Destiny Robotics – Technical Paper

Lela Mirtskhulava - Chief Scientist, Ph.D
Erekle Shishniashvili - Senior AI Engineer
Jason Jordan - Chief Robotics Officer
Irakli Datashvili - Head Software Engineer
Megi Kavtaradze - Founder and CEO of Destiny Robotics

Contents

Part I
Destiny Robotics - Facial Expressions Recognition in Humanoid Robots

Abstract

– Humanoid robots are expected to be developed and operate in a close relationship with human beings, serve the needs of people, and work to create a connection between the user and the rest of the world. In order to bridge this gap of technology, these robots must be able to handle the wide variety of emotional states of humans Our goal is to create a next-generation robot that can interact with humans at a much deeper level than currently possible. Destiny will be able to identify human emotions and react compassionately. The resemblance to humans makes the communication process much easier with the robot. Our goal is to create a standard for robot-to-human interaction, making communication much more organic and efficient.

Introduction

Abraham Maslow, in his influential theory on the hierarchy of needs, proposed that our physiological needs are fundamental to our survival. The physiological need for social connection and belonging may be among the most basic human needs along with material items such as food or water. Research has linked the lack of social connection and belonging to higher risks of a variety of physical and mental conditions such as high blood pressure, heart disease, obesity, a weakened immune system, anxiety, depression, cognitive decline, Alzheimer’s disease, and even death [1]. For example, scientific research and study has found the general use of technology to be one of the largest drivers for the retreat of young adults from social connections and the cause of the physical and mental symptoms of obesity, depression and anxiety. There is an ever-growing opinion and popularity about social networks, automation, and a readiness of material possessions which assigns the decline of human connection among key demographics. Headlines like “Is Technology Making People Less Sociable?” [2] and “Is the Internet Bad for Society and Relationships?”[3] Point to technology in general, as a contributing cause of the feeling of lack of connection. Although the advancements in information technology, automation, and the physical sciences have prospered exponentially for decades, it is only within the last few years that missing technology of artificial human connection has been capable of closing the needed gap of technology and human need for belonging.

Destiny Robotics is the next generation of technology that can interact and connect with humans on a deeper level previously only considered in science fiction. Destiny is the merger of Artificial General Intelligence and Humanoid Robotics to solve the problem of human connection with current technological advancements. Destiny will be able to identify human emotions and react compassionately. Being available, trustworthy, and reliable, Destiny’s character and appearance will bridge the current gap of the lack of social connection as a result of the use of modern technology. The resemblance to humans makes the communication process much easier with the robot. Destiny will be the standard for Humanoid robotics for human interaction. Humans find it difficult to effectively connect with modern technology because they are not able to properly form the social connection humans require. Destiny facilitates a deep connection between humans and technology, which is an ideal mechanism for connecting humans with the global technological world in a deeply engaging manner.

Destiny, recognizing emotional signals from humans will lead to a huge advancement in humanoid robotics and the growth of facilitating a deep connection between humans and technology. In addition to emotional support and interaction, Destiny will also have to be able to follow instruction and complete routine tasks. In order to achieve these tasks, the Destiny robot will require software capabilities of computer vision and natural language processing among the other functional elements of the robot. Humans can easily recognize other human emotions in a multitude of situations through social communications, visual communication and verbal interactions. Computer scientists have developed the technology of deep neural networks giving the possibility to read human expressions (emotions) through facial analysis, interpret environmental conditions, and categorize and contextualize verbal speech. Facial expression recognition and Natural Language Processing is very important in the field of artificial intelligence and robotics [4]. These two fields of computer science will therefore be very important to the development of the destiny robot.

A classical approach to facial expression recognition is the Facial Action Coding System (FACS) which was developed by Ekman in 1978. In this classical framework, movement of the facial regions are described as AUs (Actions Units) describing deviations from a neutral expression. Human emotions are not only expressed in words or verbal expressions, but also the deviations from neutral expression. The Destiny robotics team will extend the classical approach of the Facial Action Coding System to include the latest in convolutional neural network technology and artificial intelligence.

