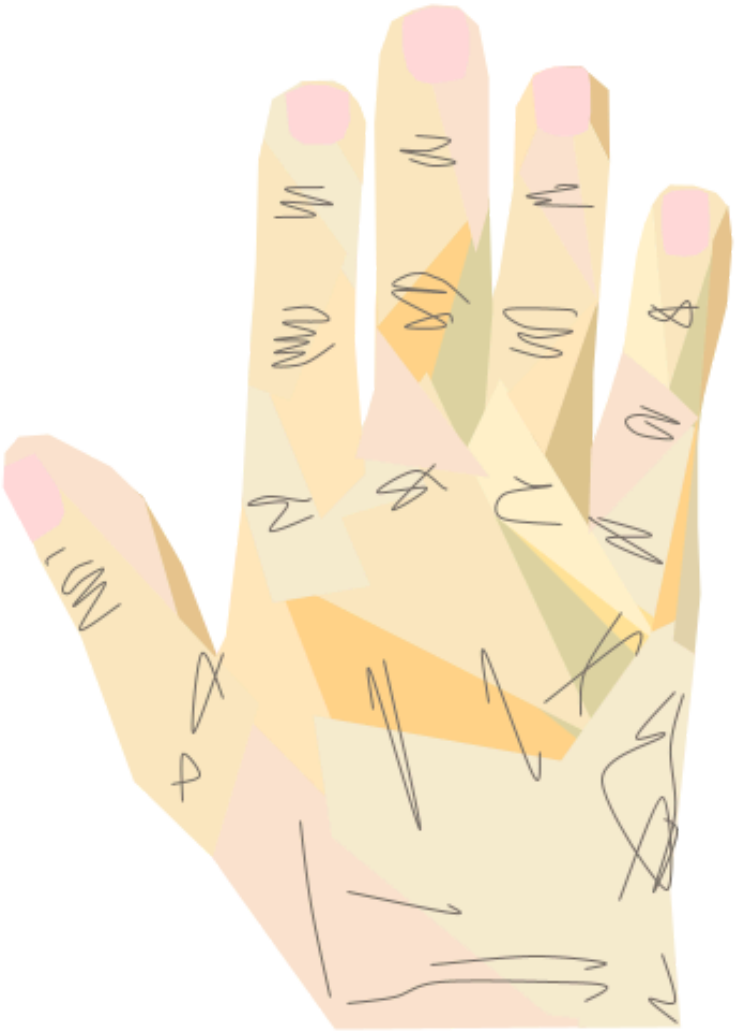
SMART GESTURES

Pervasive Computing Smart System Project

Vrije Universiteit Amsterdam
15.12.2021



Michael Adrian Polesensky
Jakub Olaf Dryja
Mubaraq M. Fuseini

VU UNIVERSITY AMSTERDAM

Table of Content

Introduction

It's imperative to note that as the world evolves with increasing technology, we are all tempted to adjust to the new trend. As a result, we try to acquire any meaningful appliance or gadget that will help make our lives easier and safer, from the use of smart curtains to smart phones, smart televisions, microwaves, desktop computers, refrigerators, etc. These things function effectively as a result of embedded systems or microprocessors in them. This also means that embedded systems are now found in our everyday life and have become part of us in many instances. It's practically difficult to see a household without an embedded system.

Goal

The goal of this project is to improve the use of gestures to control systems around us with the sole purpose of improving the quality of life. That is to say, one can possibly sit at the comfort of his home and control some selected smart systems or appliances around him within a certain range. It will help reduce the number of gadgets (appliances with remote-controls) and also save the cost of maintaining them with regards to replacement of cell batteries etc.

We will create a prototype of a remote control of a hand/body gesture type. We will combine hardware and software to interact with each other as well as interact with humans. A code(software) in an appliance(hardware) being controlled by human gestures

Background study

The use of hand gestures are often culturally bound and can vary from group to group. But there are a few of them which, if not universal, are very common indeed around the world. The salute, the thumbs up, the high five, the handshake and etc are all gestures used as a communication tool among humans. Technological equipment and human activities/innovations are being used together to ease the lives of humans in terms of communication, e-learning, e-business, etc. A number of research and innovations has been carried out globally in this regard. A continuous improvement is therefore an

essential requirement for sustaining and gaining global or competitive advantage of any similar project. In our literature review, we were overwhelmed with the level and amount of interest in smart systems hence we will give a brief summary of two papers we reviewed.

