Exercise 1

Let's Communicate | Sharon Ku

PART 1: The construction of a model with only my data

1. My initial data set consists of 50 pictures of Jigglypuff, 50 pictures of Red Dragon, and 50 pictures of other objects in my room. I chose Jigglypuff and Dragon because they were small, portable, and had interesting features from all sides. Since I had to take photos of the objects, I made sure they weren't completely symmetrical to allow me to take unique shots from various angles. I knew we had to bring the objects to school so I chose the smallest objects in my room that won't get squashed in my school bag. The 50 objects from my room were chosen randomly, based on small objects that were closest to me so that I didn't need to do much work setting them up and putting them back.

Jigglypuff

Dragon

Examples of other objects: "Other"

2. The purpose of the task was to train a machine to detect whether a single object belonged in a pre-defined category of objects ("jigglypuff", "dragon", or "other") based on similar features. The model was trained by importing data of photos of the objects and photos that are not the objects.

3. The steps involved:
   a. Taking 50 pictures of Jigglypuff, 50 pictures of Red Dragon, and 50 pictures of other objects.
   b. Saving those photos under different folder names: "jigglypuff", "dragon", and "other."
   c. Uploading data: Importing those sets of photos into EdgeImpulse under the labels corresponding to the folder names.
   d. EdgeImpulse randomly chooses 80% of the photos to train the machine learning model, and uses the remaining 20% to test the model.
   e. Transfer learning allows previous data collected in EdgeImpulse to assist with extracting features of images to help with object detection.
   f. Deploying and testing model on computer by placing an object in front of the camera to determine which label the machine identifies it as.

4. **Model testing results**
   Accuracy: 83.87%

ACCURACY
83.87%

	DRAGON	JIGGLYPUFF	OTHER	UNCERTAIN
DRAGON	100%	0%	0%	0%
JIGGLYPUFF	0%	77.8%	0%	22.2%
OTHER	8.3%	8.3%	75%	8.3%
F1 SCORE	0.95	0.82	0.86	

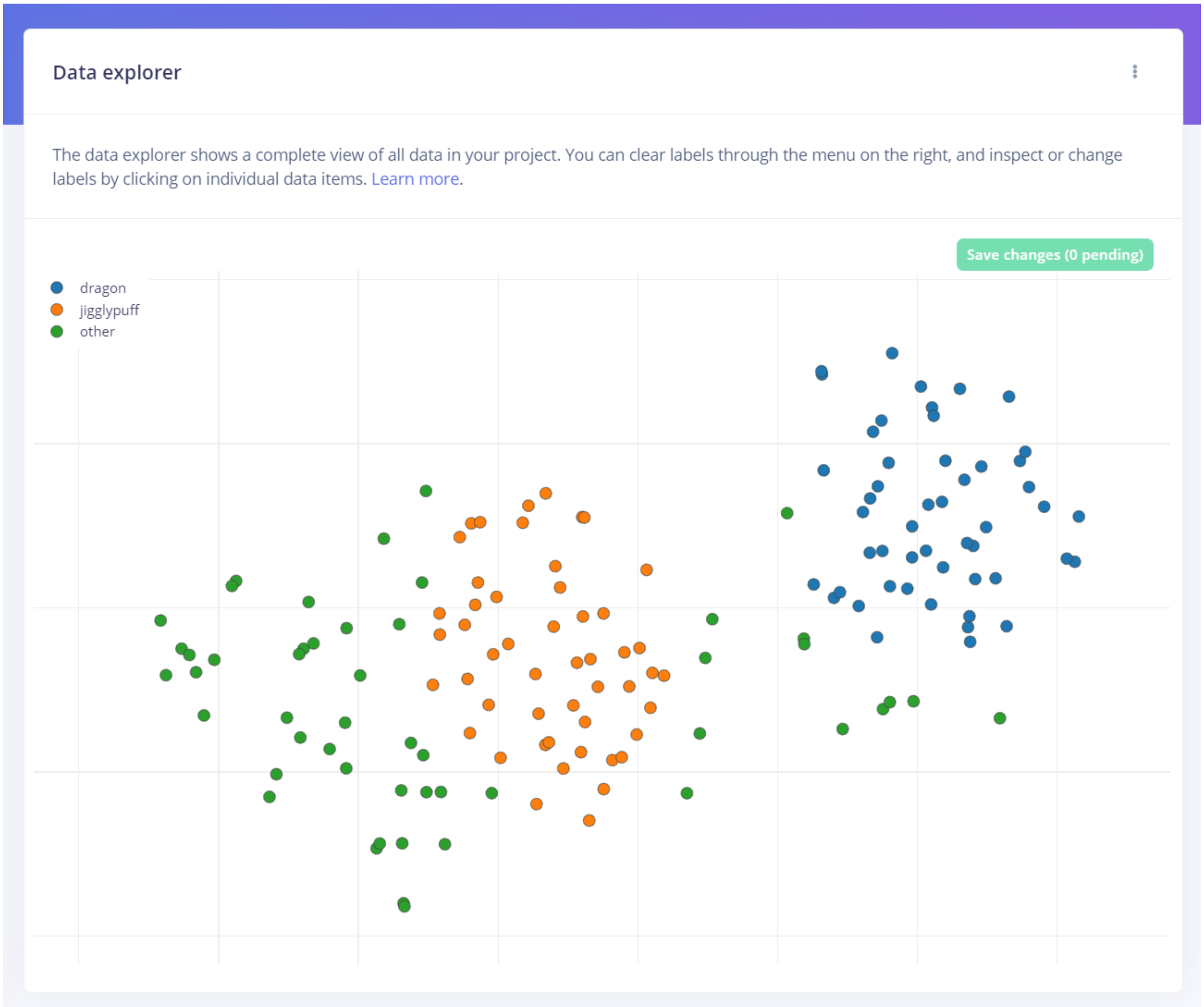
Dragon accuracy: 100%
Dragon precision: 0.91
Dragon recall: 1.00