In order to understand a certain emotion, a person will use nonverbal expressions along with verbal expressions. In verbal expression, we include the words with the sentences but in nonverbal can be included tone of the voice or facial expressions. Current technology uses voice assistant machines to process verbal communication. Virtual Private Assistants such as Google Assistant, Siri and Alexa have become a part of the lives of humans. Some of the previous works focused on detecting emotions using verbal or nonverbal emotional expressions have been what has been called Multimodal Recognition Systems.[5][16] The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) for model training is among these techniques which the Destiny robotics team will further develop and exploit to train and recognize verbal and non-verbal emotional detection.

Robot software systems are complex. This complexity is due, in large part, to the need to control diverse sensors and actuators in real time. Robot systems must work to achieve tasks while monitoring for, and reacting to, unexpected situations. Doing all this concurrently and asynchronously adds immensely to system complexity. The use of a well-conceived architecture, together with programming tools that support the architecture, can often help to manage that complexity. It is important to understand strengths and weaknesses when choosing an architectural approach for robot application.[6] When individuals talk about fully functional humanoid robots, the images are a multitude of how these robots should be functioning and behaving, what tasks robots should perform and in what conditions. However, there are several main functionalities in which the Destiny robot will be designed and developed. That is, Destiny should be able to physically perform tasks normally designated to humans and complete these tasks within the form familiar to human function.

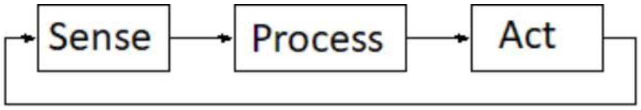
About This Document

This document is intended to provide a fundamental understanding of the physical hardware features and software capabilities of the Destiny Robot. Although the Destiny robot will be a fully mobile humanoid robot with functional capabilities mimicking human capabilities, specifically, this document is intended to address the design of the Destiny robot for the purpose of facial expression, to detail the muscle groups for facial expression and recognition. The hardware control, operation of facial expression, software architecture and implementation are all address to address the value proposition of developing a human connection with robotics. Features not contained in this document include the specific software design, the development process from prototype to production, and deployment of robotic limbs such as hips, arms, and feet with only exceptions to include such details that reference accurately describe the facial expression and recognition. In addition, sub-system architecture, design and deployment will also be referenced as “being used” but not detailed within this document such as power systems, memory and cloud services. This document also does not contain the complete design specifications, materials, and methods of the Destiny Robot head and shoulders. The intent is to provide a fundamental understanding of the construction, materials, and methods in the key areas of interest which is the expression of facial expression and recognition for Destiny robot. All information, designs, and ideas contained within this document are confidential and proprietary to Destiny Robotics unless otherwise cited as referenced material. All materials, ideas and concepts shall remain confidential unless otherwise specified by Destiny Robotics Corporation.

Robot Control Architecture

Software to Hardware Layers

One of the most used architectures for robotic operation is the basic function analyzing processing received data and reacting relatively. After the operation completes, the robot runs the same software loop again. Using sense-process-act functional architecture allows the robot to gather data and respond to the user or environment. The picture below demonstrates the sense-process-act loop.



Here is an example showing subsystems controlled by each of the part of the architecture blocks:

Sense (user input)

- Audio
- Video
- Kinect
- Data from IoT devices

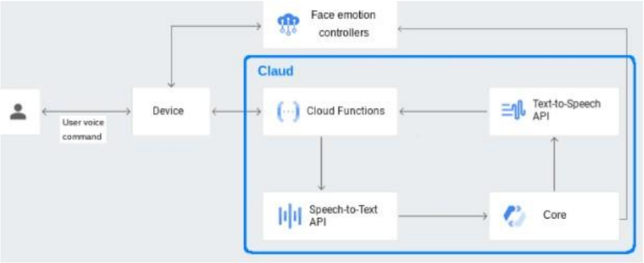
Process

- Speech to text API
- Analyze and Planning
- Text to Speech API
- Voice and Tone API
- Data logger API

Act

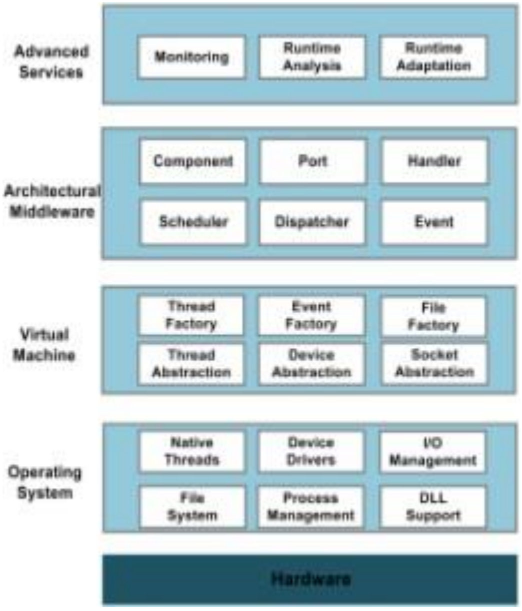
- Task execution
- Motor control
- Play Audio
- Control power systems