Let's talk about a paper from César Osimani, Jose A. Piedra-Fernandez, Juan Jesus Ojeda-Castelo, and Luis Iribarne on:

Hand Posture Recognition with standard webcam for Natural Interaction [4]

Their paper talked about a prototype designed for natural human-computer interaction in an intelligence environment system. They used computer vision resources to analyse images captured by a webcam to recognize a person's hand movement or gesture. They also implemented a mechanism for natural interaction to analyse images captured by a webcam based on hand geometry and posture, to show its movements in their model. The camera has to be installed in such a manner that it can discriminate movements a person makes using background subtraction. Then their system will search for hands assisted by segmentation by skin colour detection and a series of classifiers. Finally, the geometric characteristics of the hands are extracted to distinguish defined control action positions. Our prototype design will be simpler and quicker in the sense that the system will capture the hand gesture with a standard/normal webcam and convert the BGR image into a binary interpretation and run through the classifier to detect a match and perform an action.

The second paper we would like to review is the one written by Noor A. Ibraheem & Rafiqul Z. Khan

Vision Based Gesture Recognition Using Neural Networks Approaches: A Review [5]

In this paper, they discussed research done in the area of hand gesture recognition based on Artificial Neural Networks approaches. That is to say that several hand gesture recognition researches that use Neural Networks are discussed in their paper and they also made comparisons between these methods, advantages and drawbacks of the discussed methods also included, and implementation tools for each method were presented as well.

The paper made a clear distinction between a Data glove approach that employs mechanical or optical sensors Attached to a glove that transforms finger flexions into electrical signals to determine the hand posture being used by others and the Vision based approach that captures input images using camera(s).

Data glove

Vision Based

Vision base approach can further be categorised into either **appearance-based approach** using features extracted from visual appearance of the input image model of the hand, and compares the modelled features with features extracted from input camera(s) or video input or **3D model based approach** that depends on the kinematic hand DOF's of the hand that will infer some hand parameters like, pose of palm, joint angles from the input image, and make 2D projection from 3D hand model. There is a summary of the type of gesture projects in a table below

Method Name	Type of input device	Segmentation operation	Feature vector representation	Neural network type	# sample gestures	Recognition rate	Recognition time
Japanese language recognition	Data glove	threshold	13 data item (10 for bending, 3 for coordinate angles)	back propagation network	42	71.4%	Several seconds
			16 data item (10 for bending, 3 for coordinate angles, 3 for positional data)	Elman recurrent network	10	96%	N
Arabic language recognition	Colored glove, Digital camera	HSI color model	Available Features from resource	Elman recurrent network	30	89.66%	N
				Fully recurrent network	30	95.11%	N
Myanmar language recognition	Digital camera	threshold	Orientation histogram	supervised neural network	33	90%	N
signal Gesture	accelerometer sensor, wireless mouse	Automatically (magnitude acceleration signal) / manually (wireless mouse button)	do not require in signal predictors	Continuous Time Recurrent Neural Networks	160	94%	N
shape fitting gesture	Digital camera	YCbCr color space	Two angles of the hand shape, compute palm distance	Self-Growing and Self-Organized Neural Gas	31	90.45%	1.5 seconds

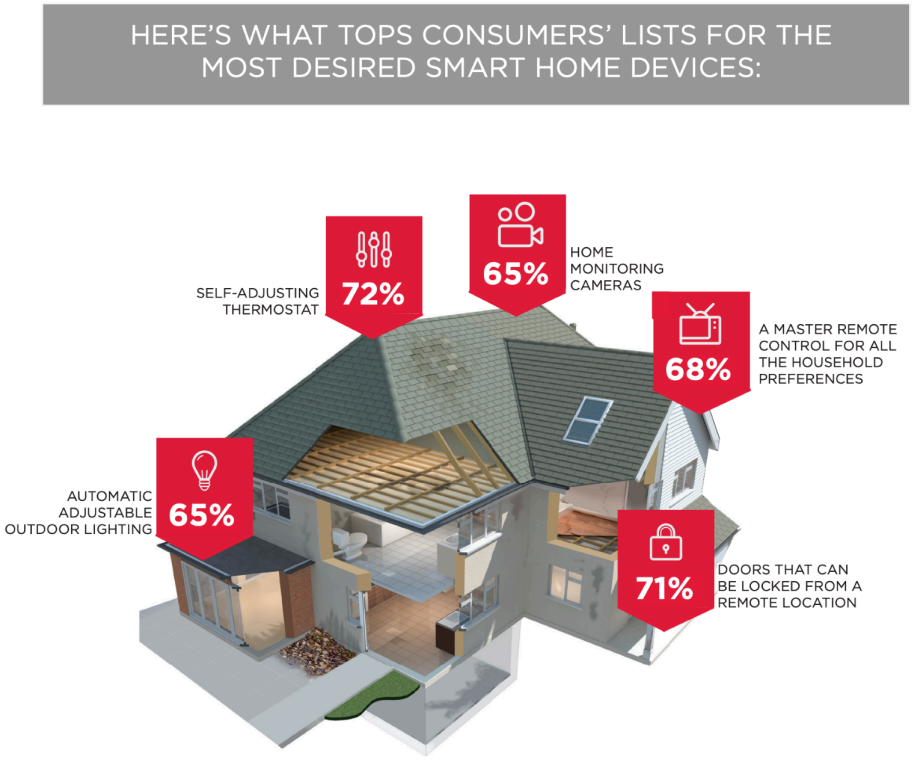
TABLE 3: Comparison between recognition methods in hand gesture recognition approach used.

