

5-2019

# Granny Pod Virtual Assistant

David Connolly

*University of Nebraska at Omaha*, [dconnolly@unomaha.edu](mailto:dconnolly@unomaha.edu)

Bing Chen

*University of Nebraska-Lincoln*, [bchen1@unl.edu](mailto:bchen1@unl.edu)

Follow this and additional works at: [https://digitalcommons.unomaha.edu/university\\_honors\\_program](https://digitalcommons.unomaha.edu/university_honors_program)

Part of the [Digital Communications and Networking Commons](#), [Electrical and Electronics Commons](#), and the [Other Computer Engineering Commons](#)

---

## Recommended Citation

Connolly, David and Chen, Bing, "Granny Pod Virtual Assistant" (2019). *Theses/Capstones/Creative Projects*. 67.  
[https://digitalcommons.unomaha.edu/university\\_honors\\_program/67](https://digitalcommons.unomaha.edu/university_honors_program/67)

This Dissertation/Thesis is brought to you for free and open access by the University Honors Program at DigitalCommons@UNO. It has been accepted for inclusion in Theses/Capstones/Creative Projects by an authorized administrator of DigitalCommons@UNO. For more information, please contact [unodigitalcommons@unomaha.edu](mailto:unodigitalcommons@unomaha.edu).

Footer Logo

Small Sustainable House



Granny Pod virtual Assistant

## User Manual

David Connolly, B.S. Electrical Engineering, Student, UNL  
Dr. Bing Chen, Ph.D. Electrical Engineering, Professor, UNL

Published: April 29, 2019

# ABSTRACT

Dr. Chen is working on a sustainable small house (SSH) project, sometimes called the “Granny Pod”. Regulations will soon allow homeowners to house their parents on their property, which can be an opportunity live independently in a cheap, sustainable, and convenient alternative to a retirement community. To help achieve this vision, a Virtual Assistant system for the SSH was developed. The system uses a Google Home or Amazon Echo to respond to the voice command “Hey Google (or Alexa), I need help” by contacting the nearby homeowner or caretaker. It alerts the resident who is at the door when the doorbell is rung. The system is also programmable if future development is desired. With the Virtual Assistant, the living experience in the SSH will be more convenient, safe, and well-connected.

## Overview

The SSH will allow as much self-reliance as possible to the resident, while also keeping the caretaker close by. To accomplish this, the Granny Pod Virtual Assistant (GPVA) project was conceived by Dr. Chen and assigned to Electrical Engineering student David Connolly as a Capstone project. It has two main functions: The Help Needed Message and the Front Door Facial Alert.

For the Help Needed Message function, the SSH resident will be able to simply say: “Hey Google, I need help” (or similar phrases) and an instant alert message is sent to the caretaker’s phone along with a picture of the inside of the SSH (a optional feature). The program is customizable by the caretaker to allow for customizable messages to be sent.

For the Front Door Facial Alert function, the Virtual Assistant will connect to a front door camera. Then whenever someone appears at the front door, the camera will identify who is there and alert the resident, saying “Attention! \_\_\_\_\_ is at the door” through a speaker.

The GPVA will be utilized in the Small Sustainable House when it is built in latter 2019. As the program continues to be developed by other students and staff beyond David’s Capstone, the instructions may become outdated, but the Github site listed below will always have the latest instructions and current code available.

## Materials Needed

- Raspberry Pi 3 or later
- Micro SD Card for Raspberry Pi
- Micro USB Cable
- Computer with SD Card Reader
- USB wall power brick (5V=2.5A)
- HDMI Cable
- USB Mouse
- USB Keyboard
- 2 x Logitech C270 Desktop or Laptop Webcams
- Google Home
- Phone with Service
- Monitor with HDMI input

# Total Cost

Prices on Amazon:

Product	Quantity	Price per unit
Raspberry Pi 3 or later	1	\$37.88
Micro SD Card for Raspberry Pi	1	\$7.38
Micro USB Cable	1	\$2.33
USB wall power brick (5V=2.5A)	1	\$9.99
HDMI Cable	1	\$6.99
USB Mouse	1	\$5.99
Small amplified speaker	1	\$8.99
USB Keyboard	1	\$8.99
Logitech C270 Desktop or Laptop Webcams	2	\$19.86
Google Home Mini	1	\$30.00
Total		\$138.40

## Installation Guide

Please complete ALL of these installations (online, phone, Raspberry Pi) for the system to work.