Sense is the basic function responsible for gathering input data from the user, the environment or from another third-party API. These sensors could be video and audio files or tactile and temperature sensors. The sense feature is the basic part of the robotic system needed to have a tight connection with the status of the robot. Process is the block that combines internal and external data sources to administer the best way to analyze, react and generate relative output. Some process block functions are deterministic, and some require the processing of neural networks. The Act bock starts basic reactions and actuates all necessary parts to make the robot mimic human-like behavior and motion. At a high level the diagram demonstrates an example of single elementary Sense-Process-Act procedure.



Software to Hardware Layers

There are several layers to connect hardware to software which also back-propagate through the Sense-Process-Act function. The first layer begins at the hardware and continues to the operating system, virtual machine, architectural middleware, and finally to the advanced Services. Each layer has sub systems and an array of the control functions necessary for handling synchronized complex data exchange. The hardware is the lowest level presented to the user or environment containing



the base sensory devices such as vision or radio signal. The operating system is responsible for continuous data processing from the hardware level, receiving, processing, storing and managing input and output signaling. The operating system contains such features as the native threads, device drivers, IO management, file system, process management and supporting libraries. The 3rd layer or virtual machine is designed to contain the thread factory, event factory, file factory, thread authentication, device administration, and socket administration. The Architectural Middleware is constructed to contain components, ports, handlers, schedulers, dispatchers, and events. Finally, the 5th layer is the Advance Services. The advanced services are responsible for monitoring, runtime analysis, and runtime adaptation.

Connecting Components

All the architecture components need to communicate with each other.[7] They need to both exchange data and send commands. The choice of how components communicate (often called the middleware) is one of the most important and most constraining aspects of a robot architecture.[6] There are two basic approaches the Destiny Robotics team will explore to serve as the architecture middleware; the client–server and the publish–subscribe. The client–Server (also called a point-to-point) communication protocol, is where components talk directly with other components. An example of this is remote procedure call (RPC) protocols in which one component (the client) can call functions and procedures of another component (the server). A modern, and popular, variation on this is the common object request broker architecture (CORBA). CORBA allows one component to call object methods that are implemented by another component. All method calls are defined in an interface definition language (IDL) file that is language independent. The Publish–Subscribe (also called a broadcast) protocol, a component publishes data, and any other component can subscribe to that data. Typically, a centralized process routes data between publishers and subscribers. In a typical architecture, most components both publish information and subscribe to information published by other components. There are several existing publish–subscribe middleware solutions [6].

Planning

The planning component of our prototype layered architecture is responsible for determining the long-range activities of the robot based on high-level goals. Where the behavioral control component is concerned with the here-and-now and the executive is concerned with what has just happened and what should happen next, the planning component looks towards the future. In our running example of an office delivery robot, the planning component would look at the day's deliveries, the resources of the robot, and a map, and determine the optimal delivery routes and schedule, including when the robot should recharge.[6] The planning component is also responsible for replanning when the situation changes. For example, if an office is locked, the planning component would determine a new delivery schedule that puts that office's delivery later in the day

Processing of Single User Request Example

Consider a robot that operates in a typical way on a user's verbal request. The behavioral control layer contains the control functions necessary to listen to the user and respond relative to the content the user said. Next behaviors for this robot will be executed with following order:

- Verbal Requester
- Voice recognition API
- Speech to text API
- Server requester (Http, TCP) (sends data to server)
- Server Processor
- Server Responder
- Text to Speech API
- Voice and Tone API
- Emotion API
- Verbal Responder
- Emotion Responder

The above-described steps are dedicated for the single user request. Each actions robot will have supposed to be following similar stepped function calls for communicating with the user.

Extendable APIs wrapping Internal Functions

The following are examples of the wrapper framework that is used in programming to convert data into a compatible format and simplify a complex task easily using what is known as abstraction.