The Future Smart Home: 500 Smart Objects Will Enable New Business Opportunities.[1]

This paper highlighted the fast and increased interest in the purchase of smart homes with specific needs to further improve the quality of lives.

"By 2022, a typical family home could contain more than 500 smart devices". [Gartner] [3]

They think the smart home industry will experience a dramatic evolution over the next decade and will eventually offer a lot of digital business opportunities to aspiring innovators like us who can adapt projects to exploit it. Our system will in the first phase target all equipment with cameras while other sensor equipment can be controlled by automatic or self-adjusting thermostats.



control State of the Smart Home 2015 [3]

The paper also stressed on the possible rise in sophisticated devices using only sensors and remote-control functions and we believe our hand gesture control function will be an alternative.

CONCEPT OF OPERATION

Mission Statement

To improve the quality of communication and interaction in pervasive computing by combining hardware and software to interact with humans, hence, ascertaining how smart systems reflect human thoughts and actions.

Block Diagram

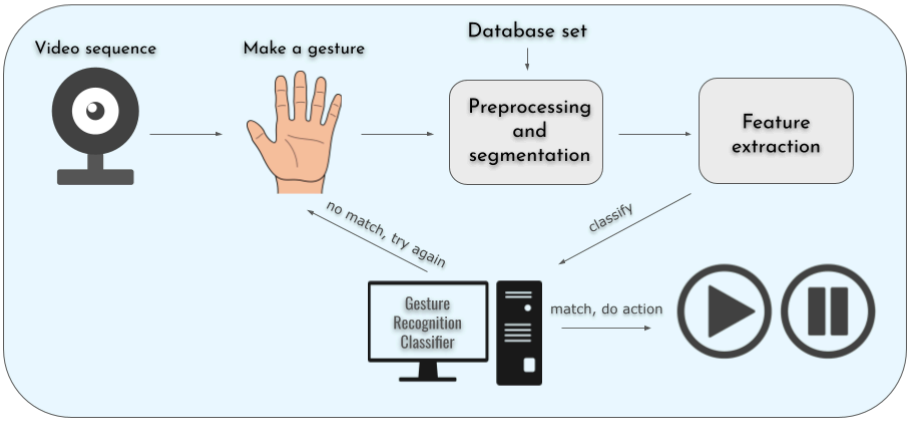


Fig 0.1 - Block diagram showing the rough functionality and high-level design

Scenarios

The system will be applicable in many instances:

- The system can be used by people living with disability who can't walk frequently around their home
- It will be used by patients in the hospital on their sick beds
- This system will provide or serve as an alternative to physical remote control of gadgets in the offices and homes.

Stakeholders

- Developers
- Lecturer
- Teaching assistant
- End users

REQUIREMENTS

These system requirements are the needs and tools necessary for the project to be successful. It is important to elicit the necessary requirements and effectively model them into action to achieve our goal. It's equally important to elicit requirements from the stakeholders since they will be influenced directly or indirectly by our project. We have agreed to adopt both the artefact-based and stakeholder driven techniques in our elicitation.

Functional

- The system shall be able to take human gestures as inputs
- The system should be able to store inputs or gesture taken
- The system must have a database of trained gestures that will enable it to identify a gesture in its classifier
- The system must be able to read and understand gestures(interpret)
- The system must be able to convert images into binary

Non-functional

- The system should freeze/halt after several failed attempts.
- The system could indicate with a flash after a successful attempt
- The system database classifier should be reliable
- The system could interpret between different sizes of gestures
- The system should process a gesture faster

Prioritisation (MoSCoW)

We have centralised and restructured our requirements from stakeholders into actionable insights in an order of priority and importance. Prioritisation was done in a collaborative process and in an order that linked random requirements from the stakeholders to the goals and success of our project.

In our MoSCoW technique, letter ‘M’ is a Must-Have deemed to be non-negotiable and mandatory for the success of the project while ‘S’ is Should-Have indicating vital initiatives that adds significant value to the project. The letter ‘C’ Could-Have referred to initiatives that will have small impact if left out while ‘W’ Won’t-Have referred to initiatives that are not priority.