### Online Installation

1. Create a free trial account on Twilio ([www.twilio.com](http://www.twilio.com))
2. Create an account on ngrok.io ([www.ngrok.com](http://www.ngrok.com))
3. Create imgBB ([www.imgBB.com](http://www.imgBB.com))
  - a. Go to imgBB settings and set autodelete to after 5 minutes, then **press save changes** at the bottom of the settings page
4. Create account at IFTTT ([www.ifttt.com](http://www.ifttt.com))

### Raspberry Pi Installation

1. Download balenaEtcher (<https://www.balena.io/etcher/>) and download latest version of Raspian (<https://www.raspberrypi.org/downloads/raspbian/>). The Raspian version you choose doesn't have to be the Lite Version or the version with recommended software
2. Use a MicroSD Card reader and balenaEtcher to burn Raspian onto the SD Card
3. Insert MicroSD Card into Raspberry Pi
4. Plug in power to Raspberry Pi through USB
5. Plug in a USB keyboard, a USB mouse, and both USB webcams
6. Plug the speaker into the auxiliary port
7. Open console
8. Run `sudo apt-get update` and `sudo apt-get upgrade`
9. Install Python 3.6.0 or any later version and dependencies required for OpenCV  
Enter these commands into the console (provided by Vishwesh Shrimali from [learnopencv.com](http://learnopencv.com) [3]):  
`sudo apt-get install python3-dev libffi-dev libssl-dev -y`

```
wget https://www.python.org/ftp/python/3.6.3/Python-3.6.3.tar.xz
tar xJf Python-3.6.3.tar.xz
cd Python-3.6.3
./configure
make
sudo make install
```

**Run the console command:** `sudo apt-get -y install _____`

**where the blank space is filled with these libraries separately:**

```
build-essential checkinstall cmake pkg-config yasm
git gfortran
libjpeg8-dev libjasper-dev libpng12-dev
libtiff5-dev
libtiff-dev
libavcodec-dev libavformat-dev libswscale-dev libdc1394-22-dev
libxine2-dev libv4l-dev
```

**Run these console commands:**

```
cd /usr/include/linux
sudo ln -s -f ../libv4l1-videodev.h videodev.h
cd $cwd
```

**Continue running the console commands below:** `sudo apt-get -y install _____`

```
libgstreamer0.10-dev libgstreamer-plugins-base0.10-dev
libgtk2.0-dev libtbb-dev qt5-default
libatlas-base-dev
libmp3lame-dev libtheora-dev
libvorbis-dev libxvidcore-dev libx264-dev
libopencore-amrnb-dev libopencore-amrwb-dev
libavresample-dev
x264 v4l-utils
libprotobuf-dev protobuf-compiler
libgoogle-glog-dev libgflags-dev
libgphoto2-dev libeigen3-dev libhdf5-dev doxygen
```

10. Install pip3 by entering this line in the console: `sudo apt install python3-pip`

11. Use pip3 to download all the python libraries needed for the GPVA

Use this line of code in the Raspberry Pi Console: `pip3 install _____`

In the blank space of the code, fill in these library names one by one and press 'y' when asked to install:

- gTTS
- numpy
- opencv-contrib-python
- pickle
- pygame
- requests
- thonny
- twilio

12. Download and unzip the GPVA repository from Github ([github.com/IntiEngineer/SSH-GPVA](https://github.com/IntiEngineer/SSH-GPVA))

