M&T Bank
Artificial Intelligence
PoC Project

Instructions for use

This documentation is intended for a turnover

In the following slides, the reader should be able to follow the project from the start to its finish. The major pain points, pivots, and reasons for those obstacles will also be clearly described and explained. Tools, open source resources, and forums will also be linked. This includes training resources, youtube videos, and other helpful articles.

You will be able to further contact the author by reference from Dave Hunt or Ashish Vikram

Made by Sefath Chowdhury, Enterprise Architecture & Governance Intern

Hello!

My name is HAL

I am here because I love to give presentations.

You can find me at:

M&T Bank

ATMs, Office use, Consumer Experience

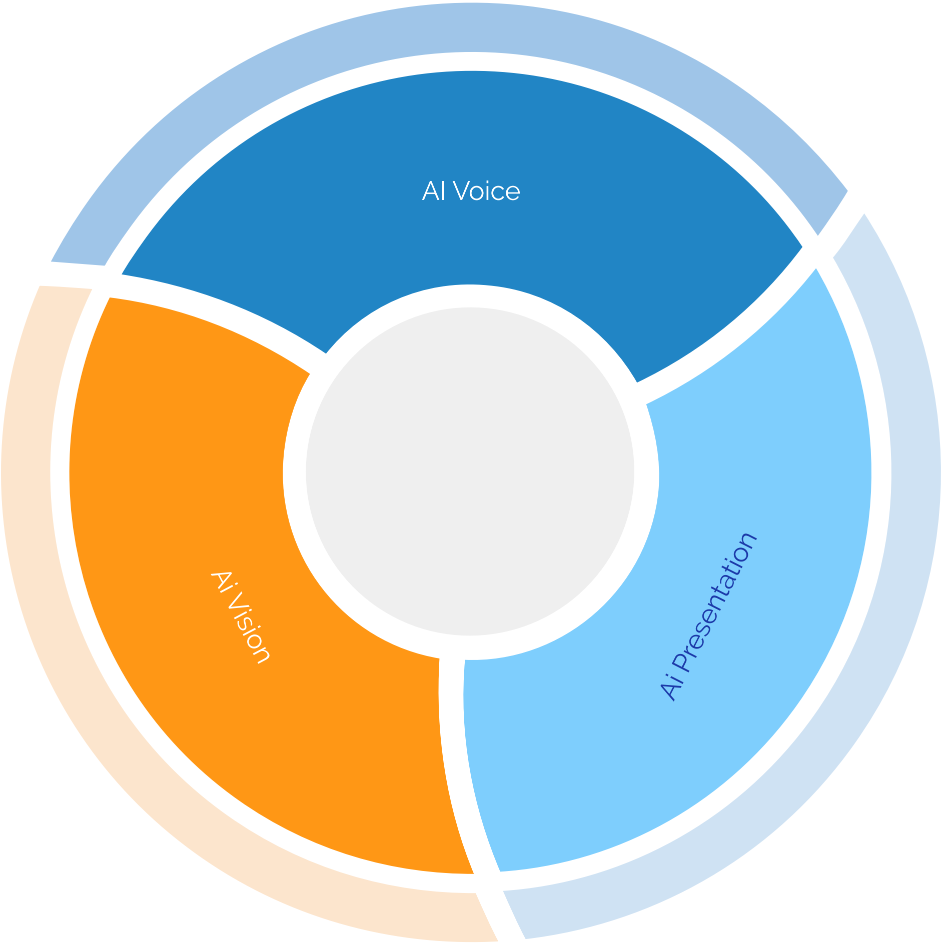
1.

Minimum Viable Product

Let’s start first with our goals for the project

> *Simulate an AI presence in the room, for the board to truly experience the power and depth of this bleeding edge technology*

AI presence can be achieved by



MVP Goals

- AI Vision/Voice that can pause and resume video based on prompts from human presenter
- AI Vision that can recognize and greet people

Google AIY

Google Designed an “artificial-intelligence-yourself” kit that will make our project much easier

AIY?

Here is Google’s clever AI-yourself kits

We have two AIY kits

Vision 1.1

This has all the hardware to experiment with image recognition using neural networks, all to fit in one cleverly designed box

Voice 1.0

This has all the hardware to experiment with voice recognition and assistance using Google APIs, all in one cleverly designed box

Get Ready for Development

You'll need some extra hardware to get your two kits running (we will be using peripherals, not an SSH to hack)

Voice 1.0:

- ▷ Raspberry pi 3
- ▷ Phillips Screwdriver "00"
- ▷ SD Card (8gb preferred)
- ▷ Tape

Vision & Voice peripherals:

- ▷ 5v power cords
- ▷ HDMI to HDMI (voice) & HDMI to HDMI mini (vision)
- ▷ Micro USB hub
- ▷ Monitor, Keyboard, Mouse

Certain Skill Sets Needed

Python (2.7 & 3.0+)

A general knowledge of python is required to develop this project, because most of the AIY scripts are in python

AI/ML

General knowledge of Artificial Intelligence / Machine Learning is also needed. This is to create your own inception models and image recognition models

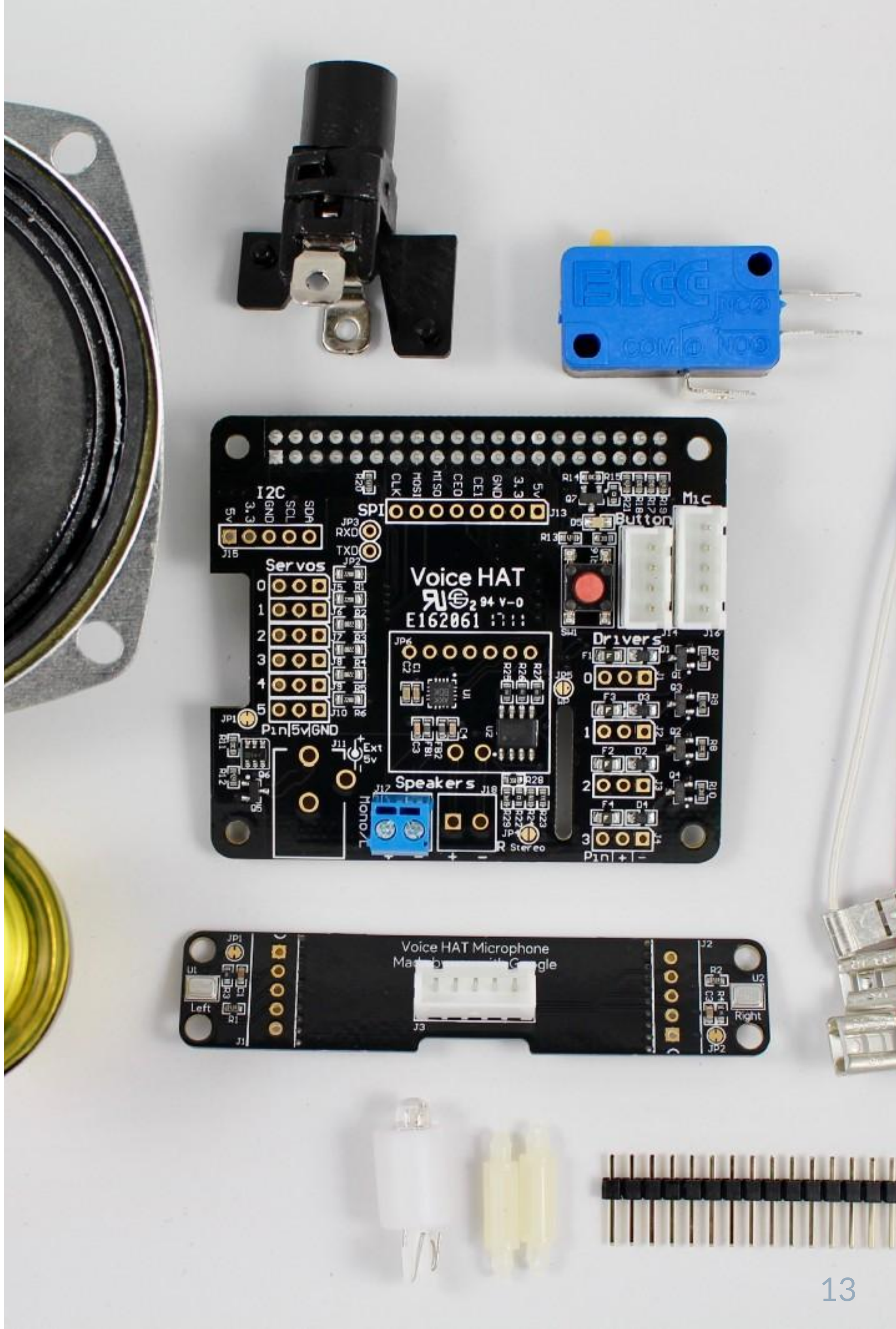
IoT

General knowledge of "internet of things" is also preferred, because you will have to connect hardware without peripherals with each other and have them be able to react to certain events

Get Started with Assembly (Voice)

This is the Instructions for the Voice Kit (Version 1.0)
The latest release is the 2.0

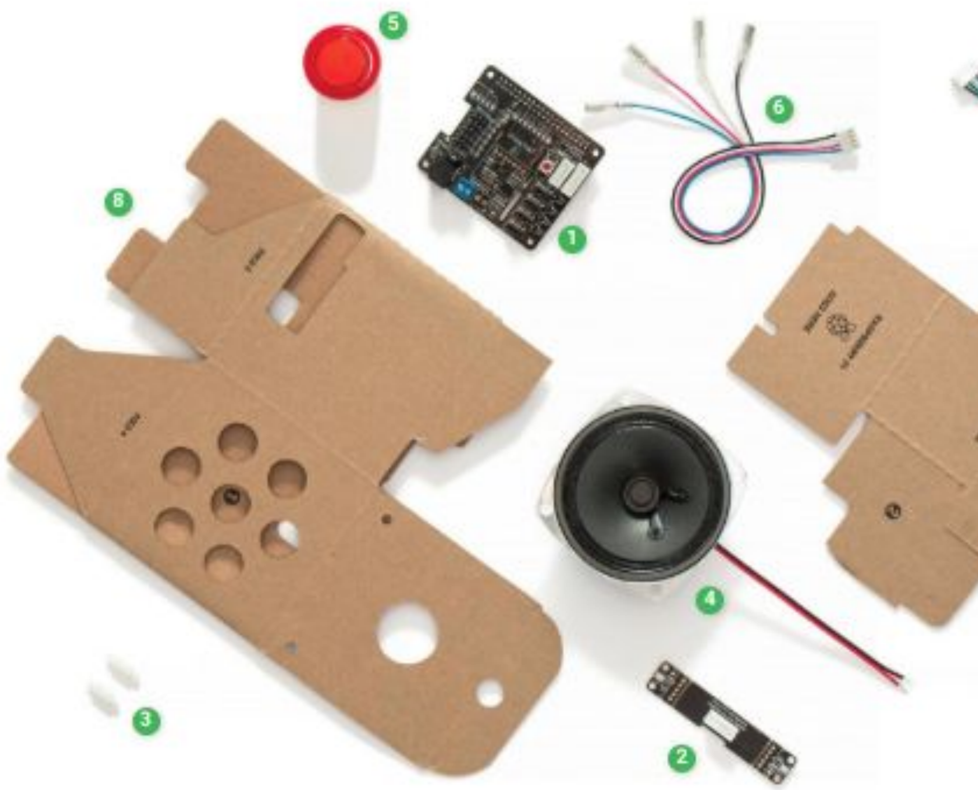
Google’s AIY Projects website has a very well put together instruction set that you can use. Following, is a supplemental video from Hackster.io

