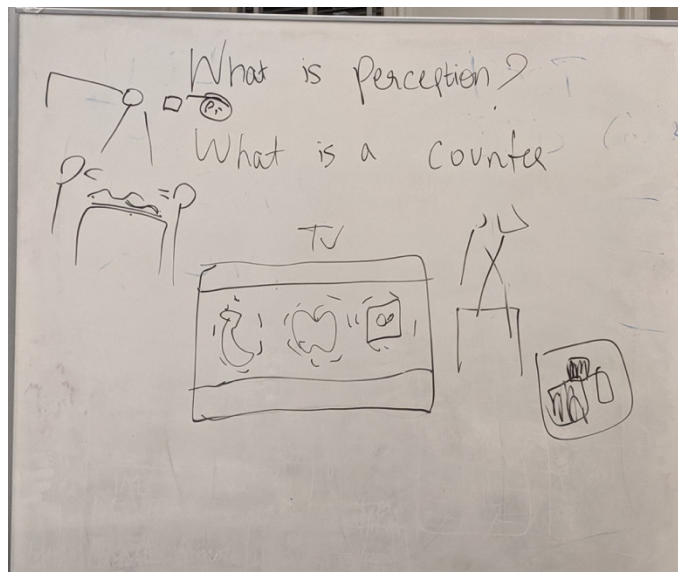


# Design Process Documentation

*Barrak Darweesh, Wesley Deng, Mu-Ti Huang, Aaryaman Sen*

## Generating the Original Idea

We started by asking ourselves a question: What is perception? It is, by psychologist Daniel Schacter, the identification, and interpretation of sensory information in order to represent and understand the presented environment. We envisioned several scenarios and defined our theme being the transformation of people's perception towards an already vested idea and the creation of novel interacting experience.

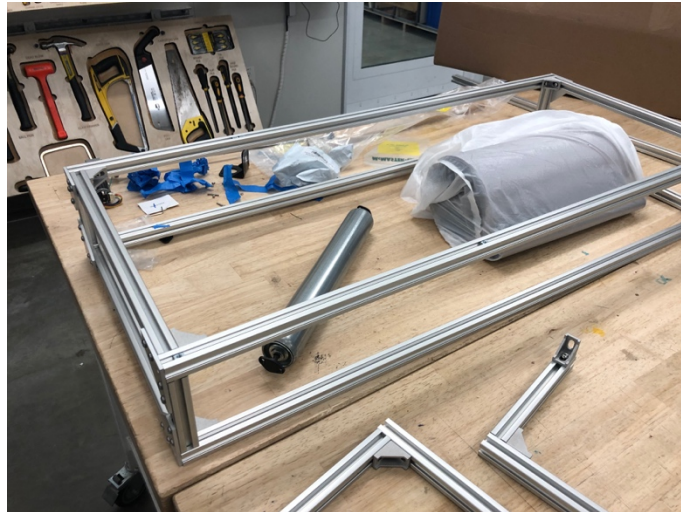


Brainstorm for the first meeting

After several day's observation, we picked the checkout counter at the grocery store as our We aimed to create a socially interactive environment around the checking-out counter that encourage individuals to participate actively. By navigating and integrating assorted technology, our eventual goal is to positively transform people's perception towards checking out from a trivial, mundane errand to a delightful, immersive, and expressive social activity.

## Building the Automatic Counter

One of the main specifications of a grocery checkout counter is the conveyor belt and the semi-automated movement of the objects towards the cashier. The physical prototype design of the express checkout counter aims to create a conveyor belt counter with extra layers of intelligence to create an interactive experience.



Before Assembling

To simplify the construction process in a limited timeframe, 20x20 aluminum extrusion systems are cut into different sizes and assembled to create a rectangular frame. The hardware associated with 20x20 extrusions makes it easier to iterate between ideas and design decisions, varying frame dimensions until the desired design is reached.

A roll of 2mm neoprene is used to create the conveyor belt. The neoprene is then stitched together using an industrial sewing machine, and a strip of canvas that holds the ends together, creating a closed loop.



The DC motor provided with the Kit had to be upgraded to a 12V 55 rpm DC motor for increased torque, that is capable of resisting the neoprene tension. The motor is powered by an external 12V power supply and is engaged using a switch. Using a coupling, the motor is attached to a rubber-coated PVC roller and shaft. Two other bearing rollers are used. The three rollers are inserted into the neoprene loop and are mounted to the frame using laser cut acrylic brackets.

A central aluminum bridge crosses over the neoprene belt, holding the raspberry pi, the camera, an LED strip, and the speaker. When the objects are placed on the belt, the motor switch is turned on to move the object under the bridge. The white LED light turns green when the object is recognized, and the associated object sound is played.