Jigglypuff accuracy: 77.8%
Jigglypuff precision: 0.88
Jigglypuff recall: 0.78

Other accuracy: 75%
Other precision: 1.00
Other recall: 0.75

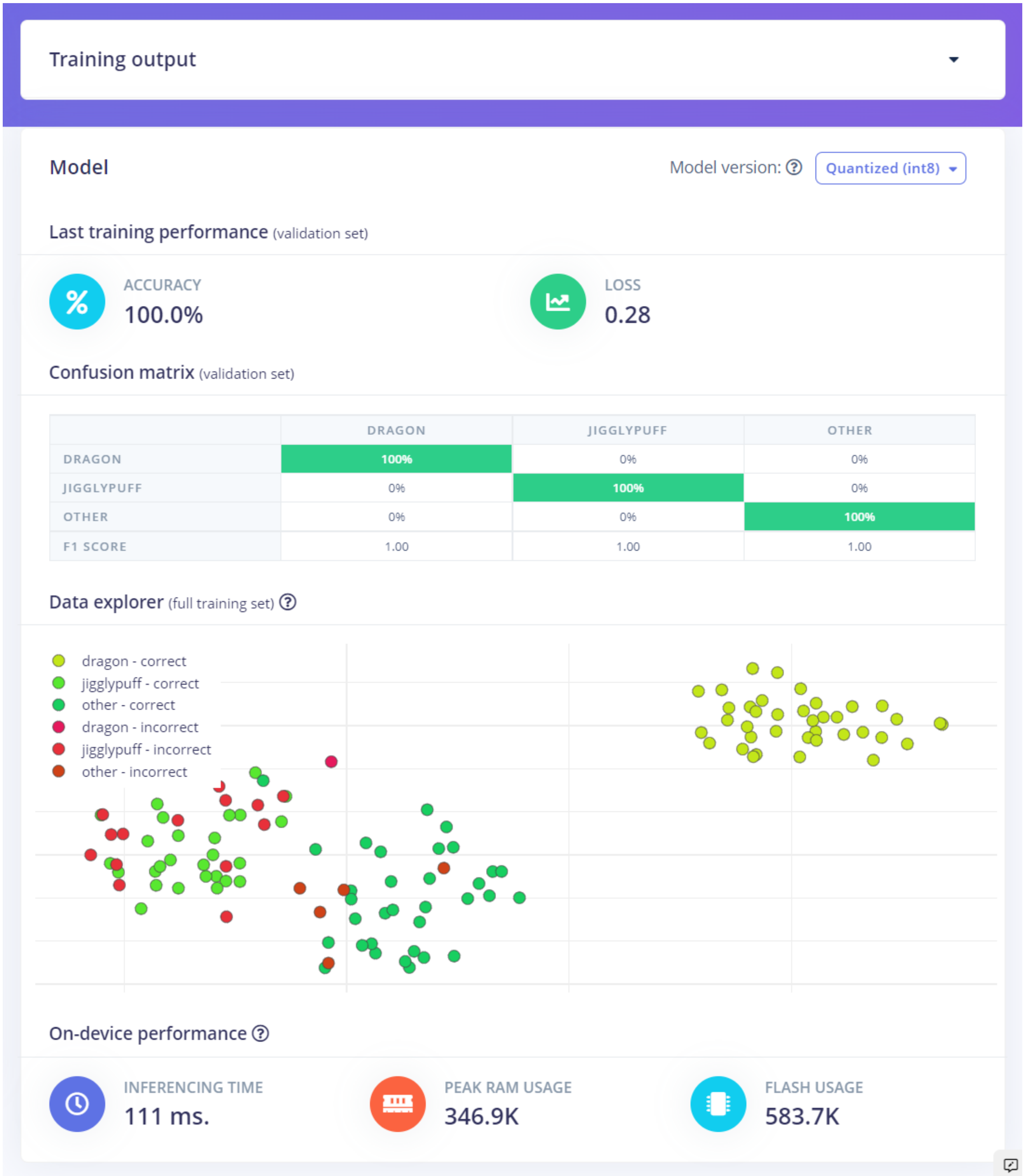
5. Screengrabs of the graphs

Data explorer: pretrained visual model



My dragon (blue) and jigglypuff (orange) data are well clustered. However, objects from “other” (green) are scattered among the jigglypuff and dragon images. This indicates that there is some blending in the object detection clustering between the jigglypuff and other objects.

Training output