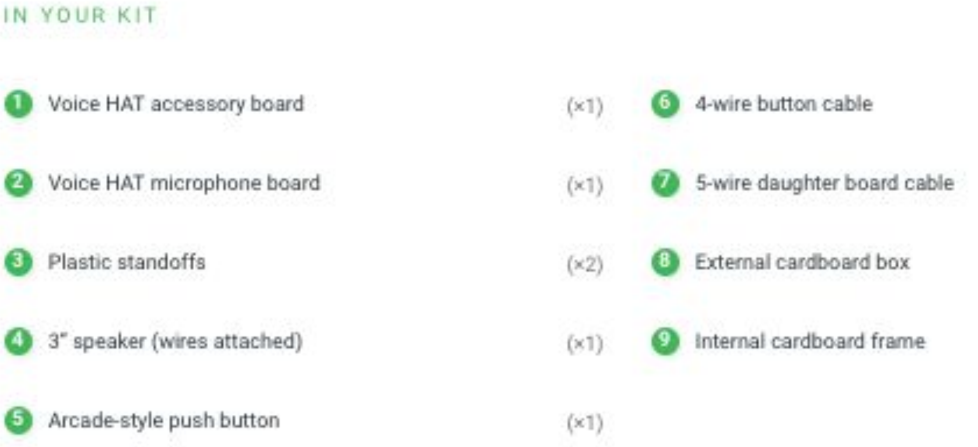


Assembly (Voice)

The assembly instructions for the Voice Kit (Version 1.0) are pretty clear. Follow them step by step and once your done with the assembly, move on to the next slide. The latest release is the 2.0

Google's AIY Projects website instruction set. Supplemental video from Hackster.io





IN YOUR KIT

1	Voice HAT accessory board	(×1)	6	4-wire button cable
2	Voice HAT microphone board	(×1)	7	5-wire daughter board cable
3	Plastic standoffs	(×2)	8	External cardboard box
4	3" speaker (wires attached)	(×1)	9	Internal cardboard frame
5	Arcade-style push button	(×1)		

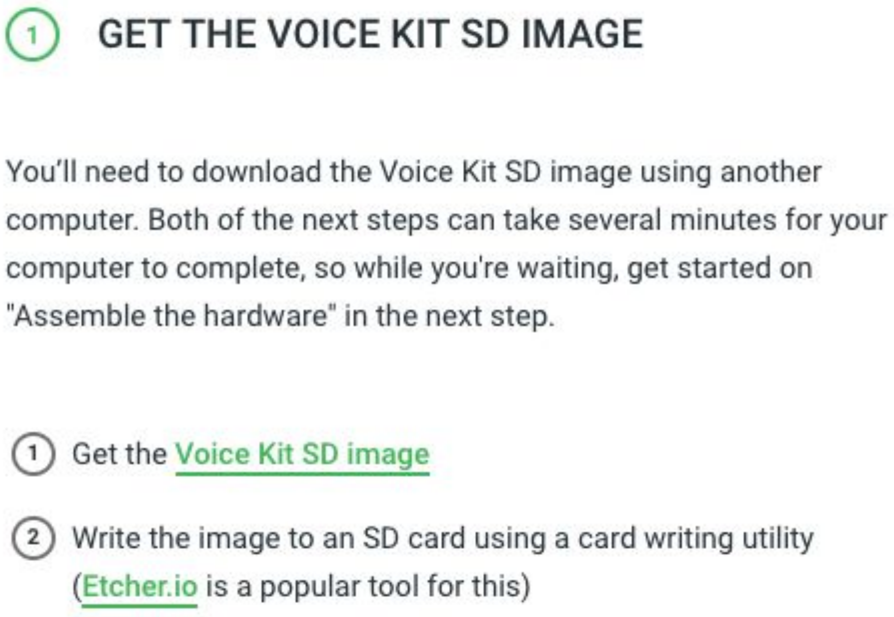
SD Card (Voice)

Install a working image

of raspbian linux, preinstalled with Google AIY's demos.

You can also download the latest images from Google's AIY github repo.

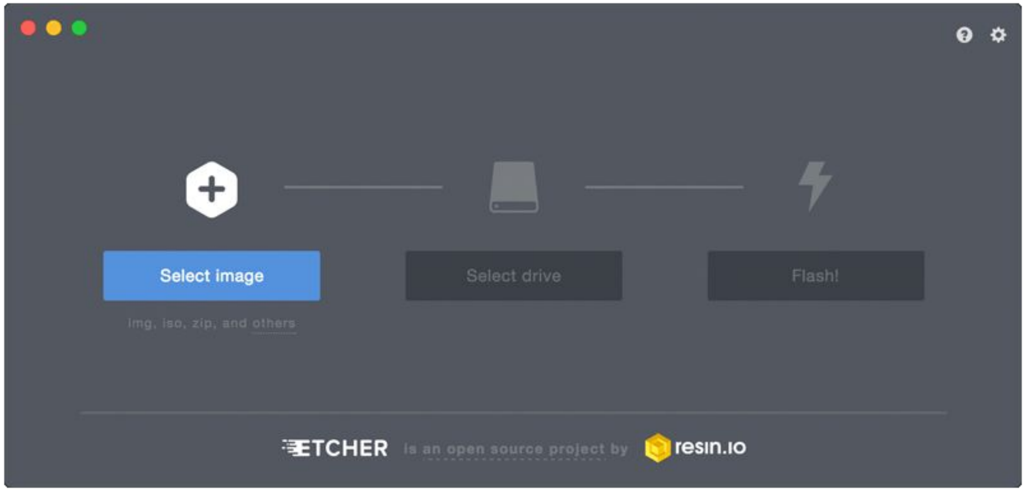
etcher.io makes it easy to flash images onto sd cards.



1 GET THE VOICE KIT SD IMAGE

You'll need to download the Voice Kit SD image using another computer. Both of the next steps can take several minutes for your computer to complete, so while you're waiting, get started on "Assemble the hardware" in the next step.

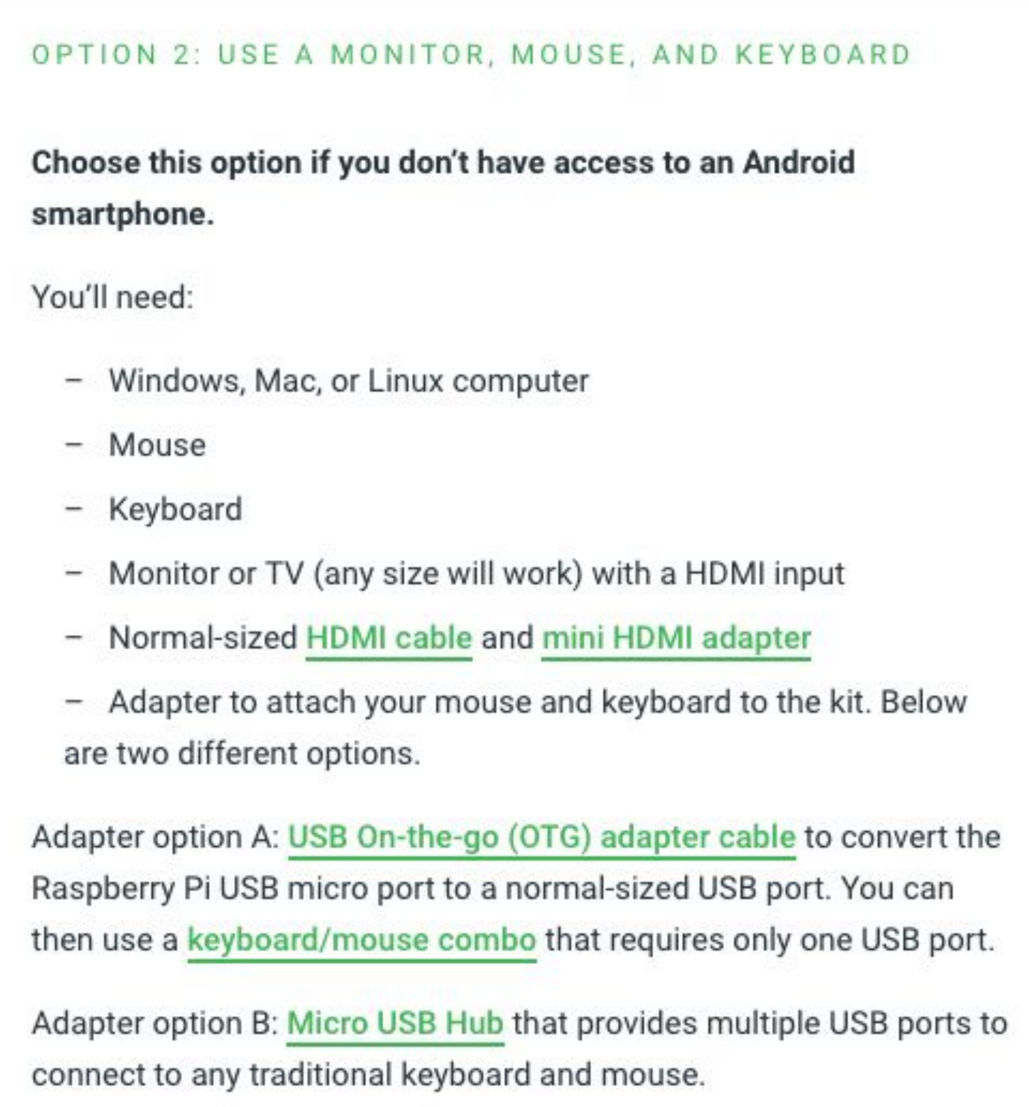
1. Get the Voice Kit SD image
2. Write the image to an SD card using a card writing utility (Etcher.io is a popular tool for this)



Select image
img, iso, zip, and others
Select drive
Flash!
ETCHER is an open source project by resin.io

Peripheral Setup (Voice)

Setup your newly assembled voice kit to a monitor, mouse and keyboard to launch raspbian linux and setup the software side of your kit.



OPTION 2: USE A MONITOR, MOUSE, AND KEYBOARD

Choose this option if you don't have access to an Android smartphone.

You'll need:

- Windows, Mac, or Linux computer
- Mouse
- Keyboard
- Monitor or TV (any size will work) with a HDMI input
- Normal-sized HDMI cable and mini HDMI adapter
- Adapter to attach your mouse and keyboard to the kit. Below are two different options.

Adapter option A: USB On-the-go (OTG) adapter cable to convert the Raspberry Pi USB micro port to a normal-sized USB port. You can then use a keyboard/mouse combo that requires only one USB port.

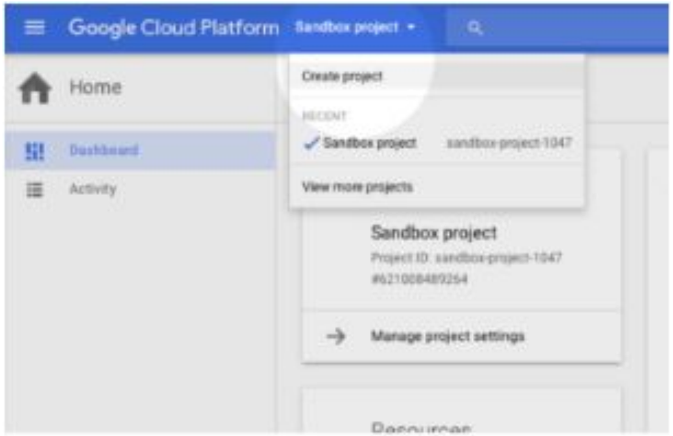
Adapter option B: Micro USB Hub that provides multiple USB ports to connect to any traditional keyboard and mouse.

API Setup (Voice)

Google supplies many API’s through Google Cloud Platform. The two APIs we want are the Google assistant and cloud speech API.

