**The problem statement**

When Walmart customers bring their bagged produce to the self-checkout, the kiosk expects them to search for the correct item. It generally takes a while for them to find their item, and even after that they still need to place their produce on the scale to weigh or type in a count. Through personal experience, this is a monotonous and frustrating task.

**Solution**

Using a Computer Vision AI model, we can drastically reduce the amount of time it takes to checkout produce. A classifier, trained on Walmart’s produce inventory, could easily display on-screen the top-5 most likely items being weighed, to which the customer would tap on the correct produce on the kiosk screen. To further speed up the process, the customer could be required to preliminarily place the fruit on the scale for item classification, which would allow the kiosk to weigh the item(s) during inference and could inform the model as to what the item is and how many there are.

**Boundaries of Model capability**

As of now, the training data for the model contains single object photos of fruits and veggies. This restricts the model to performing well only on single objects, not bunches of the same object.

This project assumes there is a camera hiding in the kiosk at the same location as the barcode scanner facing the customer. It was necessary to make this assumption so the model could perform well on a small amount of data.

The time constraint of the project restricted us from gathering more data, so the number of classes that can be identified is 16:

"Bagged banana",

"Bagged broccoli",

"Bagged Fiji apple",

"Bagged granny apple",

"Bagged jalapeno",

“Bagged orange bell pepper",

"Bagged red bell pepper",

“Bagged Roma tomato",

“banana",

"broccoli",

"Fiji apple",

Granny apple",

“jalapeno",

"Orange bell pepper",

"Red bell pepper",

“Roma tomato"

**Robustness of Model**

The model was trained on a variety of produce positions, including half-off (occlusion) photos. So, it can handle partially off-camera items on the kiosk surface.

The speed of classification is lightning fast, roughly .15 seconds.

The size of the model is about 27 megabytes, meaning the model could likely be run locally on the kiosk.

**Future Improvements**

To reduce the volatile nature of the classification in the app, we can introduce a rolling average that activates for one inference period (maybe 1-2 seconds), storing all the past confidence predictions of each class and averaging them.

Test larger models (EfficientNetB1-7)