Requirement	Must-Have	Should-Have	Could-Have	Won't-Have
Functional	-The system must have a database of trained gestures that will enable it to identify a gesture in its classifier	-The system should be able to convert images into binary - The system should be able to take human gestures as inputs	- The system could be able to store inputs or gesture taken	-The system won't be able to control mouse continuously by the gesture
Non-Functional	- The system could indicate with a flash after a successful attempt - The system database classifier should be reliable	-The system should process a gesture faster - The system should freeze/halt after several failed attempts.	- The system could interpret between different sizes of gestures	

Constraints

Every good project passes through various stages to perfection. However, a lot of hurdles must be crossed to achieve the goal of the project. our constraints were:

- Time – we didn't have the luxury of time to elicit requirement, conceptualise our dream, implement and test our project
- Tools – we also had to rely on computer webcams for the entire duration of our project when we could have used depth-sensor cameras, etc.
- Data – It was difficult gathering data from our targeted stakeholders. We estimated not less than a hundred thousand samples.

UML DIAGRAMS

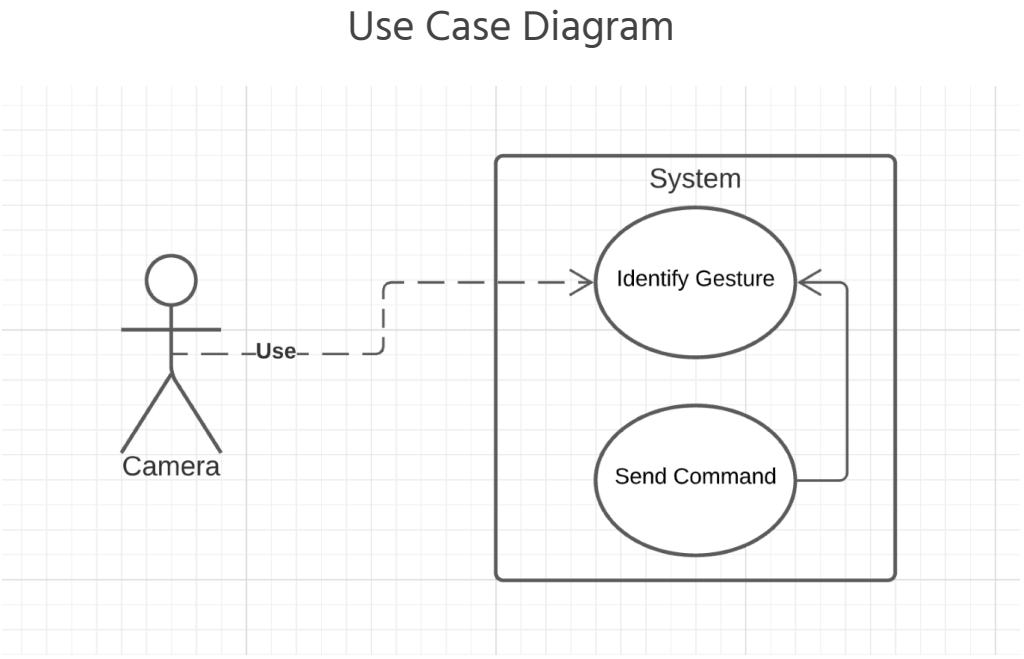


Fig 1. Use Case diagram

This is the used case diagram for our project. There is only one actor. The camera sends the images to the classifier which triggers appropriate events to be able to control the computer. The classifier reads from a stream of images that come from the webcam and classifies them. Then it triggers the appropriate action.