Use Google's instructions for setting up credentials and downloading the JSON Authentication Key on your local Raspbian machine

Here is another forum that describes the process very well (Scroll to Google API subheading)


Google Cloud Platform · Sandbox project

Home · Dashboard · Activity

Create project

RECENT

✓ Sandbox project · sandbox-project-1047

View more projects

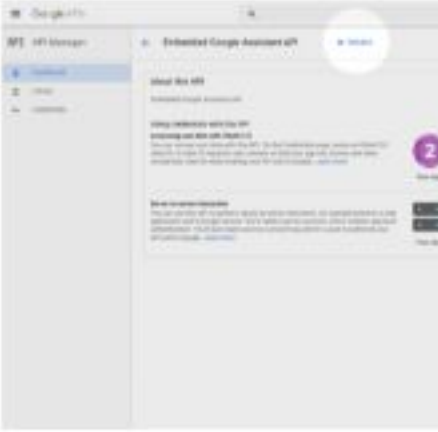
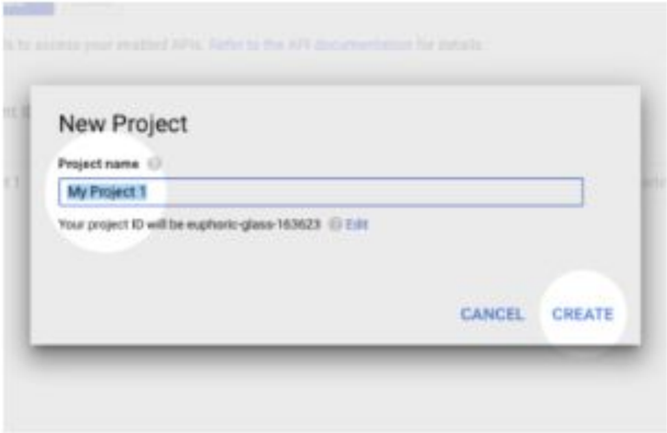
Sandbox project

Project ID: sandbox-project-1047

#621008489264

Manage project settings

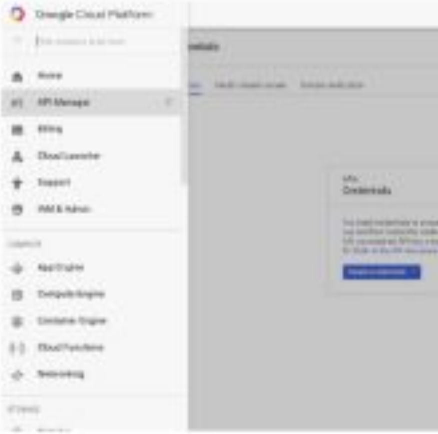
Resources



New Project

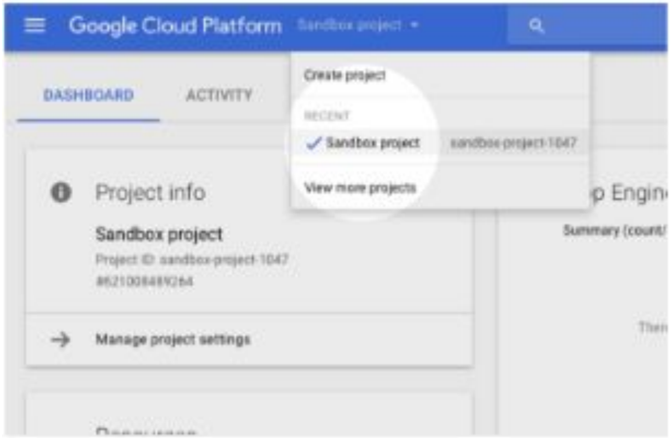
Project name: My Project 1

Your project ID will be euphoric-glass-163623 Edit

CANCEL · CREATE


Google Cloud Platform

Credentials


Google Cloud Platform · Sandbox project

DASHBOARD · ACTIVITY

Create project

RECENT

✓ Sandbox project · sandbox-project-1047

View more projects

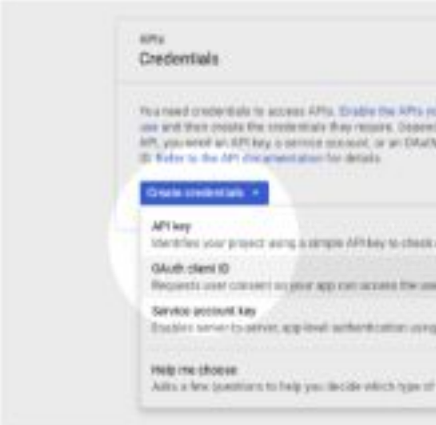
Project info

Sandbox project

Project ID: sandbox-project-1047

#621008489264

Manage project settings


APIs

Credentials

Create credentials

API key

OAuth client ID

Service account key

Help me choose

Demos (Voice)

There are three main demos that the voice kit comes with.

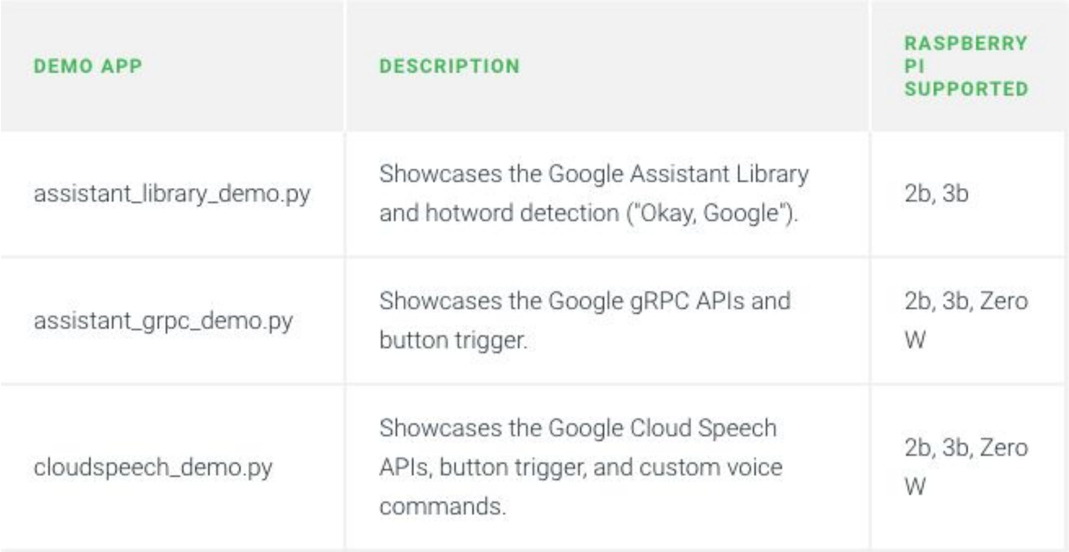
+ assistant _library_demo.py
+ assistant_grpc_demo.py
+ cloudspeech_demo.py

You can view all the newest demos on Google's AIY github repo.

With the cloudspeech API, you can create your own demo, with your own local commands.

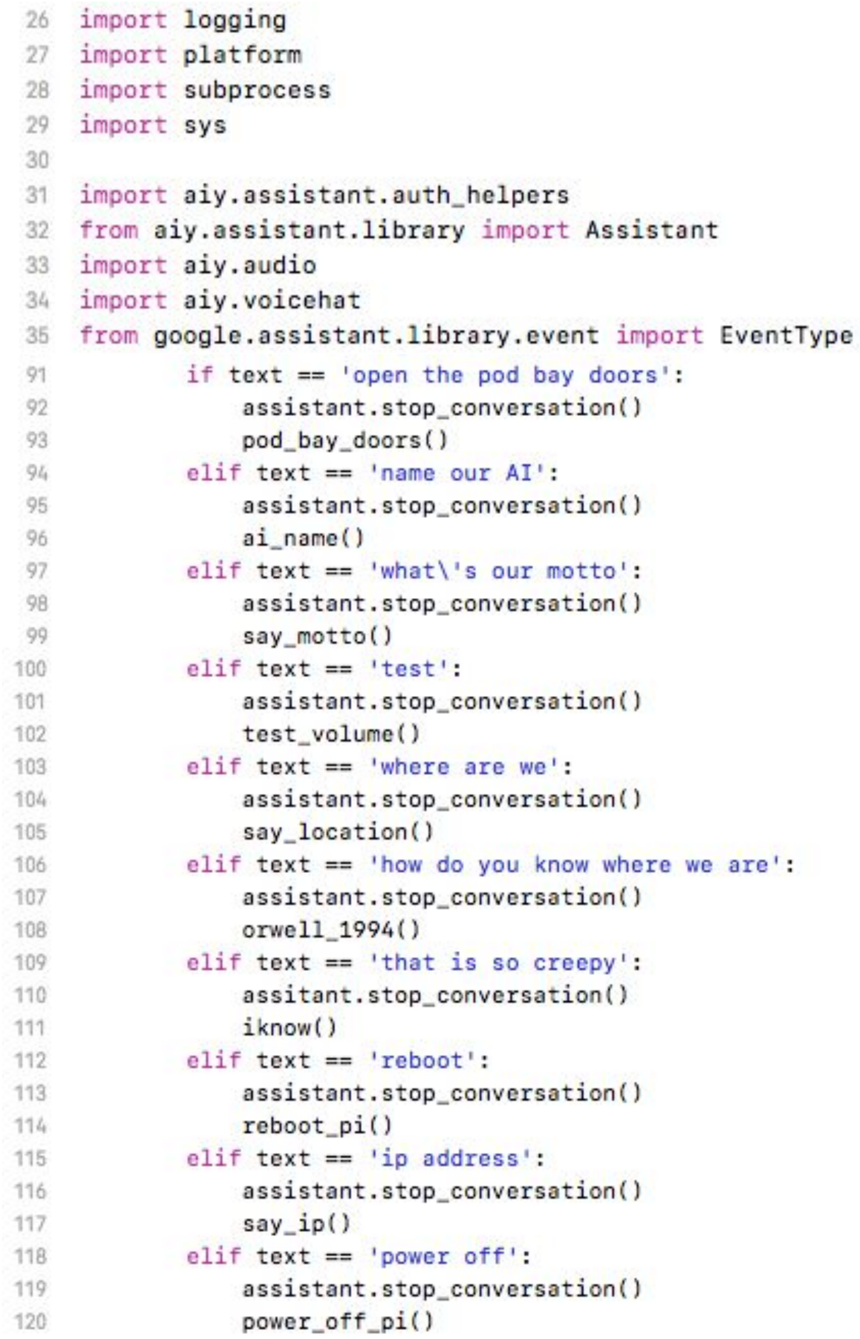
`Command Line Prompts`

```
src/examples/voice/assistant _library_demo.py
src/examples/voice/assistant_grpc_demo.py
src/examples/voice/cloudspeech_demo.py
```



DEMO APP	DESCRIPTION	RASPBERRY PI SUPPORTED
assistant_library_demo.py	Showcases the Google Assistant Library and hotword detection ("Okay, Google").	2b, 3b
assistant_grpc_demo.py	Showcases the Google gRPC APIs and button trigger.	2b, 3b, Zero W
cloudspeech_demo.py	Showcases the Google Cloud Speech APIs, button trigger, and custom voice commands.	2b, 3b, Zero W

Example Demo (Voice)



