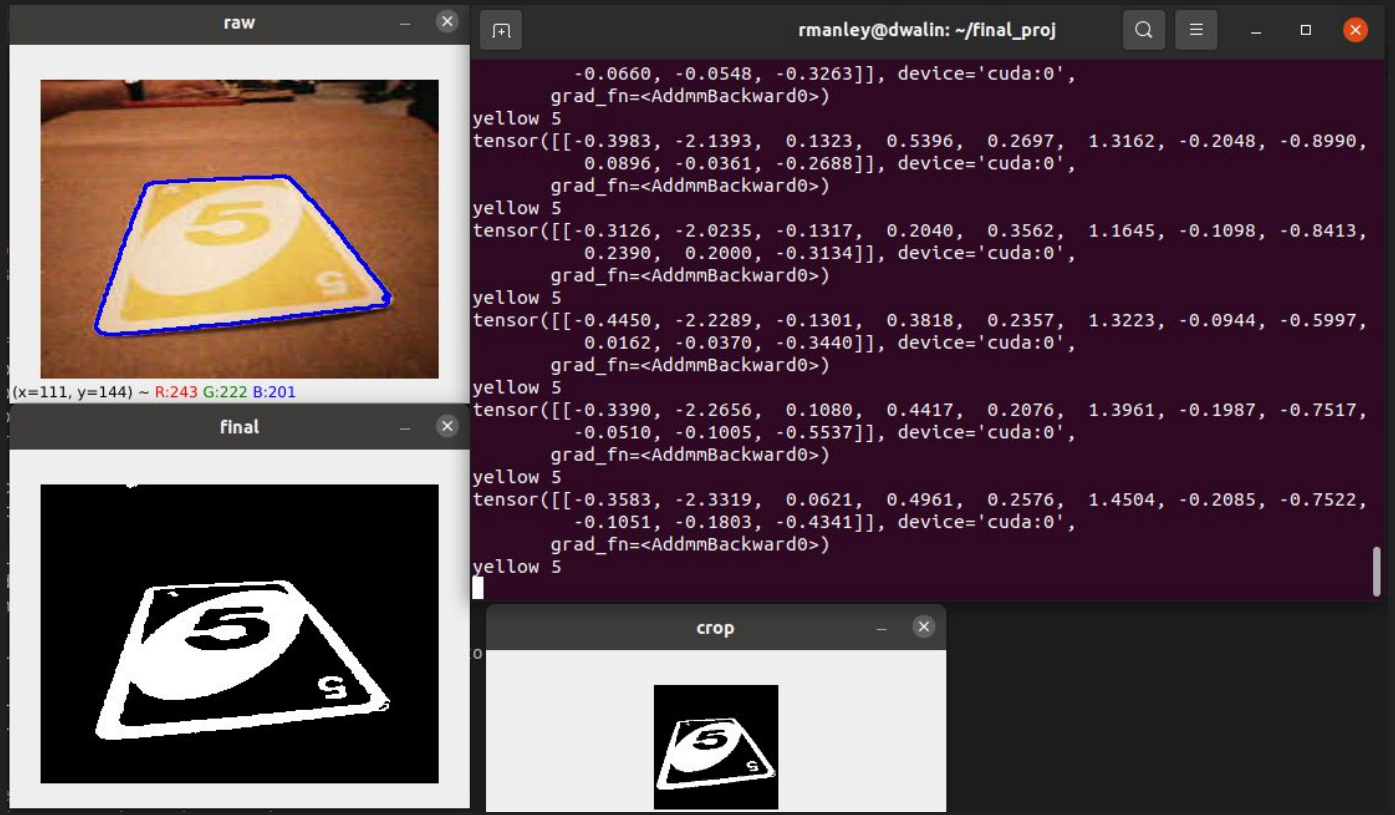
A Quick Note:

- Unfortunately somehow some of our card pictures got deleted from our training set. Specifically 3, 5, and reverse.
- We didn’t have a chance to retake these photos, so they are omitted from our most recent results, described in the next slide

Training went fine, but performance is… not the best…

- Some cards are very finicky (1, 2, 4)
  - With enough minor movements/tweaks, we can get them to reliably identify
  - But only in the “perfect” position and angle
- Some cards are correct more often than not, but still not entirely consistent. (0, 6, 8, skip)
- The only card that we tested that was highly consistent was 7
- (Note: color detection is being done via the sort of color thresholding we talked about earlier. We’ve specified ranges for each color, and assume that whichever range has the highest response is the color of the card. This is pretty reliable as long as the video doesn’t have a ton of stuff in the background)

Here is a 5 in its ideal position



Video demo of some cards

Why is it like this?

- With initial results, our best guess was that it was due to our sorta limited training set (~6k images) and/or the way we're training it, we're getting a decent amount of overfitting
  - Performance seemed to improve with the expanded (jittered) set of pictures (closer to 100k images)
- And/or maybe that something about our parameters isn't letting us catch enough detail or is letting it focus on too much detail, etc
  - Trying to force it to focus on the center of the card helped, but our method of doing this was pretty rough/unsophisticated, so it didn't give us the cleanest pics
- **Takeaway:** Our training set alterations have improved the number of cards we can semi-reliably identify, but it's still definitely not perfect. Possible improvements that could still be made:
  - Better way to isolate the center number
  - Fiddle with parameters such as weight decay, number of layers, etc

- Future work: continue trying to improve the network, and/or try a simpler method of number determination (like shape match)

Questionable card recognition aside, can we actually play the game?

- We did not have time to put all of the mechanics of the game together, but:
- We have a semi reliable system to manipulate (pick up and drop) the cards
  - Using a lift attachment created out of some wire and some tape, Cozmo can lower and raise his lift to pick up a card
  - In order to drop it, we need something to hold it down as Cozmo raises his lift back up
    - We tried a wire strung across the background, but that didn't provide a lot of force
    - Now we're trying a cardboard frame, which seems to allow for more reliable drop off.
  - It is a fine balancing act between being too sticky, and not sticky enough

Lift Demo

Future work: Putting together the pieces

- It is still our overall vision/dream/etc that Cozmo may one day play a game of Uno with minimal human intervention
- In order to accomplish this we imagine a world where:
  - Cozmo has a fixed number of card slots, each marked with an Aruco. The aruco will allow Cozmo to line up fairly reliably with any card in his hand
  - Cozmo has a little house/shield around his ‘hand’, for privacy
  - The central draw pile is face up, under a tent. An aruco will mark the entrance
  - The opponent will need to take the card from Cozmo when he wants to ‘play’ it and set it down
  - Basically arucos everywhere and a well specified world map, so Cozmo can most neatly navigate the ‘arena’
  - Cozmo will need to recognize his cards when he first picks them up. Afterwards, he’ll store what the card is (number and color) as well as the aruco slot in his hand where he places it.
  - When it is his turn to play, he will move to the central area and observe the top card, compute which card in his hand he could play and retrieve it, or (if no playable cards) draw a new card.