Class Diagram

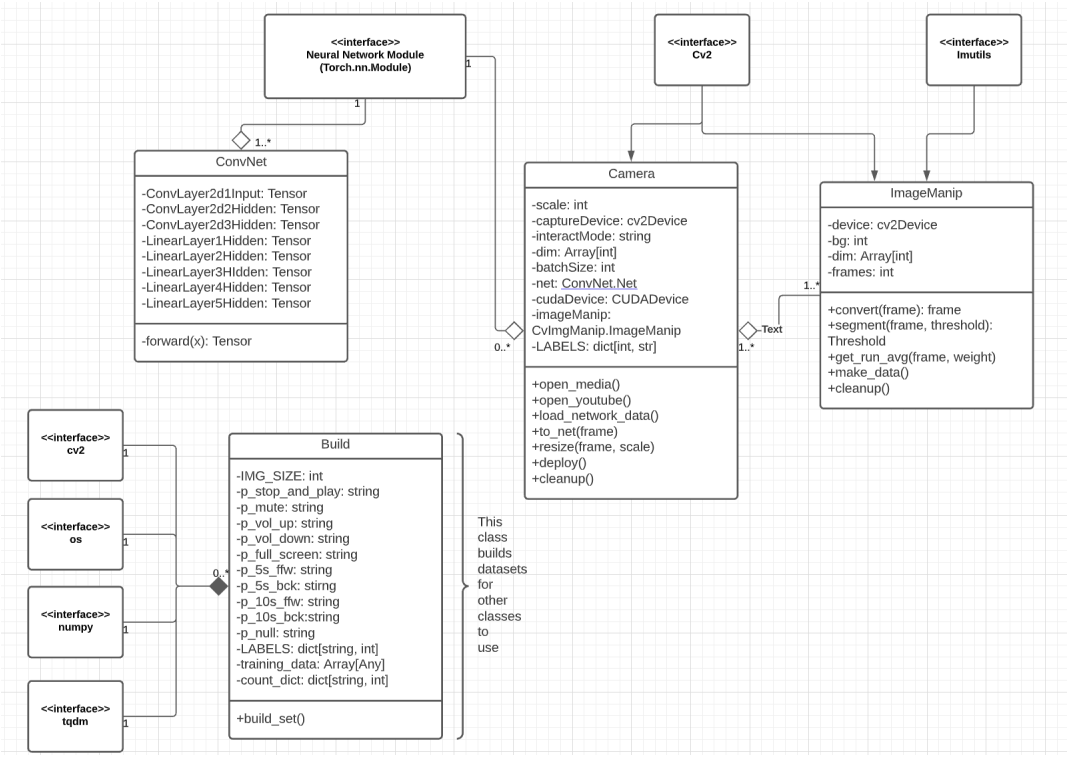


Fig 2. Class diagram

Our class diagrams illustrate how our classes cooperate with each other. And any dependencies that exist.

Event Diagram

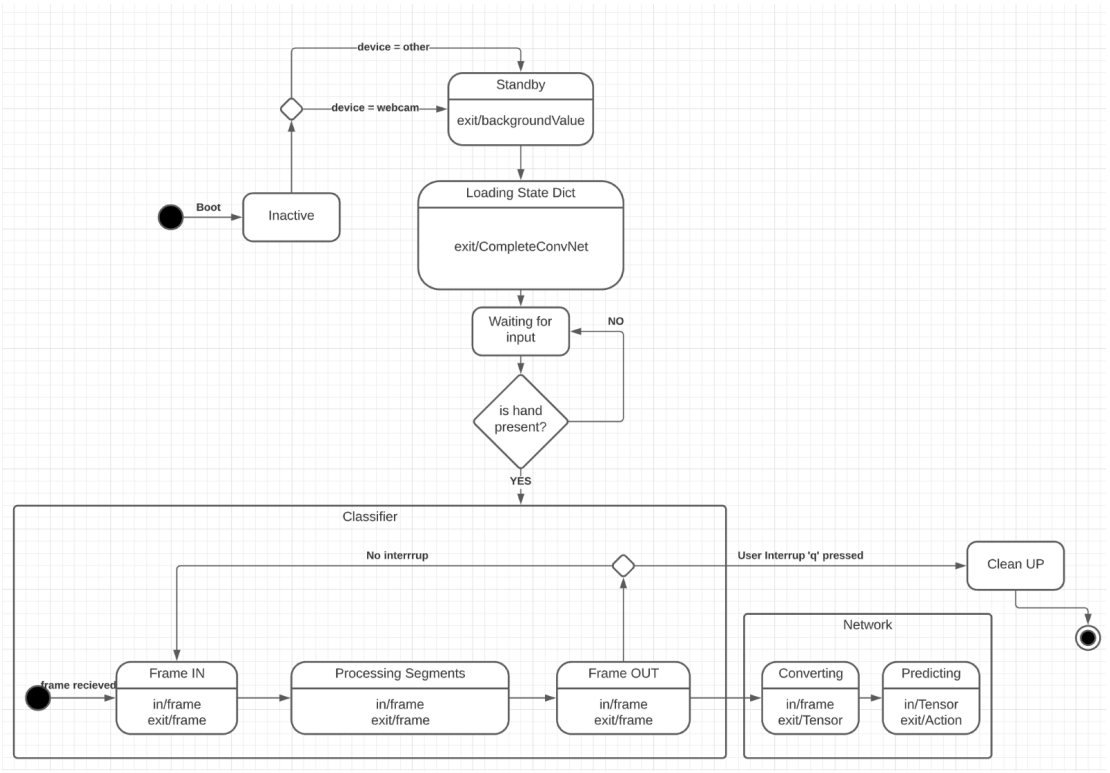


Fig 3. State Diagram

This comprehensive state diagram shows how our system operates state wise. Show what events get triggered and when.