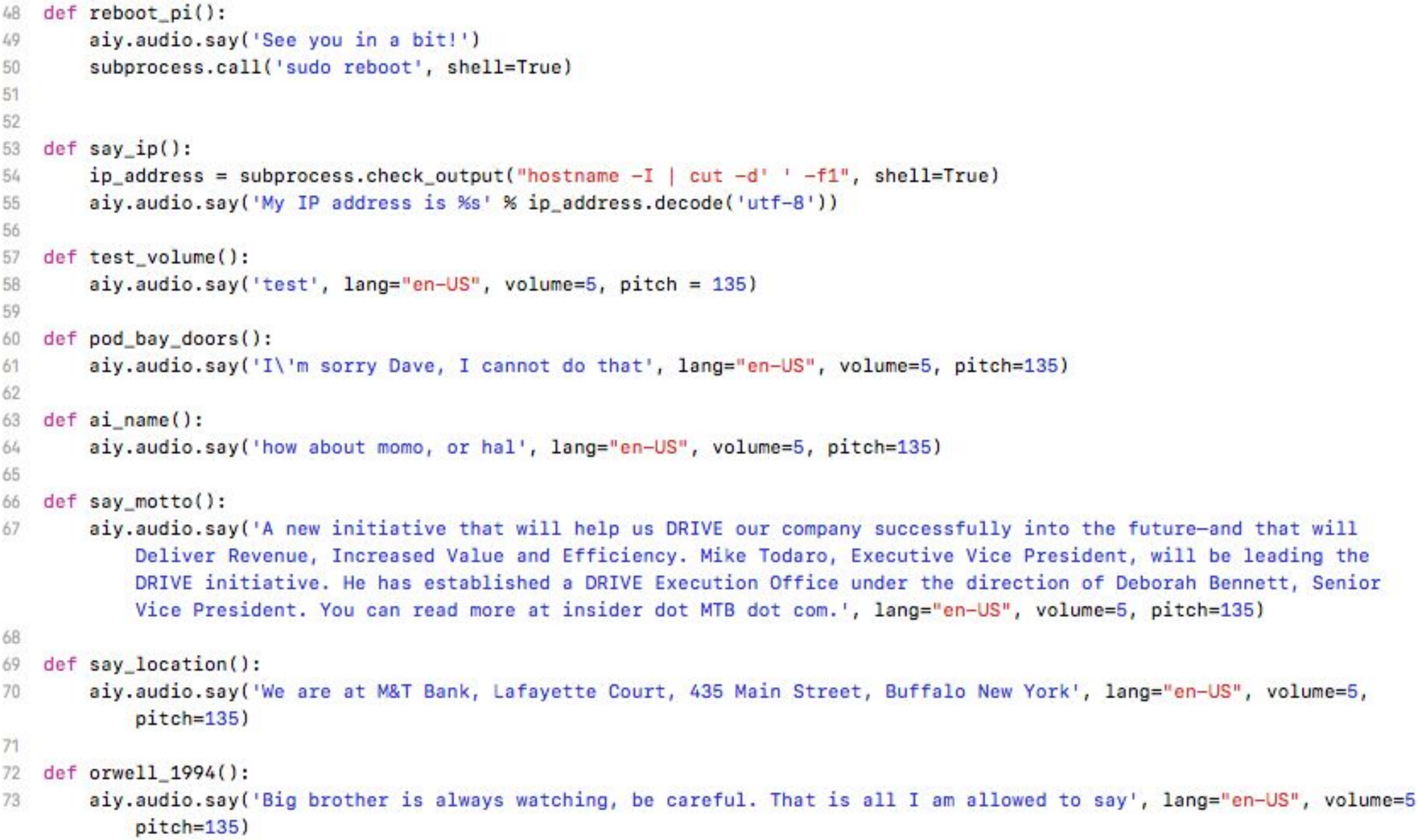
```
import logging
import platform
import subprocess
import sys

import aiy.assistant.auth_helpers
from aiy.assistant.library import Assistant
import aiy.audio
import aiy.voicehat
from google.assistant.library.event import EventType
```

```
        if text == 'open the pod bay doors':
            assistant.stop_conversation()
            pod_bay_doors()
        elif text == 'name our AI':
            assistant.stop_conversation()
            ai_name()
        elif text == 'what\'s our motto':
            assistant.stop_conversation()
            say_motto()
        elif text == 'test':
            assistant.stop_conversation()
            test_volume()
        elif text == 'where are we':
            assistant.stop_conversation()
            say_location()
        elif text == 'how do you know where we are':
            assistant.stop_conversation()
            orwell_1994()
        elif text == 'that is so creepy':
            assitant.stop_conversation()
            iknow()
        elif text == 'reboot':
            assistant.stop_conversation()
            reboot_pi()
        elif text == 'ip address':
            assistant.stop_conversation()
            say_ip()
        elif text == 'power off':
            assistant.stop_conversation()
            power_off_pi()
```

With a couple import statements, you can freely use the cloudspeech api to create your own local commands.

The actual code can be viewed on intralink

Example cont’d

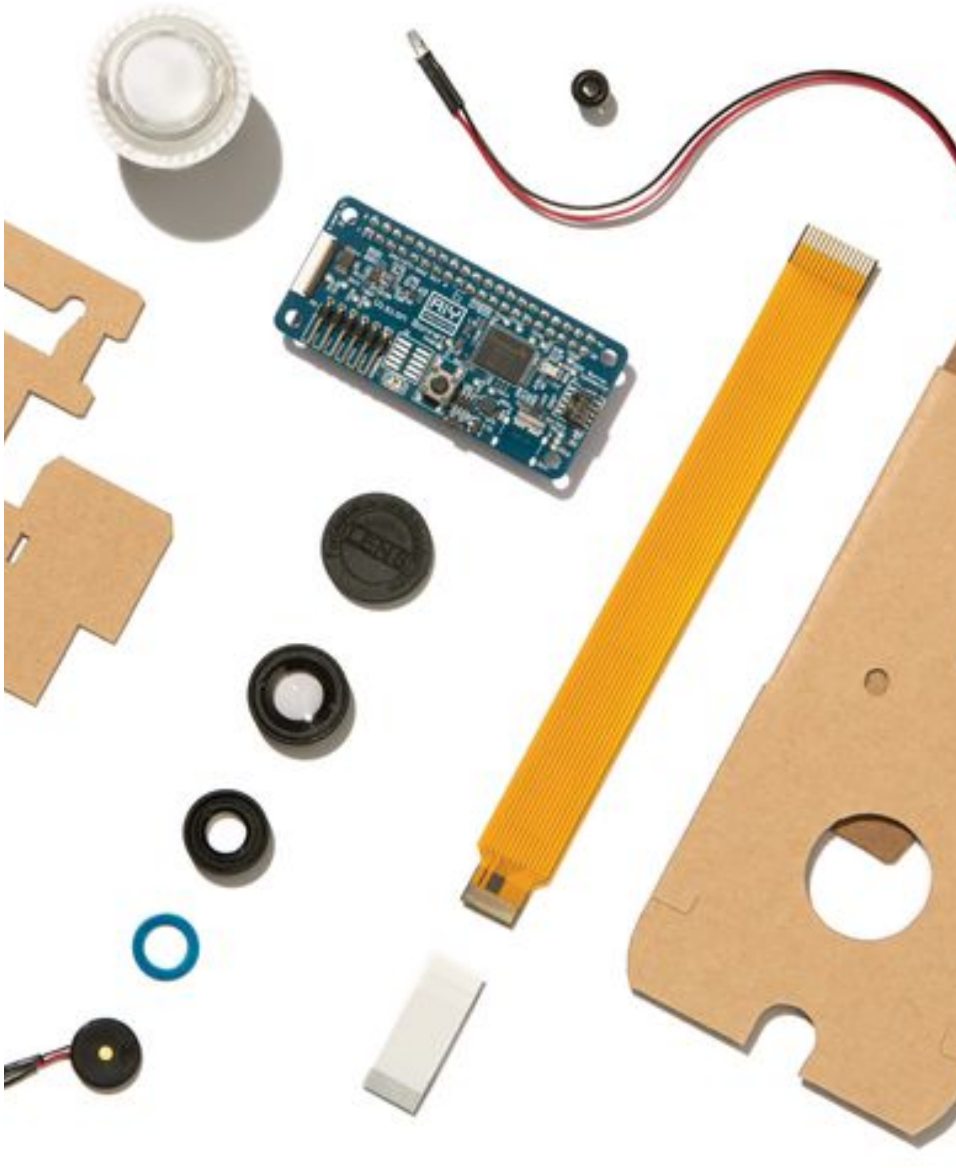


```
def reboot_pi():
    aiy.audio.say('See you in a bit!')
    subprocess.call('sudo reboot', shell=True)


def say_ip():
    ip_address = subprocess.check_output("hostname -I | cut -d' ' -f1", shell=True)
    aiy.audio.say('My IP address is %s' % ip_address.decode('utf-8'))

def test_volume():
    aiy.audio.say('test', lang="en-US", volume=5, pitch = 135)

def pod_bay_doors():
    aiy.audio.say('I\'m sorry Dave, I cannot do that', lang="en-US", volume=5, pitch=135)

def ai_name():
    aiy.audio.say('how about momo, or hal', lang="en-US", volume=5, pitch=135)

def say_motto():
    aiy.audio.say('A new initiative that will help us DRIVE our company successfully into the future—and that will
        Deliver Revenue, Increased Value and Efficiency. Mike Todaro, Executive Vice President, will be leading the
        DRIVE initiative. He has established a DRIVE Execution Office under the direction of Deborah Bennett, Senior
        Vice President. You can read more at insider dot MTB dot com.', lang="en-US", volume=5, pitch=135)

def say_location():
    aiy.audio.say('We are at M&T Bank, Lafayette Court, 435 Main Street, Buffalo New York', lang="en-US", volume=5,
        pitch=135)

def orwell_1994():
    aiy.audio.say('Big brother is always watching, be careful. That is all I am allowed to say', lang="en-US", volume=5
        pitch=135)
```

Get Started with Assembly (Vision)

This is the Instructions for the Vision Kit (Version 1.1)
The latest release is the 1.1

Google's AIY Projects website has a very well put together instruction set that you can use. Following, is a supplemental video from Hackster.io



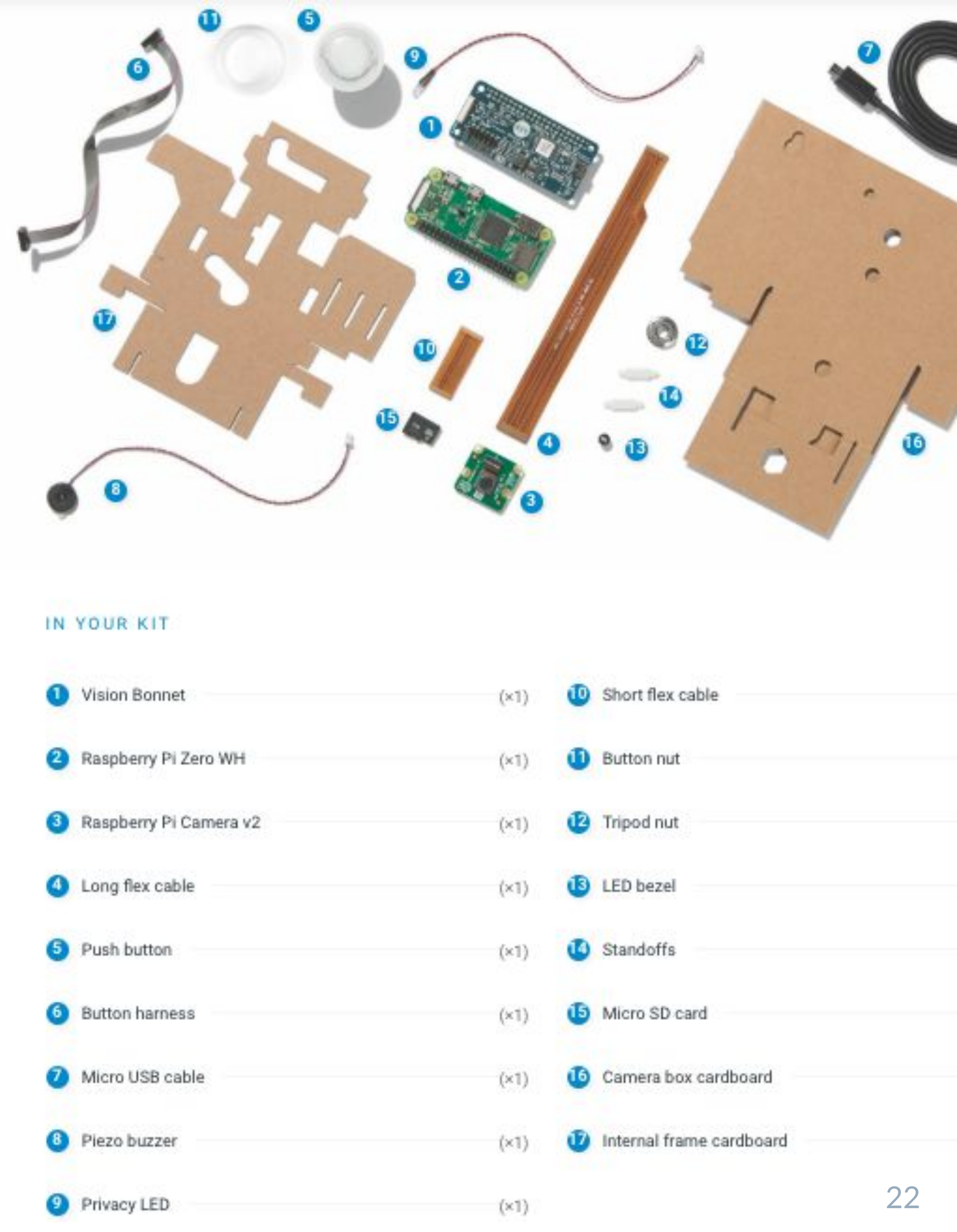
Assembly (Vision)

The assembly instructions for the Vision Kit (Version 1.1) are pretty clear. Follow them step by step and once your done with the assembly, move on to the next slide.