Smart Home API

- Light Control
- Volume control
- Audio (play music)

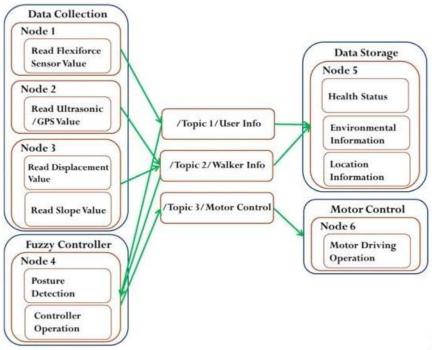
Device API

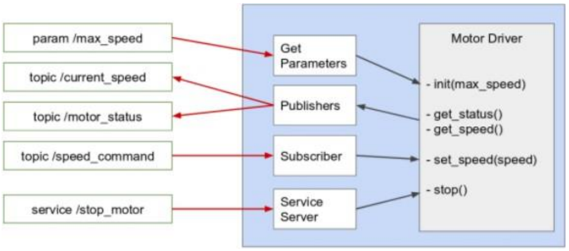
- Get Data from smart watch (respond related to the data)
- Activate RoomBot.
- Start dishwasher when people are out.

Update API

Detailed Single Motor Control Class and Its Associated Members

The following is demonstrating the calls handling the processes coming from the software. While these are typically hand-crafted functions written for executing specific actions, they are key parts of the code to make actuators and motors move.

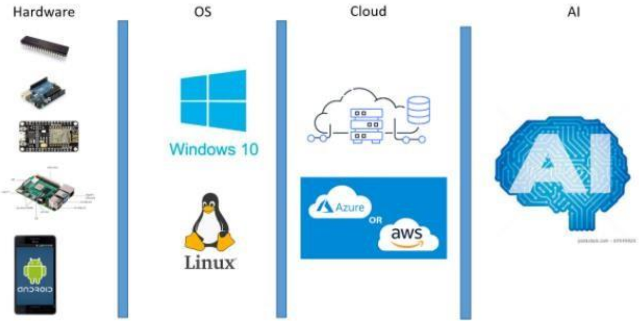


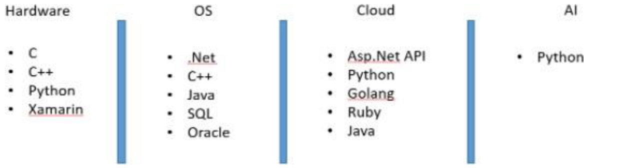


Data receiving and processing steps

Presented below are multiple layered structures of hardware and the respectively supported languages on each layer which make the Destiny robot system. Each layer has a relative language that is best at handling and supporting functionality of the layer. Hardware layer operates on C and C++ libraries for quick retrieval and processing. The Artificial Intelligence neural network makes calculations using the Python language and libraries. The below diagram demonstrates the environmental parts responsible for different operations.

The following are the languages supporting each layer:





Hardware	OS	Cloud	AI
• C • C++ • Python • Xamarin	• .Net • C++ • Java • SQL • Oracle	• Asp.Net API • Python • Golang • Ruby • Java	• Python

ROS (Robot Operating System)

Robot Operating System (ROS) is an open-source, meta-operating system for robots. It provides the services you would expect from an operating system. ROS can be used in building and simulating robotics applications, as well as unmanned ground vehicles and simultaneous localization and mapping (SLAM). To facilitate better integration within the ROS ecosystem. Microstrain has develop an open-source License free (MIT License) series of drivers specifically designed and tested for ROS.[9] The features of the Robot Operating System are hardware abstraction, low-level device control, message-passing between processes for commonly used functions, and package management.[10]

The above diagram demonstrates an example of the ROS implementation. VSCode is one example of the tool supporting multi-language IDE for the ROS user interface.

Destiny Vision and Natural Language Processing

Computer Vision

In the design, development, and deployment of a fully functional humanoid robot, there are several main functionalities the Destiny robot will require.

- Emotional Support/Interaction,
- Ability to follow instructions
- Ability to take care of repetitive tasks.

To achieve all these functionalities, the Destiny robot will have to utilize 4 fields of artificial intelligence

- Computer Vision
- Natural Language Processing (NLP)
- Audio Processing
- Data Science/Data Engineering

Those 4 fields can be part of one module if it is required, or 1 field can be 1 module, depending on the specific requirements.