## Sound Effect Generating

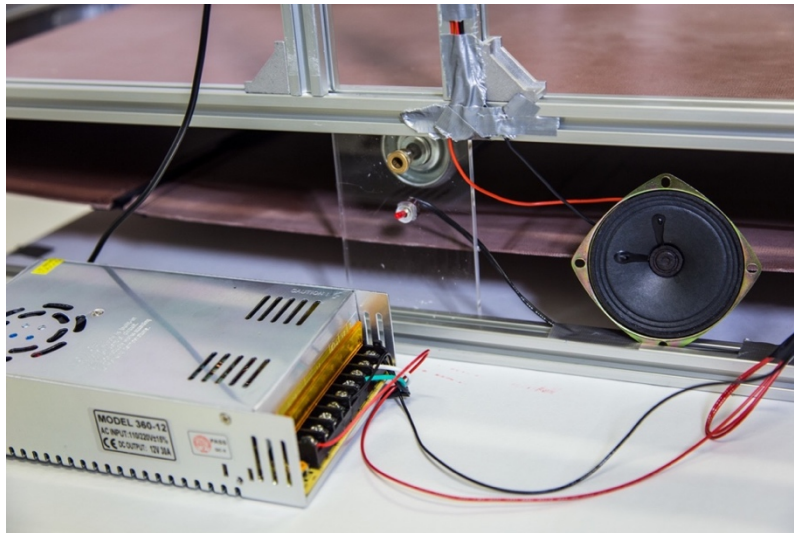
The sound and music we played during the scanning is a crucial part of our project. As a team, we went through several iterations for the sound effect design. Our initial goal was to generate sound according to the pattern of the object. Though it sounds compelling, the object pattern is too ambiguous for Google vision to detect accurately.

We then thought about representing different type of groceries using different instruments and gradually adding them as the objects being scanned, which seems like something a professional music producer would do. Unfortunately, we had limited amount of recourses and background to implement this idea.

What we did for the final demo is to use PyGame creating three channels—one for ambient music that we single out from real music, two for sound effect that creates an ambiguous mapping to the object. Right before the scanning, we randomly select a piece of ambient music and loop it as the background music. During the scanning, every object will be mapped to a pre-processed sound effect that vaguely expresses the attribute of the item. We were using ten channels for the object as we prepared around ten items to be scanned, but the sound overlay was overwhelming after the several items. We eventually eliminated the object channels to two and limited the loop time and created a sink mechanism such that as object sounds were added to channels, the earliest ones were deleted to maintain the number of active channels.

For better sound effects, we use a speaker with a larger front opening that we borrowed from another team. For the final demo, we also include a voice amplifier to achieve a better real-time effect.

From a technical perspective, we navigated several audio processing python libraries. In general, we need a versatile library that can process wav file efficiently. Librosa, a standard library used in natural language processing, came to our mind at the first place. After some trials, we found that, though powerful and user-friendly, Librosa tends to work well on low level signal processing problems, not on manipulating songs and music files as a whole. We also tried PySox and PyAudio.

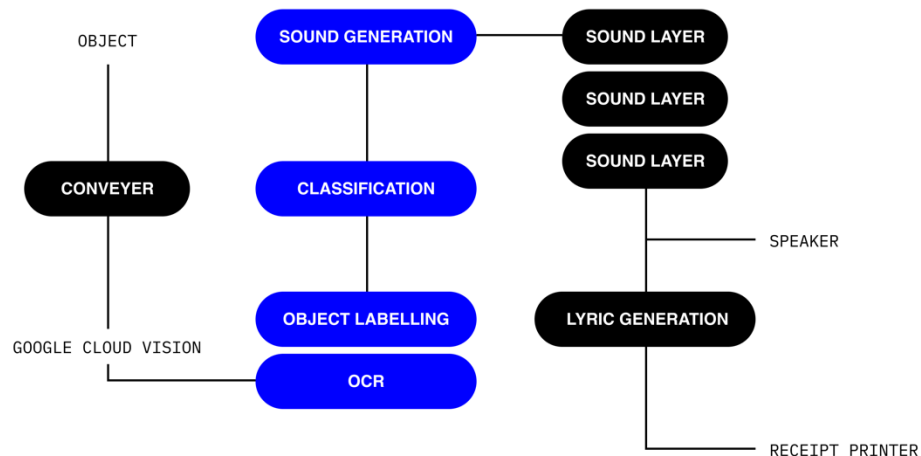


The final library we adopted for almost all the sound effect preprocess is a customized python library called PyDub based on ffmpeg and libav. It contains powerful and elegant commands for concatenating, overlaying and fading in/out music, all of which suit our goal very well.

All the sound files, sound libraries, and the sound generating Jupyter Notebook can be found at <https://github.com/WesleyDeng1016/Express-Counter>.