The latest release is the 1.1

Google's AIY Projects website instruction set.

Supplemental video from Hackster.io



IN YOUR KIT

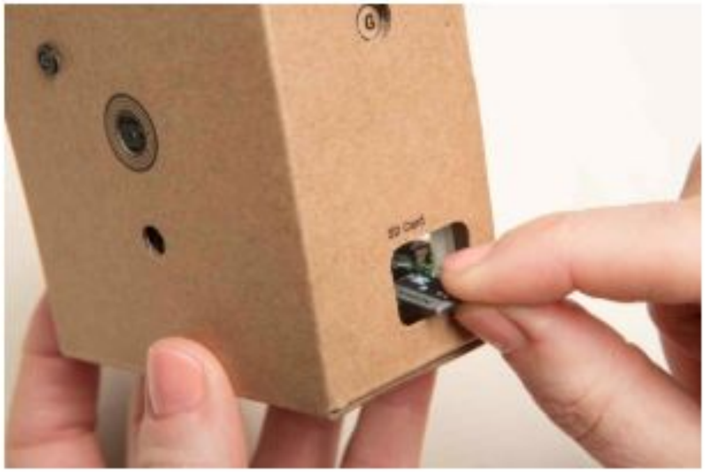
1	Vision Bonnet	(×1)
2	Raspberry Pi Zero WH	(×1)
3	Raspberry Pi Camera v2	(×1)
4	Long flex cable	(×1)
5	Push button	(×1)
6	Button harness	(×1)
7	Micro USB cable	(×1)
8	Piezo buzzer	(×1)
9	Privacy LED	(×1)
10	Short flex cable	
11	Button nut	
12	Tripod nut	
13	LED bezel	
14	Standoffs	
15	Micro SD card	
16	Camera box cardboard	
17	Internal frame cardboard	

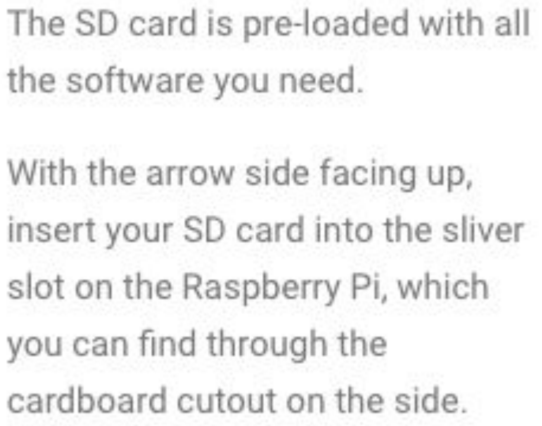
SD Card (Vision)

The Vison Kit version 1.1.
Comes with a working image

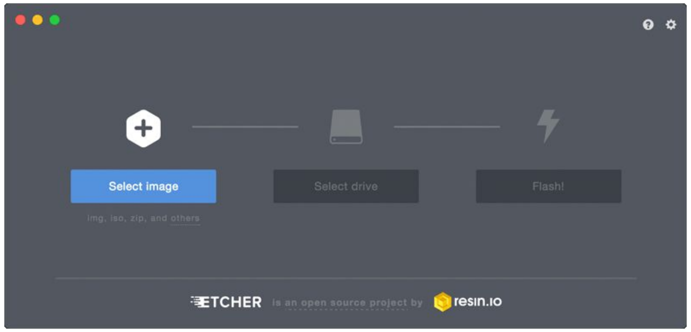
of raspbian linux,
preinstalled with Google
AIY’s demos.
You can also download the
latest images from Google's
AIY github repo.

etcher.io makes it easy to
flash images onto sd cards.



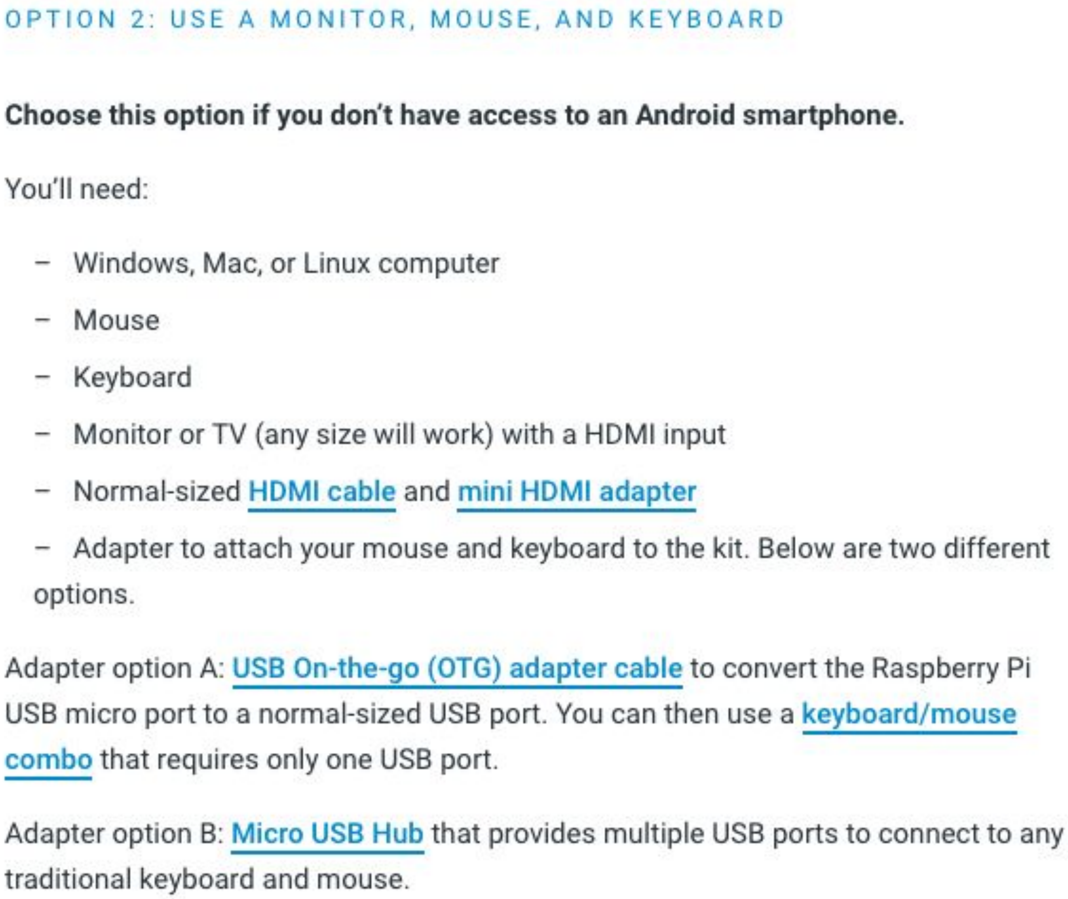

The SD card is pre-loaded with all the software you need.

With the arrow side facing up, insert your SD card into the sliver slot on the Raspberry Pi, which you can find through the cardboard cutout on the side.


Select image
img, iso, zip, and others
Select drive
Flash!
ETCHER is an open source project by resin.io

Peripheral Setup (Voice)

Setup your newly assembled voice kit to a monitor, mouse and keyboard to launch raspbian linux and setup the software side of your kit.



OPTION 2: USE A MONITOR, MOUSE, AND KEYBOARD

Choose this option if you don't have access to an Android smartphone.

You'll need:

- Windows, Mac, or Linux computer
- Mouse
- Keyboard
- Monitor or TV (any size will work) with a HDMI input
- Normal-sized HDMI cable and mini HDMI adapter
- Adapter to attach your mouse and keyboard to the kit. Below are two different options.

Adapter option A: USB On-the-go (OTG) adapter cable to convert the Raspberry Pi USB micro port to a normal-sized USB port. You can then use a keyboard/mouse combo that requires only one USB port.

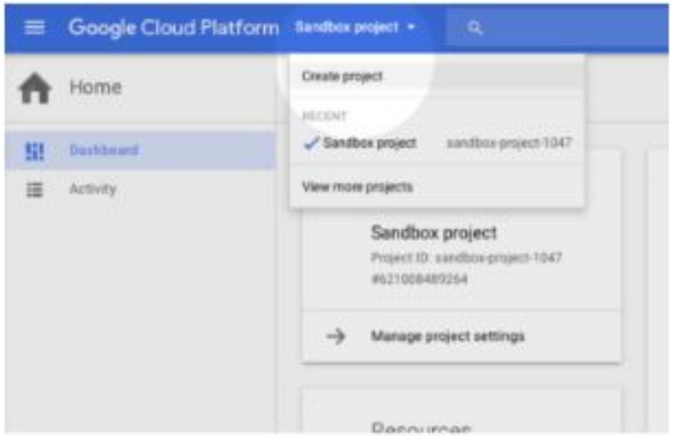
Adapter option B: Micro USB Hub that provides multiple USB ports to connect to any traditional keyboard and mouse.

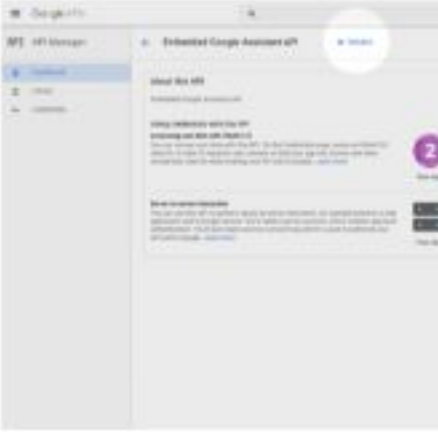
API Setup (Voice)

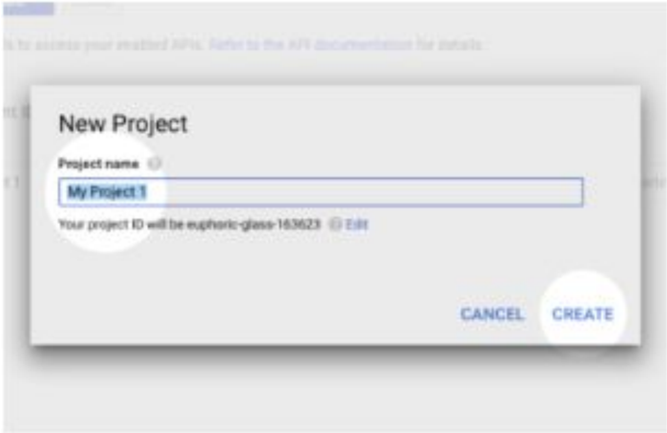
Google supplies many API’s through Google Cloud Platform. The two APIs we want are the Google assistant and cloud speech API.