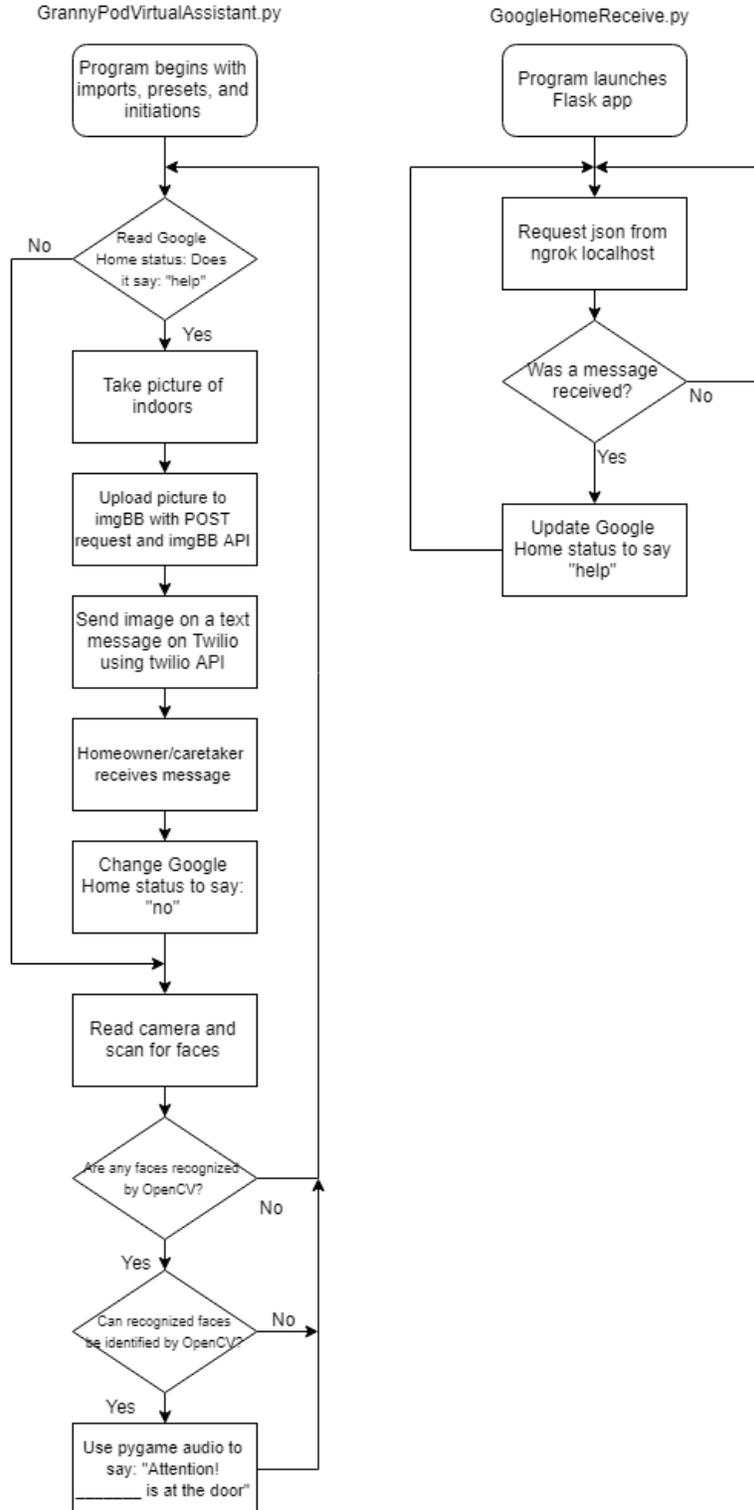
13. Run the face trainer program in one of two methods:
  - a. Use NewFace.py program
    - i. Gather everyone that will be facial recognized
    - ii. Run the NewFace.py program and follow the instructions of entering names and different expressions for each person
    - iii. Run FaceTrainer.py
  - b. Manually fill the FacesForTrainer folder
    - i. Within the FacesForTrainer folder, create a folder for each person labelled with their respective name
    - ii. Fill each person's folder with many different pictures of their face, more pictures results in a more accurate program
    - iii. Run FaceTrainer.py
14. Download and unzip the IOT Google Home Github repository by Nassir Malik at: <https://github.com/nassir-malik/IOT-Pi3-Google-Home-Mini-Automation>
15. Run tutorialsetup command lines with file or in console manually
16. In the console, use cd command to access IOT-Pi3-Google-Home-Mini-Automation directory  
Example: `cd Documents/IOT-Pi3-Google-Home-Mini-Automation`
17. Enter into console: `./ngrok http 5000` to start ngrok localhost
18. In new console window, use cd command to access GPVA directory  
Example: `cd Documents/GPVA`
19. Enter into console: `python3 GoogleHomeReceive.py`
20. Open GrannyPodVirtualAssistant.py and run the program
21. Unplug the mouse, keyboard, and monitor