The training output shows how the learned model behaves after the training. I should be suspicious of the 100% Accuracy. With an F1 score of 1.00, all labels have a Precision and Recall score of 1.00. My images are being recognized as the same since they are sharing very similar features, which means that I need to add in more noise.

Model testing output

Model testing output

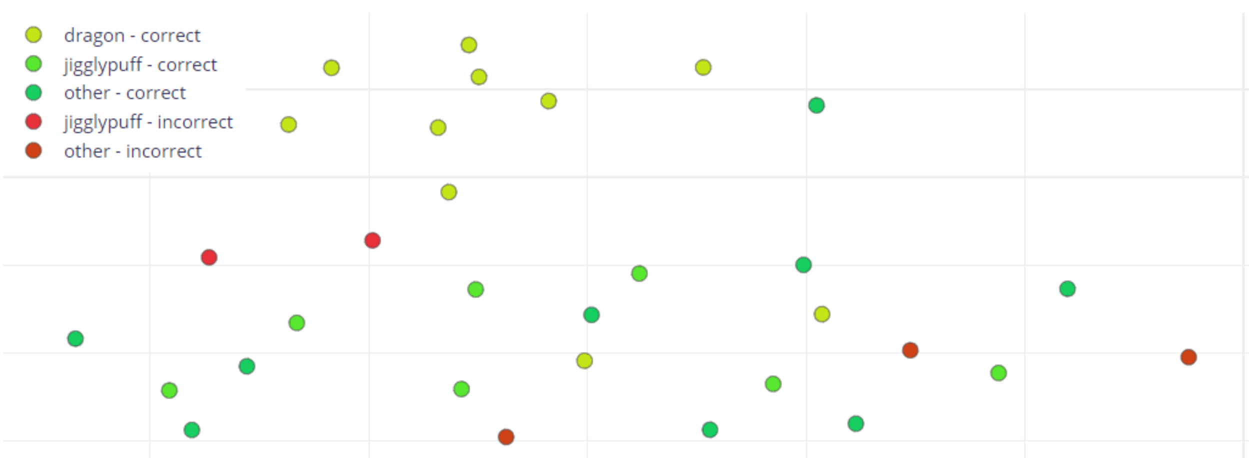
Model testing results

ACCURACY

83.87%

	DRAGON	JIGGLYPUFF	OTHER	UNCERTAIN
DRAGON	100%	0%	0%	0%
JIGGLYPUFF	0%	77.8%	0%	22.2%
OTHER	8.3%	8.3%	75%	8.3%
F1 SCORE	0.95	0.82	0.86	