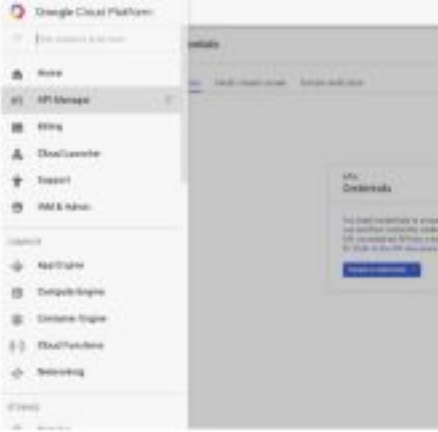
Use Google's instructions for setting up credentials and downloading the JSON Authentication Key on your local Raspbian machine

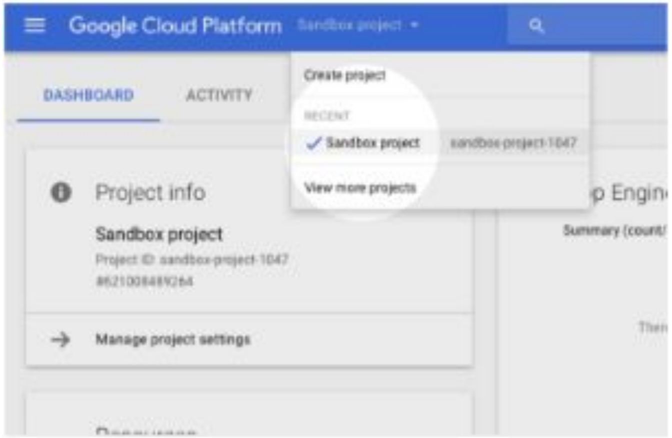
Here is another forum that describes the process very well (Scroll to Google API subheading)

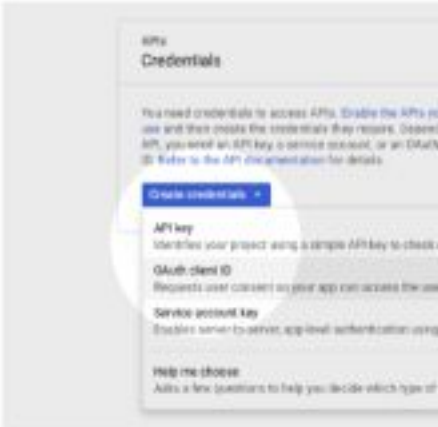












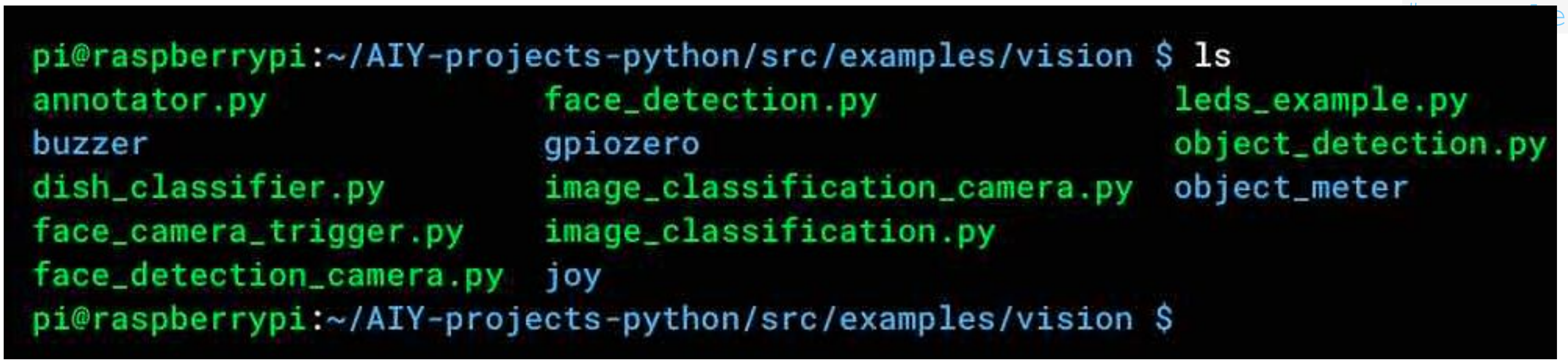
Demos (Vision)

The Vision kit runs the Joy detector demo right out the box. Using a confidence score, it will light up the button on a scale of white to blue, indicating happy to sad.

You can view all the newest demos on [Google's AIY github repo](#).

Command Line Prompts

```
sudo systemctl stop joy_detection_demo
#Stop the joy detection demo
cd
~/AIY-projects-python/src/examples/vision
#See all the demos to choose from
###Photo below
./image_classification_camera.py
./face_detection_camera.py
./face_camera_trigger.py
./dish_classifier.py --input image.jpg
es^
```

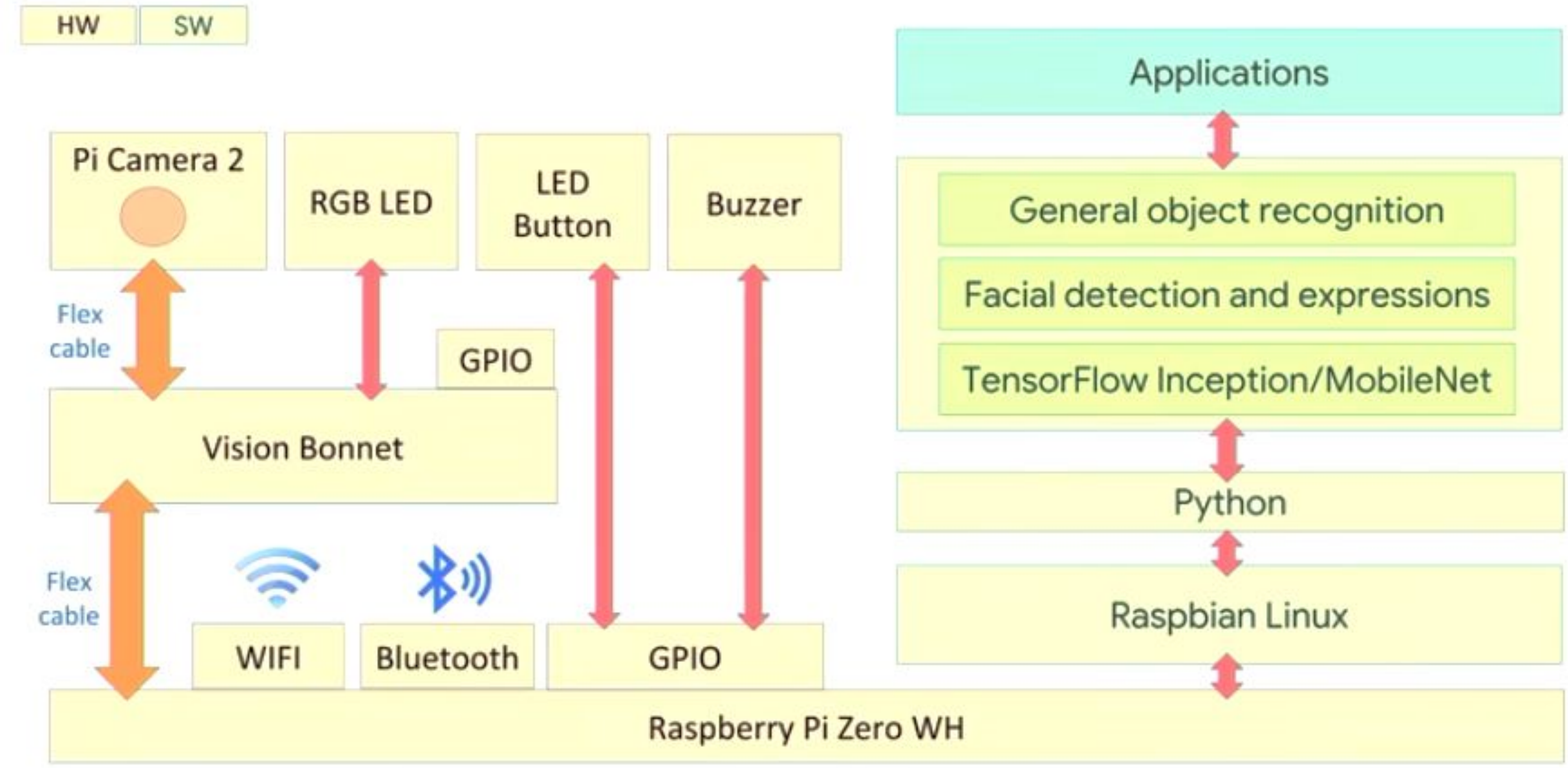


```
pi@raspberrypi:~/AIY-projects-python/src/examples/vision $ ls
annotator.py               face_detection.py               leds_example.py
buzzer                     gpiozero                        object_detection.py
dish_classifier.py         image_classification_camera.py  object_meter
face_camera_trigger.py     image_classification.py
face_detection_camera.py   joy
pi@raspberrypi:~/AIY-projects-python/src/examples/vision $
```

Get Ready for Development 2

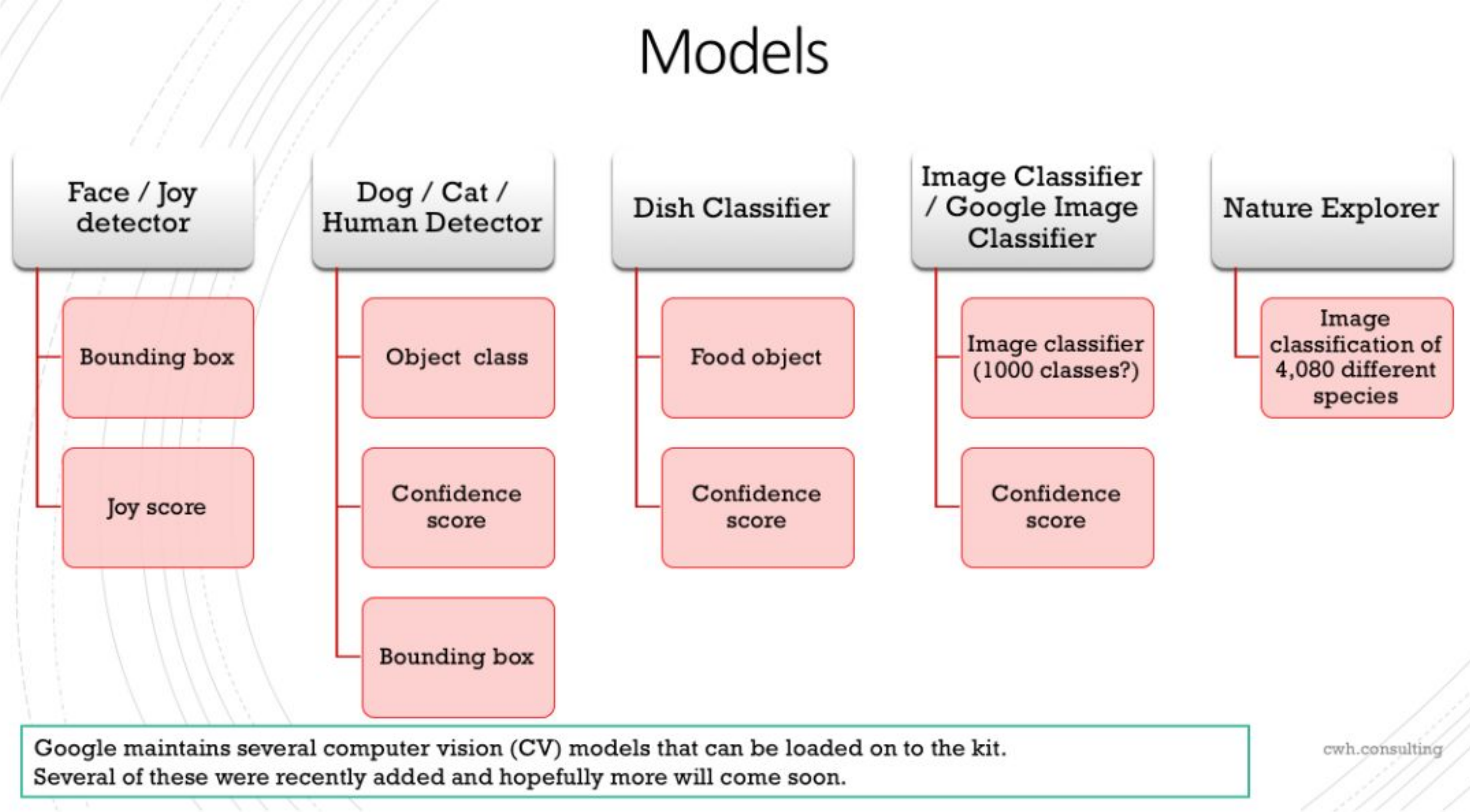
Now we also have to understand the Architecture of our Development

Below is a diagram for both hardware (hw) and software (sw) [vision]



Get Ready for Development 3

We need an understanding of the pre-built models as well [vision]