Use Cases

Gesture Name:	Volume UP
Condition:	Camera is active and functional
Error Condition:	The gesture gets wrongly identified
System state on error:	Wrong action will be taken
Actors:	User and Camera
Trigger:	Palm shape ‘thumbs up’
Standard Process	1. The camera takes an image of gesture 2. The image is sent to a classifier 3. Image is turned to binary 4. Image sent to network

	5. Prediction and action is made
Alt process:	1. Wrong gesture gets identified 2. Network does the wrong action.

Gesture Name:	Volume DOWN
Condition:	Camera is active and functional
Error Condition:	The gesture gets wrongly identified
System state on error:	Wrong action will be taken, and wrong command is sent to system
Actors:	User and Camera
Trigger:	Palm shape ‘thumbs down’
Standard Process	6. The camera takes an image of gesture 7. The image is sent to a classifier 8. Image is turned to binary 9. Image sent to network 10. Prediction and action is made
Alt process:	3. Wrong gesture gets identified 4. Network does the wrong action.

Gesture Name:	5s forward
Condition:	Camera is active and functional
Error Condition:	The gesture gets wrongly identified
System state on error:	Wrong action will be taken
Actors:	User and Camera
Trigger:	Palm shape ‘V-shape right’
Standard Process	11. The camera takes an image of gesture 12. The image is sent to a classifier 13. Image is turned to binary 14. Image sent to network

	15. Prediction and action is made
Alt process:	5. Wrong gesture gets identified 6. Network does the wrong action.

Gesture Name:	5s backward
Condition:	Camera is active and functional
Error Condition:	The gesture gets wrongly identified
System state on error:	Wrong action will be taken
Actors:	User and Camera
Trigger:	Palm shape ‘V-shape left’
Standard Process	16. The camera takes an image of gesture 17. The image is sent to a classifier 18. Image is turned to binary 19. Image sent to network 20. Prediction and action is made
Alt process:	7. Wrong gesture gets identified 8. Network does the wrong action.

Gesture Name:	10s forward
Condition:	Camera is active and functional
Error Condition:	The gesture gets wrongly identified
System state on error:	Wrong action will be taken
Actors:	User and Camera
Trigger:	Palm shape ‘thumbs right’
Standard Process	21. The camera takes an image of gesture 22. The image is sent to a classifier 23. Image is turned to binary 24. Image sent to network

	25. Prediction and action is made
Alt process:	9. Wrong gesture gets identified 10. Network does the wrong action.

Gesture Name:	10s backward
Condition:	Camera is active and functional
Error Condition:	The gesture gets wrongly identified
System state on error:	Wrong action will be taken
Actors:	User and Camera
Trigger:	Palm shape ‘thumbs left’
Standard Process	26. The camera takes an image of gesture 27. The image is sent to a classifier 28. Image is turned to binary 29. Image sent to network 30. Prediction and action is made
Alt process:	11. Wrong gesture gets identified 12. Network does the wrong action.

Gesture Name:	Full screen
Condition:	Camera is active and functional
Error Condition:	The gesture gets wrongly identified
System state on error:	Wrong action will be taken
Actors:	User and Camera
Trigger:	Palm shape ‘expanded palm’
Standard Process	31. The camera takes an image of gesture 32. The image is sent to a classifier 33. Image is turned to binary 34. Image sent to network

	35. Prediction and action is made
Alt process:	13. Wrong gesture gets identified 14. Network does the wrong action.

Gesture Name:	stop/play
Condition:	Camera is active and functional
Error Condition:	The gesture gets wrongly identified
System state on error:	Wrong action will be taken
Actors:	User and Camera
Trigger:	Palm shape ‘squashed palm’
Standard Process	36. The camera takes an image of gesture 37. The image is sent to a classifier 38. Image is turned to binary 39. Image sent to network 40. Prediction and action is made
Alt process:	15. Wrong gesture gets identified 16. Network does the wrong action.

Gesture Name:	mute
Condition:	Camera is active and functional
Error Condition:	The gesture gets wrongly identified
System state on error:	Wrong action will be taken
Actors:	User and Camera
Trigger:	Palm shape ‘fist’
Standard Process	41. The camera takes an image of gesture 42. The image is sent to a classifier 43. Image is turned to binary 44. Image sent to network

	45. Prediction and action is made
Alt process:	17. Wrong gesture gets identified 18. Network does the wrong action.

Gesture Name:	null
Condition:	Camera is active and functional
Addition:	This is not a gesture just an array of zeros (black image) so that system can detect “no gesture case”
Error Condition:	The gesture gets wrongly identified
System state on error:	Wrong action will be taken
Actors:	User and Camera
Trigger:	Balck image, array of zeros
Standard Process	46. The camera takes an image of gesture 47. The image is sent to a classifier 48. Image is turned to binary 49. Image sent to network 50. Prediction and action is made
Alt process:	19. Wrong gesture gets identified 20. Network does the wrong action.