## Phone Installation

1. Download Google Home app on the app store
2. Connect Google Home to its app on your phone, ensuring that they are on the same wifi
3. Download IFTTT on the app store
4. Create an IFTTT applet
  - a. Enter these values into the spaces provided:
    - i. +This: Google Assistant
    - ii. What do you want to say? I need help
    - iii. What's another way to say it? Help me
    - iv. And another way? (optional) Please help
    - v. What do you want the Assistant to say in Response? Help is on the way.
    - vi. Language: English
    - vii. +That: Webhooks
    - viii. URL: `https://_____ .ngrok .io`, blank space is occupied by the code found on the "follow URL" section in console of the ngrok program (if the free version of ngrok is being used, then the address will change every time ngrok localhost is restarted and every 8 hours)
    - ix. Method: POST
    - x. Content Type: application/json
    - xi. Body: `{"message", "help"}`

# Theory of Operation

To operate successfully, the GPVA is a collection of programs that run simultaneously. These processes are shown below in two flow charts.



A third program is run that is not shown because it simply is a local host that is in a permanent state of hosting a server.

GrannyPodVirtualAssistant.py runs both functions simultaneously, checking conditions for each one in a continuous while loop. For the Help Needed Message function, the program is continuously checking the text within a file called GoogleHomeStatus.txt. If the contents of this file change from the word “no” to “help”, the program will take a snapshot from the indoor webcam, save it, upload it to the internet to a site called imgBB with a request API. A texting service called Twilio will then be given the URL of the imgBB image and send it to a phone number entered along with a customizable message currently set to say: “Granny is in need of help”. The online imgBB image will then be deleted after 5 minutes.

The program also runs facial recognition code where it continuously observes the outdoor camera looking for faces using OpenCV. OpenCV processes a video frame which contains a face, then continuously that checks for faces to appear through another conditional. When a face appears and the condition is met, then the face will be processed against trained facial data looking for a match. If a face match is found, then the message is generated in a string and processed in the gTTS library to generate an mp3 file with the spoken message. The mp3 file is saved and reopened by another library called pygame that plays the mp3 file through the designated audio device.

GoogleHomeReceive.py is a Flask app that continuously makes requests to a public server hosted by ngrok. If IFTTT sends a message to ngrok, it will be picked up by the Flask app. This will then write the word “help” to the GoogleHomeStatus.txt file to be handled by the main code.

## Alexa

The use of IFTTT makes the implementation of Alexa rather than Google Home very easy. In the same way that there is a Google Assistant applet for +This, there exists an Amazon Alexa applet for +This. By simply switching out the two, there the current version of the program will still work perfectly.

## Security and Privacy

All name and imagery data is handled by the different imported python libraries. Any data used by GPVA not handled by a library is solely handled by the Raspberry Pi and is subject to the internet security of the Pi. An internet connection is required for the GPVA, which may put the Raspberry Pi at risk. Images used for facial recognition training and previously sent pictures in the GPVA folder are retained. For Help Needed Message function, the message containing the picture from indoors is transmitted over the internet and is stored unencrypted for 5 minutes on the internet cloud, before being deleted. This presents a security risk which is why this feature is optional and is activated with a flag in the settings section.