Models

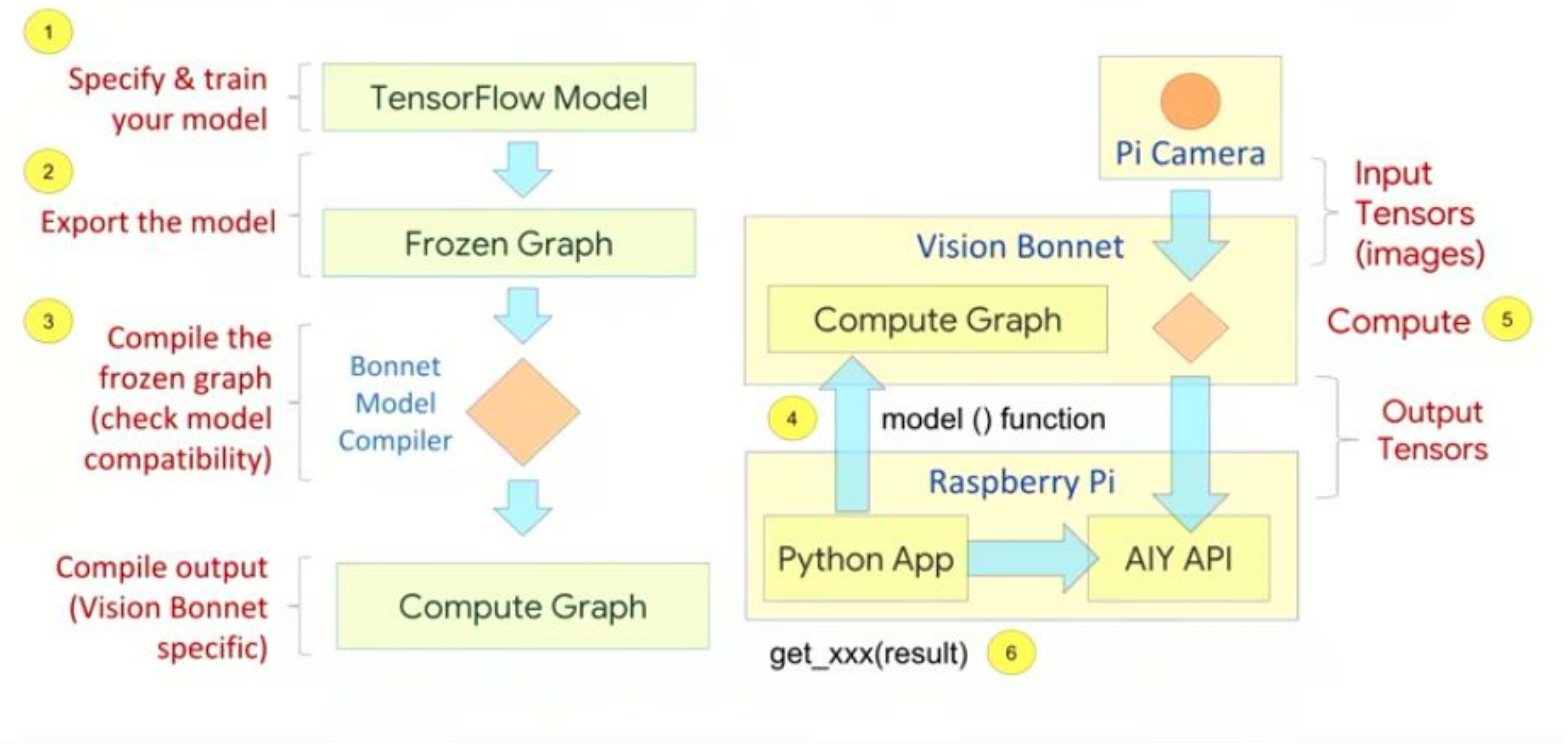
- Face / Joy detector
  - Bounding box
  - Joy score
- Dog / Cat / Human Detector
  - Object class
  - Confidence score
  - Bounding box
- Dish Classifier
  - Food object
  - Confidence score
- Image Classifier / Google Image Classifier
  - Image classifier (1000 classes?)
  - Confidence score
- Nature Explorer
  - Image classification of 4,080 different species

Google maintains several computer vision (CV) models that can be loaded on to the kit.
Several of these were recently added and hopefully more will come soon.

cwh.consulting

Get Ready for Development 4

We need an understanding of how to train our own model [vision]



What to look into

IoT and MLT

Internet of Things and the MLT Multimedia Framework would be two things to look into.

Coding the ability for the voice kit, the vision kit, and your computer to all talk to each other is apart of the project, and some research in this are would be very useful.

Here is an intro to IoT and here is

Project pain points

During the development of this project, there has been some paint points that I would like to introduce

snowboy
HOTWORD DETECTION

Custom wake word (Voice)

One of the goals of this project was to create a custom wake word.
However, a pain point in this project was trying to implement Snowboy. From Snowboy's documentation, it says it supports raspbian linux (which would help in us develop for our headless embedded system)

This is a video of someone making it work, however it is not explained.

Customization

Define and train your own hotword

High accuracy

False alarm is minimized

Always on

Low latency and no internet needed

On device computing

Small memory footprint and cross-platform support

Getting Started with AI/ML (Vision)

A Google Crash Course
The latest release

This was a primer to working with tensorflow to create my own image classifier to recognize my own key object that would control the video

Google's ML Training

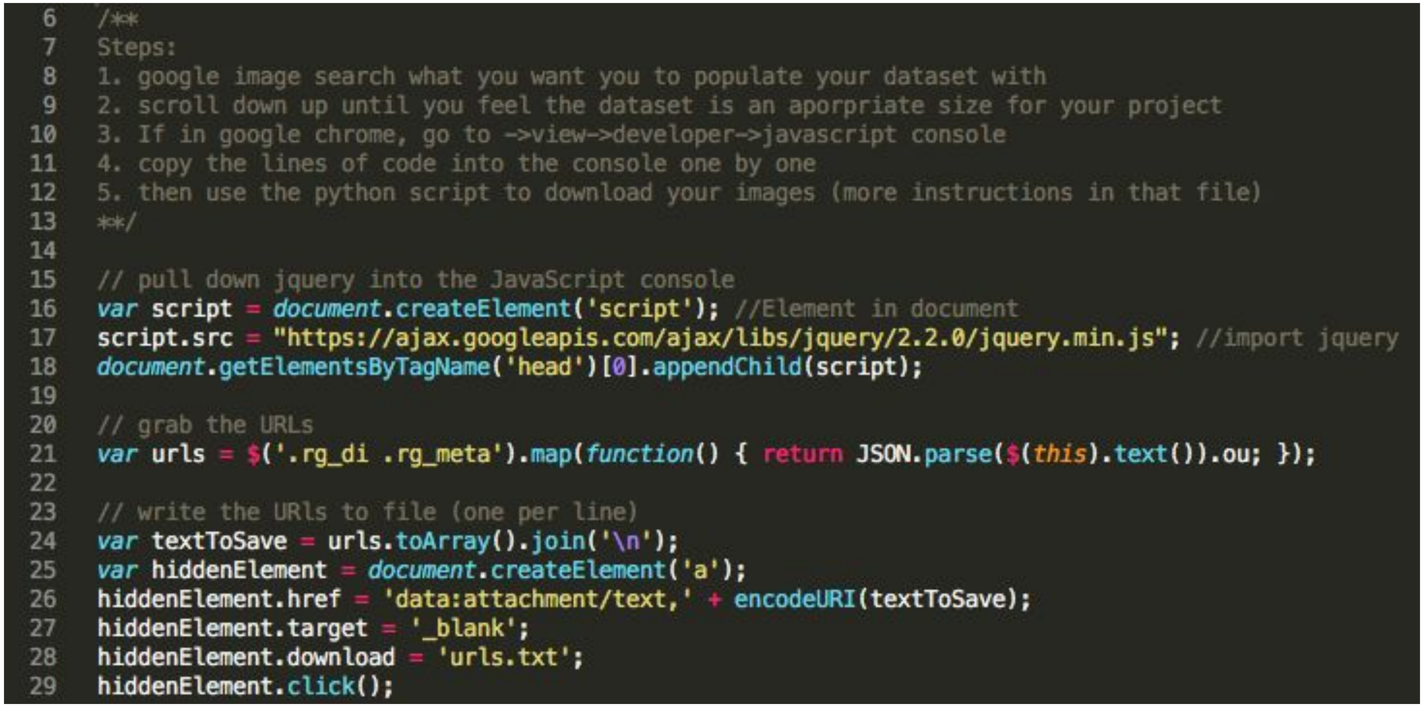
And AI Course

Creating Your own Deep Learning Data Set (Vision)

Existing code that creates datasets from google images

The pain point here was trying to use this data set to reclassify an existing tensorflow image classifier model with this tutorial (tensorflow for poets).

The code has a js file to generate a document with a list of urls. Then, following with a python script that downloads those images. Please follow this tutorial.

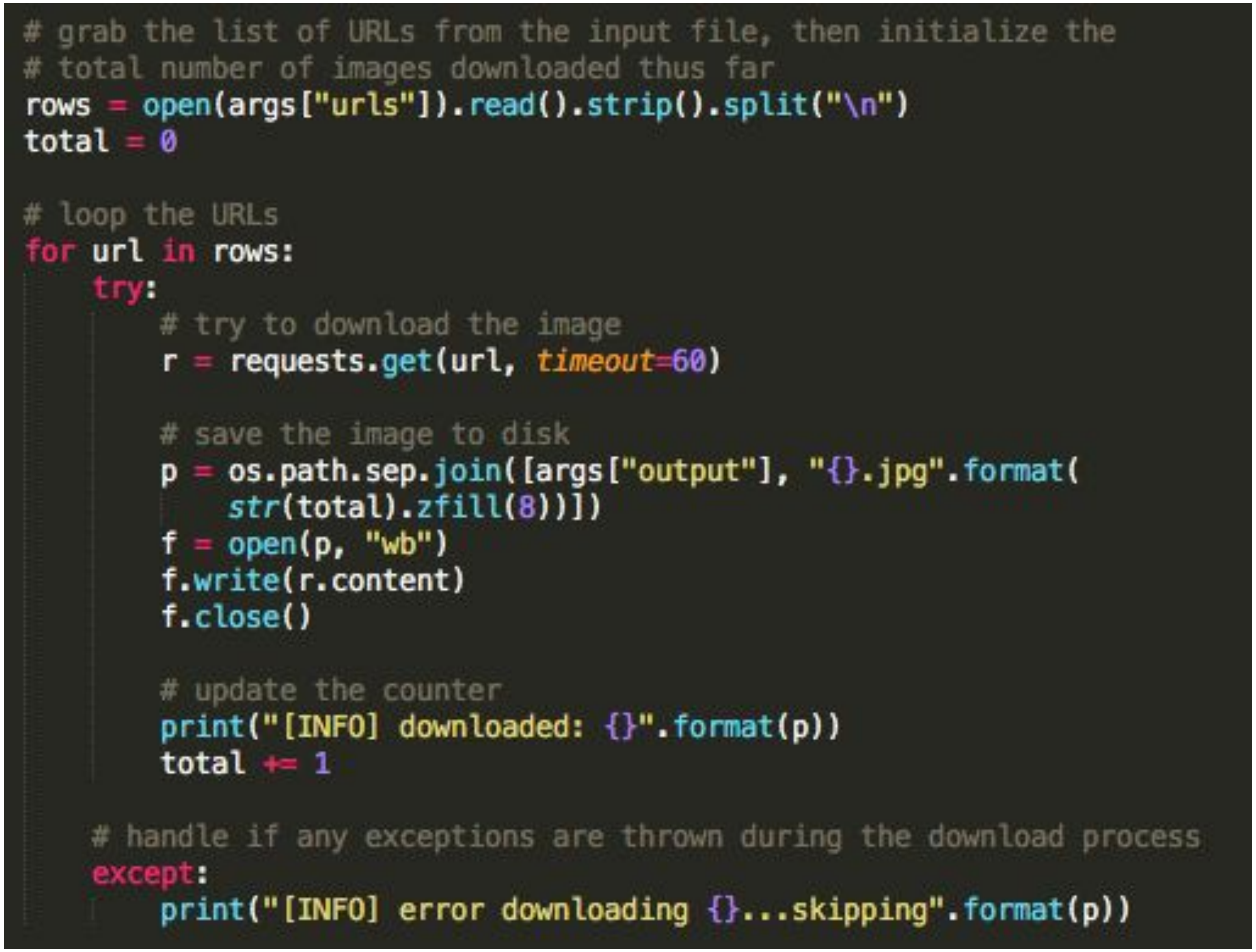


```
/**
Steps:
1. google image search what you want you to populate your dataset with
2. scroll down up until you feel the dataset is an aporpriate size for your project
3. If in google chrome, go to ->view->developer->javascript console
4. copy the lines of code into the console one by one
5. then use the python script to download your images (more instructions in that file)
**/

// pull down jquery into the JavaScript console
var script = document.createElement('script'); //Element in document
script.src = "https://ajax.googleapis.com/ajax/libs/jquery/2.2.0/jquery.min.js"; //import jquery
document.getElementsByTagName('head')[0].appendChild(script);

// grab the URLs
var urls = $('.rg_di .rg_meta').map(function() { return JSON.parse($(this).text()).ou; });

// write the URLs to file (one per line)
var textToSave = urls.toArray().join('\n');
var hiddenElement = document.createElement('a');
hiddenElement.href = 'data:attachment/text,' + encodeURI(textToSave);
hiddenElement.target = '_blank';
hiddenElement.download = 'urls.txt';
hiddenElement.click();
```