Computer vision is one of the most important features of the Destiny robot. To give the robot ability to have eyes and see the world around it, the Destiny robotics team must use computer vision. But for the purpose of this document, the concentration will be on Facial Detection and Recognition, Emotion Recognition, Activity Recognition and Danger Recognition. Facial detection and recognition are used to identify with whom the robot is talking, so that it can perform some specific tasks related to this person, or not perform at all if the person does not have the privilege. For Emotion Recognition, the Destiny robot will need to provide emotional support to the user, the team will deploy an algorithm that detects the emotion of the person. For Activity Recognition, the Destiny Robotics team will also design the robot to have the ability to detect a person's actions to give better tips for certain actions and be able to have a sense of communication.

In the case of danger recognition, the robot will use computer vision to identify some of the dangerous areas for the home, such as small fires or broken parts on the floor.

Natural Language Processing

Communication and ability to understand as well as generate meaningful responses is a crucial part of our humanoid robot. To utilize the ability to understand the context of the given information, the Destiny team will need to use NLP techniques so that the robot is able to perform the tasks beyond emotion recognition such as chatbots mixed with voice assistants, buying supplies, scheduling daily tasks, and understanding basic commands. Chatbots mixed with voice assistants is the function so the robots will be able to communicate with humans, have a dialogue, understand the sentences said to them and return meaningful answers. This can be used in two ways, as technical support or having emotional dialogue. When buying supplies, the Destiny robot will require the ability to perform static tasks, such as ordering monthly supplies from the supermarket and analyzing for good offers from nearby markets. Scheduling daily tasks would incorporate saving notes for a person in a manner that is efficient and understandable such as: “I want tomorrow afternoon to clean my room”, the robot should be able to convert the task as: “Tomorrow 7:00 PM - Clean the room.” Understanding basic commands is one of the most useful things in humanoid robots and home assistants. Given this ability to process basic commands, Destiny will help the user interact with the environment. For example: “Turn on the Music”, “Turn off the TV”, “Make the house warmer”.

Audio Processing

We have already discussed the ability to communicate with robots, but that does not mean someone sitting in front of it typing texts and reading outputs on the computer screen. Even in the 20th century when people imagined robots, they inherently assumed it would have the ability to speak and hear. On a technical level, the ability to speak comes from Audio Processing, but there are several other ideas we want to include using the audio.

Speech to Text (STT) - as mentioned above this includes ability to hear, meaning convert speech to text and then using NLP techniques above to analyze context. This technique should be very well made as there is a lot of noise, background sound in the house, so to understand the command correctly and not to perform another task, precise STT is a basic requirement.

Text to Speech (TTS) - We not only care to hear, but also to speak. So, robots should be able to output meaningful text using NLP, and then use TTS to generate clear audio.

Emotion Analysis - We mentioned in the computer vision technique to use a camera to analyze the emotion of the human, so that the interactive emotional relationship between the robot and human is as close as possible. However, we think that using the voice of the human combined with the input stream from the camera can give us higher accuracy and be more precise when detecting emotions which will allow robots to return better responses and make more sensible analysis.

Identification - We can use the speech of the human to identify people. This will give us the possibility to have higher security and control towards some special activities that will be accessible for the certain group of people that have the privilege from the owner.

Data Science/Engineering

As the humanoid robot becomes a member of daily life, there is a lot of statistical/tabular data that we can gather. There are several repeated actions, seasonal actions, and time series that we can study using techniques from data engineering. For example, in winter it's more likely to go skiing and in summer to swim. Or during weekends, it is more likely to visit the bar and be more excited. We can use all this knowledge to make some of our above-mentioned abilities of people better.

Emotion recognition - While we plan to use speech as well as camera to detect emotions with very high accuracy, tabular data can be also very helpful during the week. There is a special routine that people follow and by analyzing it we can also analyze the emotion flow. For example, if a person has a lot of work to do on Wednesday, he is more likely to be stressed at the end of the day, than on Friday after finishing all work and getting ready to relax during the weekend.

Favorite things to do - by gathering information of what a person is doing on a daily scale, we can analyze what are his/her favorite behaviors, food, places and robots can recommend pleasurable activities based on that.

Time Series - People often require assistance or help with very basic but hard to acquire skills in life, such as assistance with money spending. By analyzing previous monthly, or yearly spending, robots can also analyze if a person is spending too much and recommend ways on how to save money or just convince the person to decrease the spending.