Feature explorer



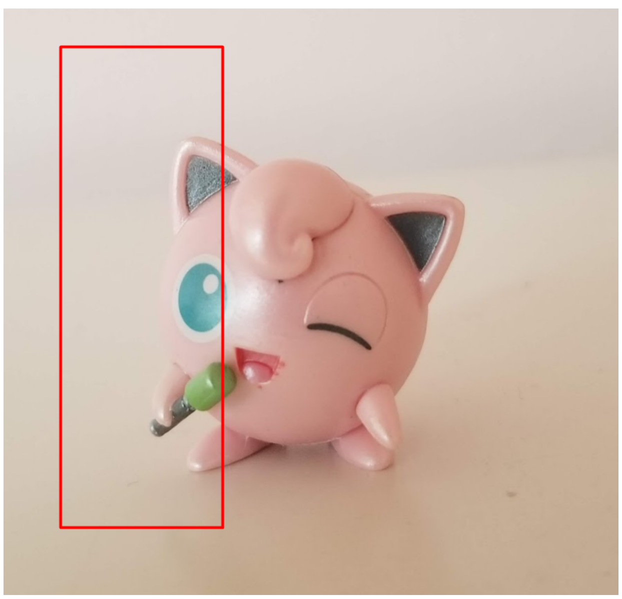
My model has a 83.87% accuracy. For all images of Dragon, I have a 100% accuracy rate. This means that the system was able to classify all Dragon images properly. For Jigglypuff, I have a 77.8% accuracy rate, and for Other, 75%. My F1 scores are high for Dragon (0.95), Jigglypuff (0.82), and Other (0.86).

22.2% of Jigglypuff images were labelled under "Uncertain" which means the system had no idea which label to classify it in. 8.3% of images from "other" were labelled "Uncertain." These uncertain photos indicate that there are not enough images within Dragon, Jigglypuff, and Other that share similar features with those images for the system to confidently recognize and place in a class.

6. **How I could improve my model's performance**

(1) I need to ensure that my objects are clustered properly together and minimize outliers. One way to approach this is to add more noise by having more variation of images in "Others" since a majority of my chosen objects were small and made of plastic; these are overlapping characteristics with my Jigglypuff and Dragon figurines. For example, in the Data explorer graph, I would delete the anomalies (photos corresponding to green dots that fall between the orange and blue dots) and input other images that don't position near the orange and blue dots.

(2) I can also improve my images by choosing a background and lighting that allowed my objects to stand out more. My light pinkish wall color blended in with Jigglypuff and several objects in "other". Here are two examples:




A greater contrast between the background and the images would have allowed the model to recognize features of the object better.

However, with these suggestions, I am introducing biases by providing what I consider to be fixes to the anomalies. As Elio suggested in class, if I encounter such anomalies, a good way to approach this is to continue building the system with them, test, and then see where I can optimize the system. Developing a "better" system is subjective because I am intervening with the original data set.

PART 2: Think of how to integrate this task

In my dreamed-up scenario, Object Detection is used to help people learn how to speak and write words from a language. People can put an object in front of the camera of a device and once it detects the object, it will output its name in a known language (e.g. English) plus its translation in the language you want to learn (e.g. Chinese). Object Detection affords an interactive way to learn a language, where you can directly apply the learned words to physical objects in your environment.

For instance, if I don't know how to say "water bottle" in Chinese, I will hold it up to the camera and once it detects it, it will output the word "water bottle" in English (so that I confirm that it detected a water bottle properly) along with translation of water bottle in Chinese (so that I can learn how to write this word in Chinese characters).

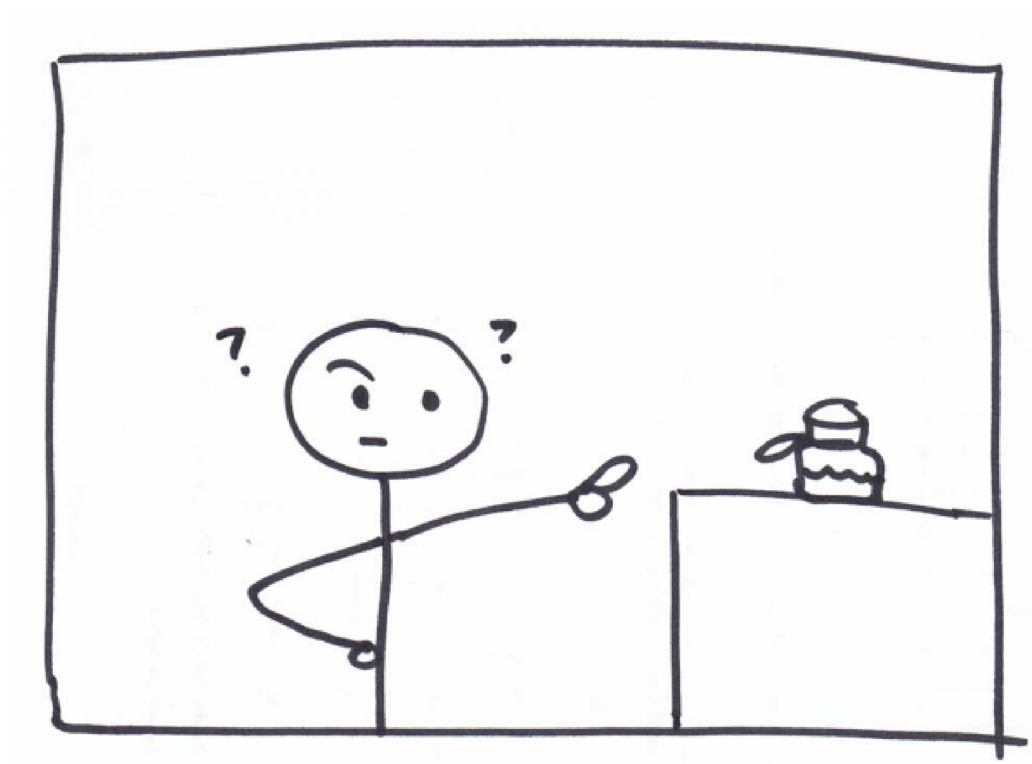
The inspiration behind this came from the frequent times when my mom was unsure how to write a word in Chinese: she would go through the arduous process of asking me to translate the spoken Chinese word in English so that she can then input that English word in Google Translate to get the Chinese characters. This roundabout process is required because it's difficult to search how to write a Chinese character since it is a visual-based language.