Creating Your own Deep Learning Data Set (Vision)

Example

code cont'd

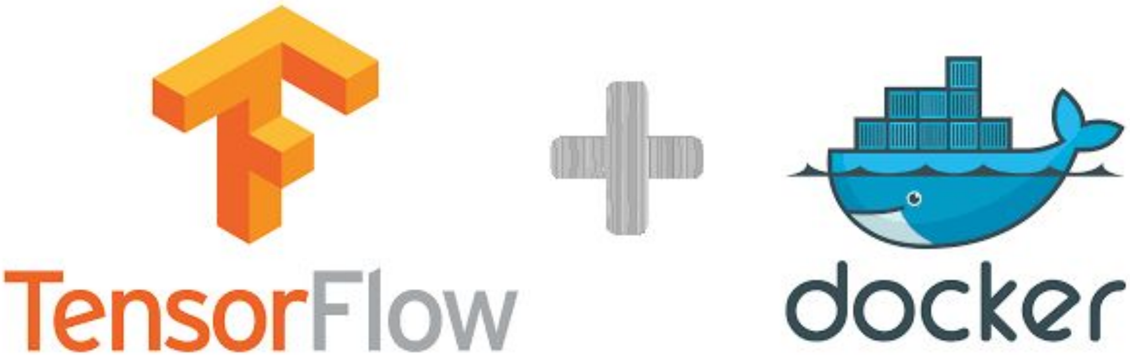


```
# grab the list of URLs from the input file, then initialize the
# total number of images downloaded thus far
rows = open(args["urls"]).read().strip().split("\n")
total = 0

# loop the URLs
for url in rows:
    try:
        # try to download the image
        r = requests.get(url, timeout=60)

        # save the image to disk
        p = os.path.sep.join([args["output"], "{}.jpg".format(
            str(total).zfill(8))])
        f = open(p, "wb")
        f.write(r.content)
        f.close()

        # update the counter
        print("[INFO] downloaded: {}".format(p))
        total += 1

    # handle if any exceptions are thrown during the download process
    except:
        print("[INFO] error downloading {}...skipping".format(p))
```

Using Docker + Tensorflow (Vision)

Docker is a helpful tool to have containers of code that can be moved around. However, a pain point in this project was trying to implement Docker.

This is a video of someone making it work.



Why do we need AI?

Is AI important in the industry?

$2,700,000,000

Whoa! That’s a big number. That’s what BofA spent on Technology. (Caroline Hudson, BizJournals)

$20,000,000 on Erica

That’s a lot of money

2,000,000 users

And a lot of users

100%

Total success!

$10,800,000,000

Whoa!! That's a bigger number. That's what JP Morgan spent on Technology. (Morell & Johnson, Business Insider)

$10,800,000,000

Spent on technology

$5,000,000,000

Set aside for investments

Focuses on Ai

Hiring high profile ai experts

Our process is easy

Interest

Develop

Produce

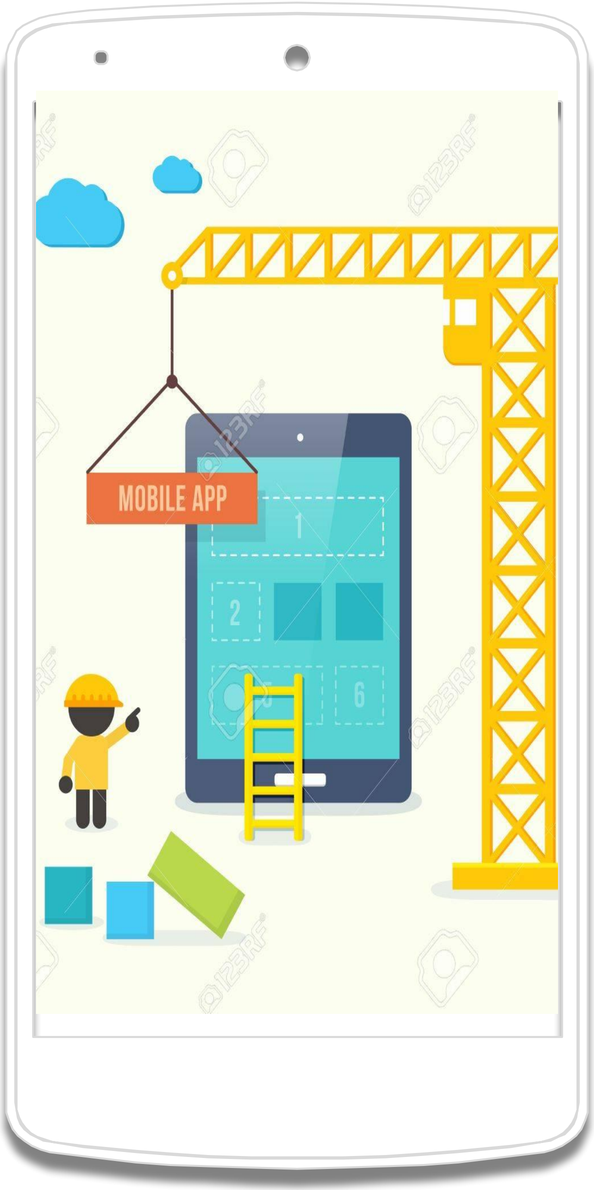
First We have to generate interest, and recruit talent for AI

We have to research and develop AI models for the many use cases there are for our bank

Put it officially in production, for our many users and shareholders to be serviced and protected by

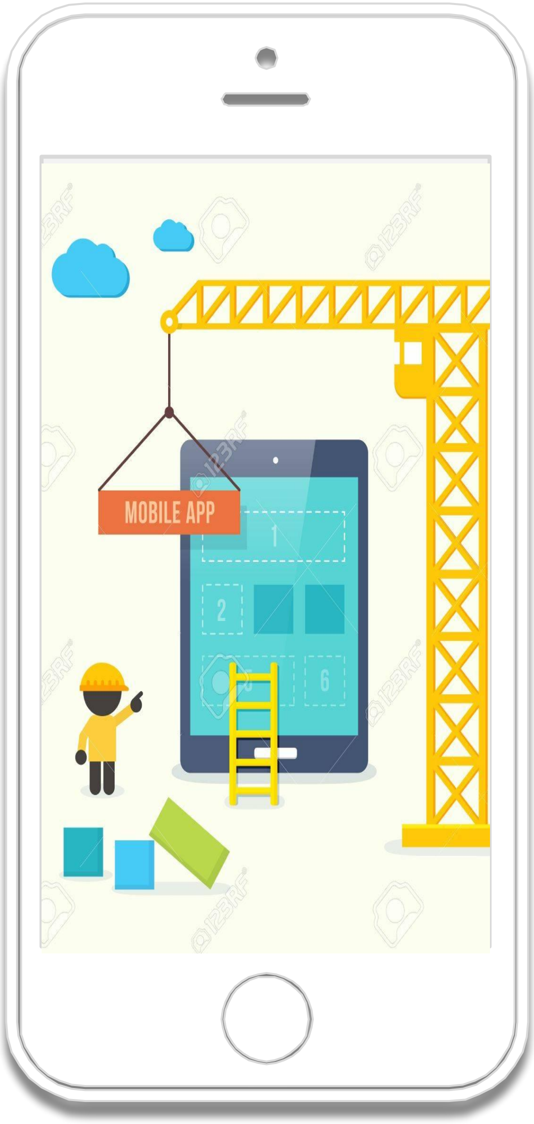
Android project

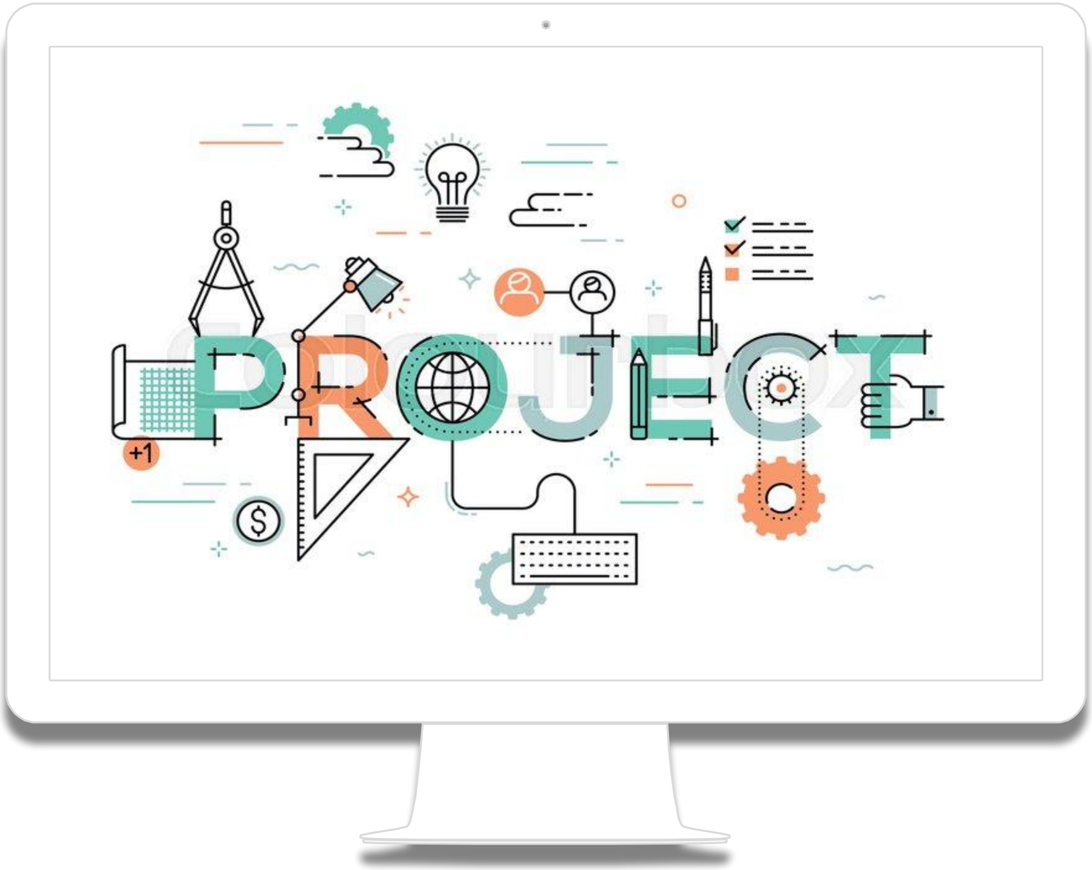
Develop AI models compatible with Android



iPhone project

Develop AI models compatible with iOS





Desktop project

Develop AI models compatible with the Web

Thanks!

Any questions?

You can find me at:

mchowdhury@mtb.com

or

mschowdh@buffalo.edu