## Wrestling with Google Cloud Vision

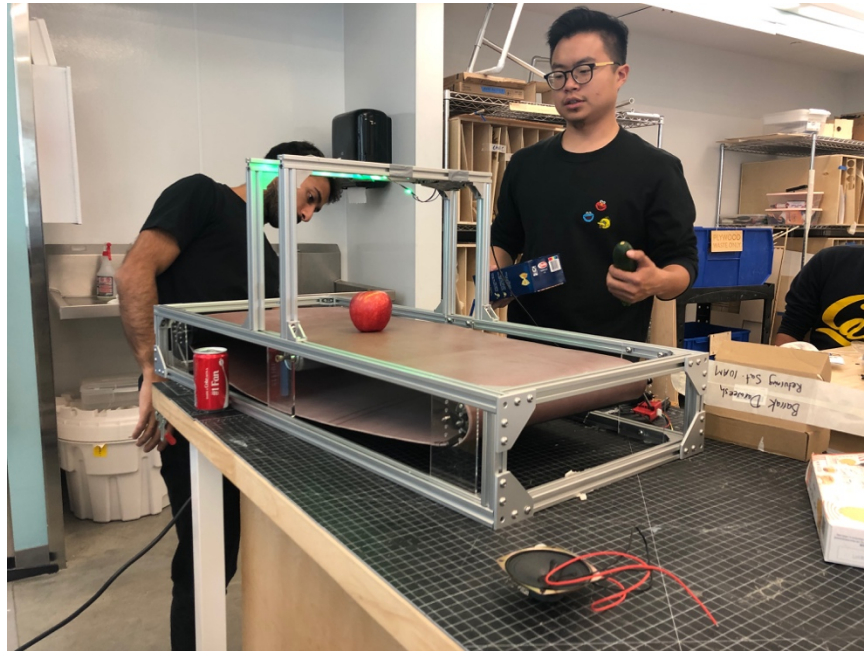
We started with the task of identifying objects in a grocery store. We tried using Google Cloud Vision's image labelling feature. The naive approach of just picking the most likely label the API would generate and then mapping it to the object proved to be quite inaccurate, as several objects that looked similar, such as multiple packaged goods, confused our system. So, we decided to use the sum total of all labels generated. To do this, we created a rudimentary classifier. We decided to use a predefined directory of objects, not unlike an inventory list in a grocery store. For every label we created a class vector that indicated all objects the label could represent. We used a simple addition of the vectors obtained to predict the likeliest object.



Software Architecture Diagram

The above algorithm improved our results significantly, but it still had trouble with some similar looking packaged items. Since a large number of items in the stores we considered would be in packages, we decided to use OCR to read labels in addition to our vision-based labelling.

Using a similar classification method for this, our final pipeline contained an initial OCR scan to see if we could read a package label and if that failed, a subsequent object detection step for items like apples.



Testing Google Cloud Vision

## **Extended devices**

In order to create a compelling demo and illustrate our idea better, we introduced several devices that we could not quickly build up within two weeks.

For the receipt printer, we ordered and compared below three affordable models:

- USB Thermal Receipt Printer TEROW 58mm Small Portable Label Printer
- Wireless Bluetooth 2 inches 58 mm Receipt Thermal Printer
- 2 inches Wireless Bluetooth Receipt Thermal Printer, MUNBYN 58mm Portable Personal Bill Mini Mobile Printer

The USB Thermal Receipt is our first choice while the USB driver could not work properly for Raspberry Pi 3B. The second model is our final demo choice. The receipt printer is connected to the phone and the lyrics/poem we used is currently generated by open source random poem generator.

For the voice amplifier, we ordered and compared below two affordable models:

- Voice Amplifier MAONO AU-C03 Ultralight(0.29 lb) Cardioid Portable Rechargeable Wired Microphone.
- WinBridge Voice Amplifier Portable Rechargeable Wired Headset Microphone Lapel Mic.

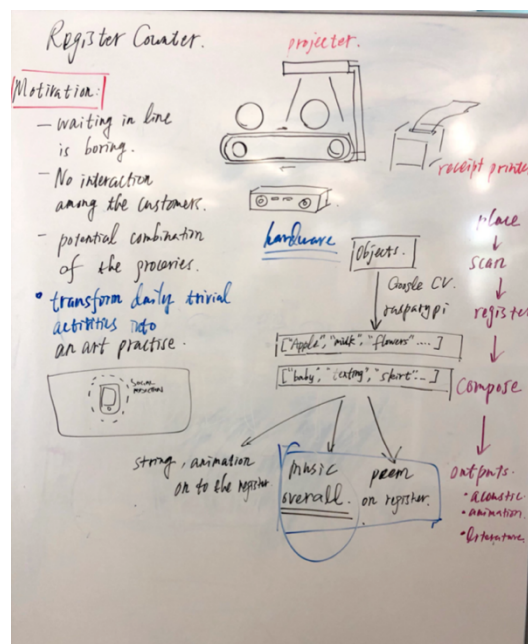
We choose the first model for its unimpressive outlook that could easily be disguised at the back of our automatic checkout machine.

Overall, these extensions enhanced our design's interaction, hence helped convey our design philosophy centered around Raspberry Pi and Google Cloud Vision.



## Additional Design Choices

We had a long debate about whether or not having the customers keep the sound collage file and adding the receipt into our design. On the one hand, we don't want to distract people from the spontaneous, real-time interactive experience. On the other hand, printing the computer-generated poem/lyrics onto the receipt really add some perception-based feedback to an existing counter artifact. We discussed a little bit about adding the sound effect to online shopping as well.



Working Session Brainstorm.

At last, we keep the receipt printer part to mimic the current checking out experience. In addition, we also prepared a designated website for the digital archive of the sound collage and the lyrics to render a well-rounded ecosystem for the customers. The receipt and the website can be treated as souvenir for the expressive and memorable checkout experience.