## Code

- The most up-to-date version of the code will always be located at [github.com/IntiEngineer/SSH-GPVA](https://github.com/IntiEngineer/SSH-GPVA)
- GPVA was proposed by Dr. Chen, Ph.D. and programmed and published by David Connolly
- OpenCV Facial Recognition code is provided by CodingEntrepreneurs ([www.youtube.com/user/CodingEntrepreneurs](https://www.youtube.com/user/CodingEntrepreneurs)) [1]

- IOT ngrok and Flask localhost software is provided by Nassir Malik (Netmedias on Youtube: <https://www.youtube.com/channel/UCn8v7OzXk7IqRdKZdf14yjA>) [2]
- OpenCV installation library and dependency information was provided by Vishwesh Shrimali from learnopencv.com (<https://www.learnopencv.com/install-opencv-4-on-raspberry-pi/>) [3]

## General Troubleshooting

- Please report any errors or bugs to the Github page.
- If the GrannyPodVirtualAssistant.py, GoogleHomeReceive.py, and ngrok are not all running simultaneously with an internet connection, then the program will not work.
- If the ngrok is not restarted and IFTTT are both update every eight hours on the free trial version, the program will not work.
- If the wifi or ethernet access is in a secure location that requires a login and password, the program may not work due to lacking Raspberry Pi and Google Home capabilities.
- Many non python/coding problems can be simply solved by unplugging and replugging in the Raspberry Pi.
- Keep the Raspberry Pi in an open place, not a tight space, since it need air to radiate heat and not get too hot.
- Problems may arise from a missing library or dependency. This can be solved by reading the error, where there is normally a part that says: \_\_\_\_\_ doesn't exist. By searching this part online, a missed library or dependency can be found and installed with pip3.
- If there are any other errors, copy and paste the error and online for further help.

## Potential Future Improvements

- In the next version, security must be improved since the current program uses 3<sup>rd</sup> party localhosts, twilio, and imgBB. Optimally, they would be replaced by hosting on a Granny Pod website. The messages could also be encrypted before they are sent. The Raspberry Pi itself could be programmed as a localhost for hosting and tunneling information in requests.
- There are currently three programs that need to be run simultaneously, and two Facial Recognition programs, which needs to be started with a HDMI enabled display every time the Raspberry Pi powers up. A vast improvement would be to make only one executable program that works on any system and starts every time the Raspberry Pi powers on.
- Unless a paid version of ngrok is obtained, the local program needs to be restarted every eight hours and its public follow URL needs to be reentered into the IFTTT applet. This is unsustainably difficult or expensive, so an alternative localhost service should be used.
- The Facial Recognition message is currently delivered to a speaker connected to auxiliary. This can be upgraded by using the Google Home as the speaker. This may be possibly achieved by using webhooks, a home assistant API, or by repurposing the Raspberry Pi itself as a Google Home.
- This project could be more thoroughly integrated into Alexa.
- The Raspberry Pi Raspian OS can be replaced by a faster Linux OS and the Raspberry Pi itself could be replaced with a faster and cheaper version of a small computer machine.
- The current facial recognition software in use is provided by OpenCV. It is quick and easy, but not extremely accurate. Google's TensorFlow would be a much better and modern alternative and would be optimally implemented in future versions.

# References

1. CodingEntrepreneurs, “OpenCV Python TUTORIAL #4 for Face Recognition and Identification,” *YouTube*, 11-Apr-2018. [Online]. Available: <https://www.youtube.com/watch?v=PmZ29Vta7Vc>. [Accessed: 29-Apr-2019].
2. N. Malik, “Raspberry pi Control Solid State & Mechanical Relay with Google Mini | Tutorial # 17,” *YouTube*, 27-Dec-2017. [Online]. Available: <https://www.youtube.com/watch?v=1Eo9NSiS3Y8>. [Accessed: 29-Apr-2019].
3. V. Shrimali, “Home,” *Learn OpenCV*, 19-Nov-2018. [Online]. Available: <https://www.learnopencv.com/install-opencv-4-on-raspberry-pi/>. [Accessed: 29-Apr-2019].