Face Recognition

Introduction

After years of research of classical computer vision algorithms, it quickly became obvious that face detection systems were reality [33]. Different Computer vision algorithms managed to achieve very good accuracy on face detection even without utilizing any data or statistical methods. But face detection alone is not a very powerful tool in identifying the person standing in front. But in recent years Google also made face recognition an easy task after introducing Face Net [32], an approach that could recognize people from a huge database in real time. The algorithm architecture introduced became very simple and easily understandable compared to the models [34][35], which tried very complex systems of multiple stages combined with PCA for dimensionality reduction and then using SVM for assigning classes to each face. Even though Face Net was released in 2015, which with the advancement of Deep Learning models might seem very old, the algorithmic idea that they offered, was so unique and simple that new state of the art models such as [36],[37],[38] can be integrated in an architecture to increase the accuracy while again leaving all the processing in the real time.

Method

In this section we will try to explain on a high level how Face Net architecture works. As stated, many times, it is simple. Convolutional neural networks are building blocks of today's [36-38] computer vision. Even though recent trends of transformer networks in computer vision tasks give promising result [39], still deep convolutional neural networks remain the most widely used and utilized algorithms for computer vision tasks, such as classification, object detection, segmentation, etc. [38]. Face Net also uses basic CNN architecture. It acts as a classification network without a classifier layer in the end, meaning we don't extract classes but instead we are interested in the embedding of the CNN. Extracted embedding represents features of the image. In our case features of the face, such as eye color, nose shape, head shape etc. So basically, the whole head of the person will be transformed into a much smaller feature space. For example, let's say we have an image of a face which has resolution of 28x28x3, meaning its width and height are 28 red, blue, and green pixels. Now we cannot directly compare two images of this size, because it's computationally too complex. Instead, we feed face images into CNN and get a result vector which has size of 3. Out of 3 in the vector we can have any number, for example [0, 1-3]. For simplicity we can say that the first element represents the color of the eye, the second one is the shape of the nose and the third one is the shape of the lips. Now we know that for example a person's eyes correspond to color “0” which will be decided by an algorithm which class to assign “0” to (Black for example). As discussed above, the main goal of the CNN is to understand what the main features of the face are and compress it too much smaller space, so that we can easily calculate afterwards who this face belongs to.

After using CNN to get the features of the image, now it's time to find similar faces in the database. Database is where face images of all the people who should be recognized are saved. In our case we can say for example images of family members, neighbors, or close people. To make it more scalable, all the images from the database are in advance fed in CNN and only feature vectors are saved in the directory, which are very efficient to keep and does not at all take much space.

Final phase is to find the detected face from a humanoid robot in the database. The pipeline as discussed previously is as follows: we detect the face; we extract the face and feed into the CNN to get the features of the face and finally we compare this feature to all the information in our database to find the corresponding match. There are several ways we can compare feature vectors to each other. We approach this problem as nearest neighbor clustering or nearest neighbor classification problem from machine learning and any statistical algorithm will perform very well if we have good architecture for the CNN. However, in our case we use C-Support vector classification which is very efficient and performs its calculations in the real time. This is our basic pipeline for Face recognition, but first, we also must talk about how we train CNN to get similar embeddings for the similar faces, while generating distant embeddings for different faces.

Training

Our main goal of face detection is to have a CNN, that would give us very similar feature embeddings for the same faces, while generating different feature embeddings for different faces. To achieve this Face Net [32] uses triplet loss to train their CNN. Triplet loss means that on each iteration of training the algorithm, they take 3 images. Out of those 3 images, 2 belong to the same person, or are very similar while the 3rd one is very different. Now, they train CNN to give very similar embeddings for the 2 faces that look very similar, while generating embeddings for different faces as far away from each other as possible. For simplicity, we can use any measuring step, for example Euclidean distance. Imagine we have 2 same faces, and we tell our CNN to create embeddings for them where Euclidean distance between the feature vectors is 0, while keeping in mind to create embeddings for different image where distance would be 1. After training for several hours, we would have an algorithm in our hands that can almost perfectly align similar faces and differentiate them with non-similar ones. Hence the name triplet loss, comes from the method, where we use triplets to train our algorithm, three images which are fed every iteration.