DESIGN

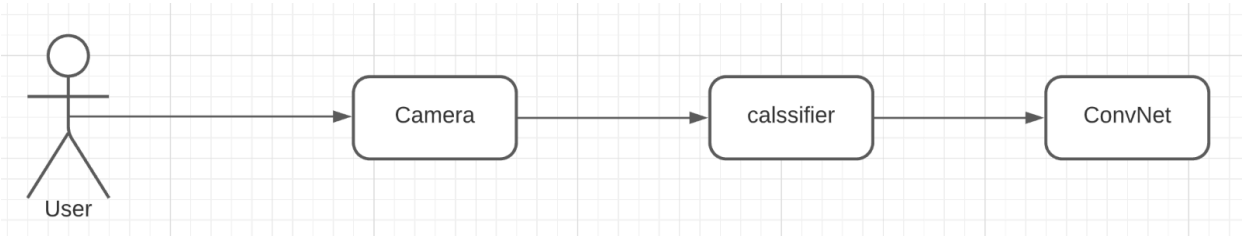
The hand gesture recognition system is one of the more basic systems for automation. We were still met with some limited functionality. We however realised that tools available to us were sufficient enough to execute the project.

Components:

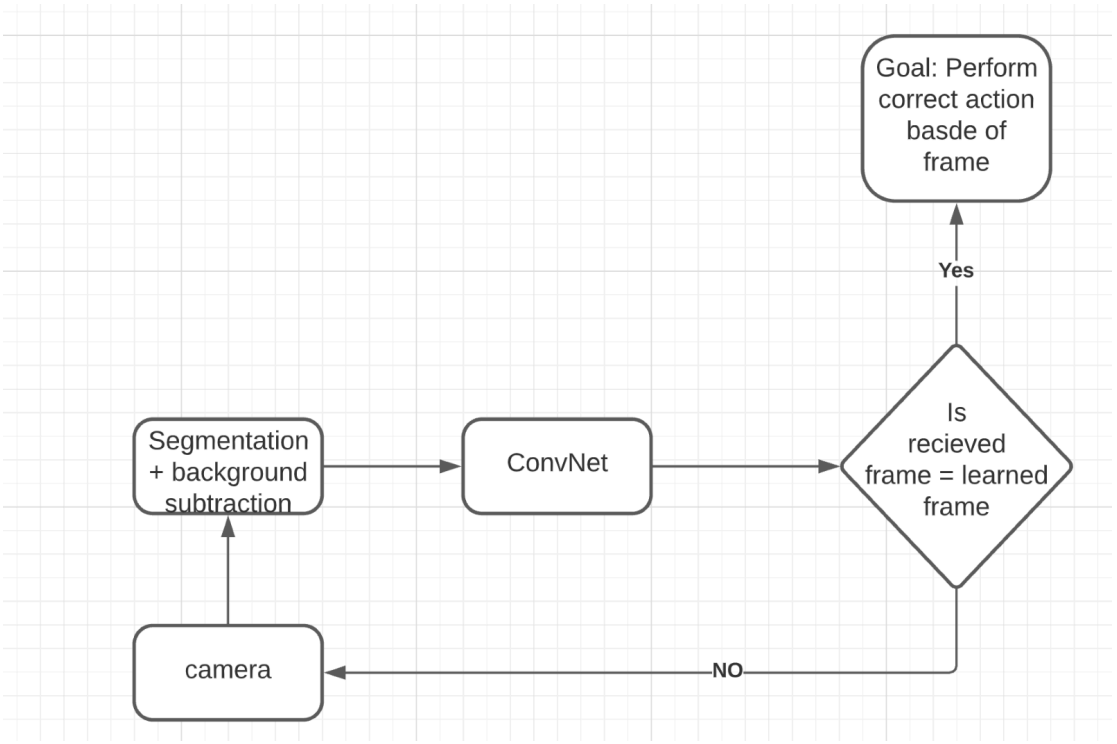
- Built-in computer camera for taking pictures
- Other cameras can be used as well
- GPU for network training
- Laptop

We didn't really need that many items as the system was mainly software oriented. We designed our own dataset of 10 gestures, each gesture has 2,461 images associated with it which gives 24,610 images in total. These images were binary images first converted to grayscale then blurred using Gaussian blur, they were used for training the network. These pictures were taken using python code. These pictures are binarized in real time and sent to the neural network to make predictions in real time. The conversion of image to binary is needed to simplify the image for the network. We also selected a specific area of the screen to read gestures from to reduce the number of pixels.

High level:



Low level:


Goal: Perform correct action basde of frame
Yes
Is recieved frame = learned frame
ConvNet
Segmentation + background subtraction
camera
NO

PROJECT PLAN

Luckily for our team, we designed an elaborative Gantt chart that incorporated the entire project schedule and division of labour among team members. Tasks were evenly distributed to reflect individual capabilities. It was transparent and easy to identify who does what and when.