Object Detection can be combined with a text-to-speech function that enables the device to also pronounce the word aloud so that the person can not only learn the spelling of the word, but also its pronunciation.

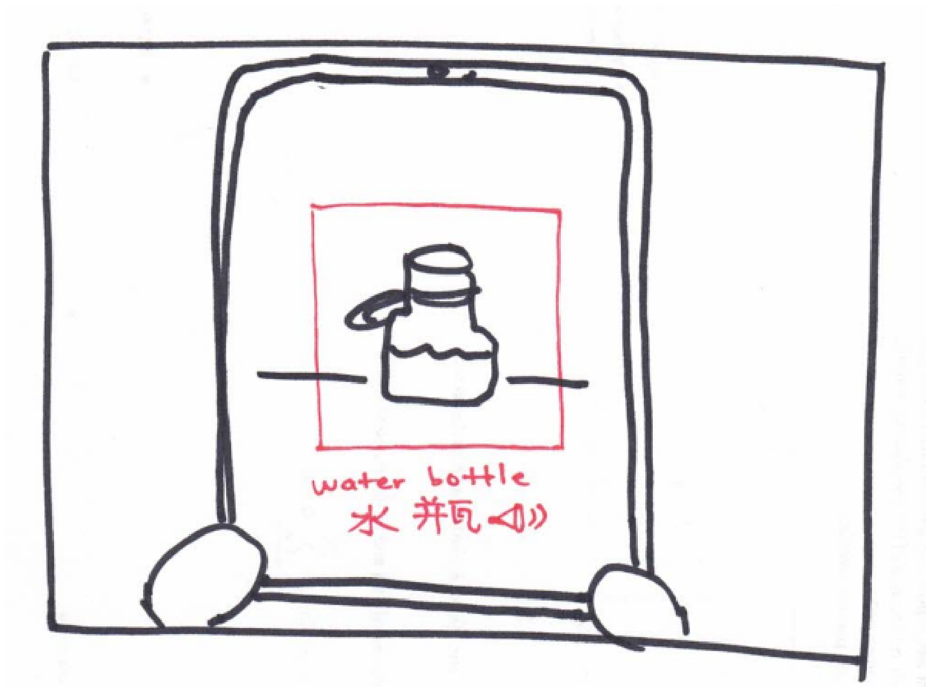
The downside to this scenario is that it may be hard to capture bigger/busier objects (for instance, in a shopping mall where there is a lot of movement, it is hard to zone in on a specific object) and abstract concepts (for instance, a feeling). Object Detection will only be handy for a particular case where an object is isolated enough from its background to be identifiable in the camera.

Storyboard

Mama is texting a message to her daughter in Chinese, asking her daughter “Do you want me to bring your water bottle?”



But Mama forgot how to write the word “water bottle” in Chinese. Now she is stuck.



So Mama pulls out an object detection app that allows her to take a photo of the water bottle. The app detects the bottle and provides (1) the English word so that Mama can check that it detected the correct object, and (2) the Chinese-translated word so that Mama can learn how to write it. A sound icon also allows Mama to hear the pronunciation of the word. This function is helpful if she is learning this word for the first time.

Now Mama learned how to write “water bottle” in Chinese and can send the message to her daughter.