CNN Architecture

As we discussed previously one of the big advantages of the method is that we can use any architecture for face recognition. After several months, many techniques to train CNN were introduced, starting from skip connections [40] to depth wise convolutional blocks [38] and algorithmically aligned parameters [36]. Recently we also had an outbreak of transformer blocks on image recognition tasks [8]. But as said, we can use any architecture for our case. We can take a state-of-the-art model with the highest efficiency, so that it performs in real time. Retrain it with triplet loss and we will get very high accuracy on face recognition task including all the new innovations and additions that were introduced in recent research.

Face Recognition Summary

As we see, architecture introduced by Face Net [32] stands out in the sense that it's very flexible to changes and performs very well in real-time. We can utilize the power of new research and new deep learning architectures for higher accuracy and more efficiency. Using this algorithm, we are confident to implement a highly accurate face recognition pipeline, with our custom changes in it to make it more robust and efficient for humanoid robot purposes.

Facial Expression and Recognition

Facial Action Coding System (FACS)

Due to its scientific objectivity, FACS has become the most well-spread and popular system worldwide. It is a detailed, 500-page tutorial on how to read faces. It contains a detailed analysis of possible facial muscle movements, their combinations, and the nature of their performance. It aims to train a system to recognize different combinations at a different speed and with different degrees of manifestation (up to barely noticeable and very fast ones). The tutorial provides photo and video examples and practical exercises.

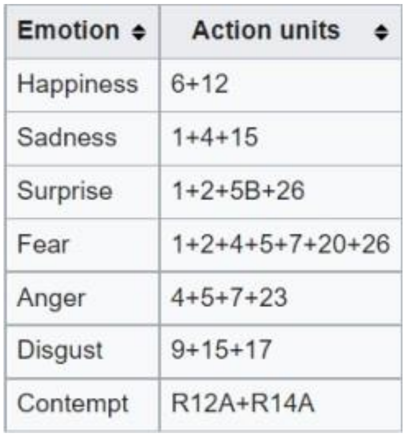
According to this FACS system, facial expressions are divided into three types:

- Macro-expressions are the daily routine expressions, usually obvious to all the sides of a communication act. They last between 0.5 and4 seconds.
- Micro-expressions are short, less than 0.5 seconds, involuntary facial expressions appear when a person is trying to hide or suppress the emotion. Micro-expressions cannot be consciously controlled.
- Subtle expressions are emotional responses to an event, environment, or another living being. Subtle expressions are not intensified and often mark the moment when a person starts feeling an emotion.

Electromyography Method

This method uses surface electrodes and enables the diagnostics of bioelectric potentials when reducing muscle fibers. During EMG, the nerve-muscular endings have varying potentials, which are then recorded.

EMG was designed to reveal those human emotions and mental states that cannot be tracked visually on our faces. Even when a person tries to avoid or conceal an emotion, the device will still record subconscious changes in the brain.



Emotion ⬍	Action units ⬍
Happiness	6+12
Sadness	1+4+15
Surprise	1+2+5B+26
Fear	1+2+4+5+7+20+26
Anger	4+5+7+23
Disgust	9+15+17
Contempt	R12A+R14A

Automatic Face Recognition

Automatic face recognition is a practical application of image recognition. Its purpose is to automatically localize the face in the photo or video and identify the person by face. The photo identification feature is already actively used in photo album management software and mobile devices' unlocking systems.

This method is based on a special algorithm that creates face signatures. They are generated using different physical features of the human face. This information is then transformed into a mathematical formula and run through face databases or compared with the given examples.

Face Analysis

After detection and recognition, a photo will capture the face and will then be analyzed.[24] The majority of face recognition technology uses 2D images instead of 3D. This is because 2D photos are more readily correlated with public photos or pictures in a database (these are typically 2D as well).

During analysis, the face will be separated into distinguishable landmarks – we can call these nodal points. A human face has eight nodal points. Face recognition technology will analyze each of these points – for example, the distance between your eyebrows.[24]

Convolutional Neural Network for FACS Recognition.

VGG-16 Architecture

Input. VGG takes in a 224x224 pixel RGB image. For the ImageNet competition, the authors cropped out the center 224x224 patch in each image to keep the input image size consistent.[12]

Convolutional Layers. The convolutional layers in VGG use a very small receptive field (3x3, the smallest possible size that still captures left/right and up/down) [12]. There are also 1x1 convolution filters which act as a linear transformation of the input, which is followed by a ReLU unit [13]. The convolution stride is fixed to 1 pixel so that the spatial resolution is preserved after convolution.