Risk Analysis

Because of the constraints of time and the complexity of our project, we identified alternative projects that will be less challenging than our current project. We also realised that working on multiple gestures as high as 10 different gestures within this short period will be tedious hence If we don't make enough progress before week 3, we will abort this idea and focus on main gestures, so we will avoid wasting time.

It's equally important to note that we took into consideration challenges with regards to information sharing and storage so we resolved to work on cloud via GoogleDocs document. An efficient way to prevent potential risk of computer associated failures that will result in loss of files.

Code Overview

Cv2 segmentation

We made a function that is responsible for segmenting the image. This function takes in an image (frame) which is a numpy.ndarray of pixel values. The frames come from the camera's video stream. A square is cut out of the image, this is called the region of interest. The frame is turned to grayscale and gaussian blur is applied. Then background subtraction is applied together with contours finding (edge detection) using chain approximation.

In short, this class was responsible for number of things:

- Background subtraction
- Contour/edge detection
- Grayscale to binary conversion
- Returning binary image with contours

Background Subtraction

This class method collects the weight of the background so that it knows where the skin is relative to it. It needs the first 30 frames after boot to calibrate the desired weight so the gesture is correctly segmented. The contours are applied with the edge detection as well. The frame returned by this method is a binary frame that will later be sent to the ConvNet to make predictions on.

Build.py (Dataset Builder)

Our class was for building the whole dataset from collected images. This is a collection of methods that convert a jpeg image to an array of pixel data. This set is then saved for it to be used later. It also gives information about the progress of the build and counts how many images were successfully built. This is useful for tracking whether our training set is balanced. This is done to prevent overfitting.

ConvNet and choosing the best fit (numpy.ndarray to tensor converter)

This is a collection of methods. The frames returned by the image segmentation need to be passed to the network itself. First each frame is converted to form an ndarray to a tensor, a data type that the network can understand. These tensors are passed to the gpu along with the network itself. Then the network can make accurate and fast predictions on the received frames. The output returned by the network is a one-hot tensor/vector. This is then translated to a string using a python dictionary, these strings are sent to the pyautogui module and action is performed. This process is repeated for each frame in the video stream, for every gesture.

Testing (Confusion matrix)

		Predicted gestures								
	Gestures	Play/Stop	Mute	Null	10s forwards	10s backwards	5s forwards	5s backwards	volume down	volume up
Actuall Class	Play/Stop	2350	4	50	10	12	20	1	2	7
	Mute	5	2400	12	5	7	10	10	9	3
	Null	0	0	2480	0	0	0	0	0	0
	10s forwards	20	5	10	2331	23	12	45	2	13
	10s backwards	10	10	5	3	2397	15	12	3	6
	5s forwards	1	2	5	5	10	2395	15	9	19
	5s backwards	2	2	2	2	2	2	2405	20	24
	volume down	0	0	0	12	15	2	0	2422	12
	volume up	10	3	11	12	21	5	0	0	2399

POST-MORTEM ANALYSIS

Overall:

Since our first brainstorming, we came up with many ideas to make our project life-improving as well as revolutionary. We were so delighted to implement our ideas into reality. Unfortunately, due to lack of time, we couldn't make all of that come true. Current version of our system is only a simplified prototype, which still needs some improvements to make it more casual for users. However, we are proud of ourselves for what we made and we are even more excited about future development.
Due to the decision of choosing Python over MatLab we have also deepened our knowledge in this language. We learnt a lot about Cv2 image binarization & segmentation and pytorch machine learning library.

System Improvement:

As it is seen in our project demonstration video, the system tends to misinterpret some gestures because of factors such as various lighting, background colour or different angles of the hand. Therefore a bigger database set will definitely improve user's performance.

Another essential thing is a hand recognition feature which would allow our system to follow hand movement across the whole camera. We found it almost impossible to implement it in a 3 weeks period - it would take much more time for coding as well as for increasing the database. Since our main motivation was aimed to improve human's life and make controlling video players more comfortable, our current version demands from users to make gestures in a specific area which make it less affordable.

The MoSCoW model evaluation:

Prioritised requirements made our work much simpler. Thanks to the MoSCoW model we were aware of our most important tasks we had to focus on. Looking back on our priorities, we are satisfied with our work. Apart from one should have requirement, the system meets all of our should and must haves. As planned, 'Won't have' requirements were not implemented. In summary everything contributed successfully.

BIBLIOGRAPHY

1. http://www.gartner.com/document/2793317

2. https://www.gartner.com/en/newsroom/press-releases/2014-09-08-gartner-says-a-typical-family-home-could-contain-more-than-500-smart-devices-by-2022

3. https://www.ajperri.com/wp-content/uploads/2018/01/b0168809-7f07-40be-9a9c-aac85cca76d2-150716032045-lva1-app6891.pdf

4. https://core.ac.uk/download/pdf/143458516.pdf

5. https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.740.3117&rep=rep1&type=pdf