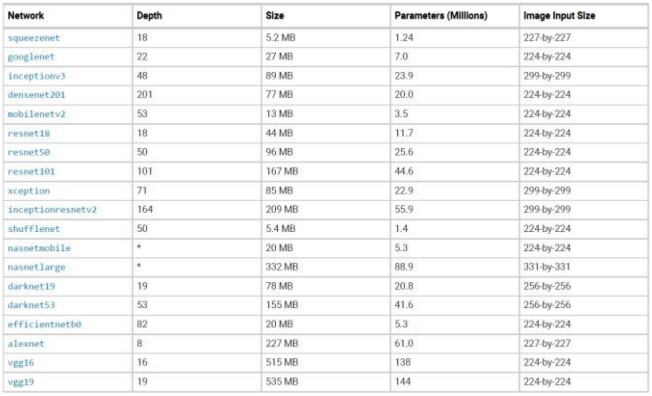
Fully Connected Layers. VGG has three fully connected layers: the first two have 4096 channels each and the third has 1000 channels, 1 for each class.

Hidden Layers. All VGG's hidden layers use ReLU (a huge innovation from AlexNet that cut training time). VGG does not generally use Local Response Normalization (LRN), as LRN increases memory consumption and training time with no increase in accuracy [12].

A good deep neural network is fast and has high accuracy. The pretrained image classification network has already learned to extract the most powerful features from the images using them as a starting point to learn the given task. Such networks already have been trained on more than thousands or a million images. They can classify more than 1000 object categories into images. So, Using a pre-trained network is typically much faster with transfer learning than training them.

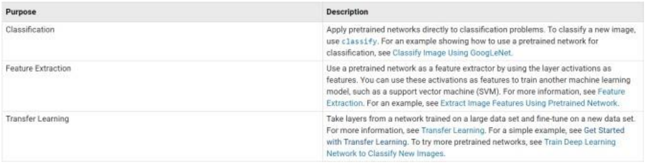
We are experimenting with using different pretrained networks according to their characteristics like network size, accuracy, and speed which gives us a possibility when we need to choose the most effective deep neural network to apply to our problem. The accuracy of classification measured for the ImageNet validation set is the a common way of measuring the network accuracy trained on ImageNet because if they are accurate on ImageNet they often are accurate on other image data sets using feature extraction or transfer learning. However, high accuracy on ImageNet does not always transfer directly to other tasks, so it is a good idea to try multiple networks. Sometimes more effective way is to use a set of multiple models because even highest accuracy on ImageNet cannot be transfered to other given task.

Table: Pretrained networks properties [https://www.mathworks.com/help/deeplearning/ug/pretrained-convolutional-neural-networks.html]



Network	Depth	Size	Parameters (Millions)	Image Input Size
squeezenet	18	5.2 MB	1.24	227-by-227
googlenet	22	27 MB	7.0	224-by-224
inceptionv3	48	89 MB	23.9	299-by-299
densenet201	201	77 MB	20.0	224-by-224
mobilenetv2	53	13 MB	3.5	224-by-224
resnet18	18	44 MB	11.7	224-by-224
resnet50	50	96 MB	25.6	224-by-224
resnet101	101	167 MB	44.6	224-by-224
xception	71	85 MB	22.9	299-by-299
inceptionresnetv2	164	209 MB	55.9	299-by-299
shufflenet	50	5.4 MB	1.4	224-by-224
nasnetmobile	*	20 MB	5.3	224-by-224
nasnetlarge	*	332 MB	88.9	331-by-331
darknet19	19	78 MB	20.8	256-by-256
darknet53	53	155 MB	41.6	256-by-256
efficientnetb0	82	20 MB	5.3	224-by-224
alexnet	8	227 MB	61.0	227-by-227
vgg16	16	515 MB	138	224-by-224
vgg19	19	535 MB	144	224-by-224

Table: Tasks Description [https://www.mathworks.com/help/deeplearning/ug/pretrained-convolutional-neural-networks.html]



Purpose	Description
Classification	Apply pretrained networks directly to classification problems. To classify a new image, use classify. For an example showing how to use a pretrained network for classification, see Classify Image Using GoogLeNet.
Feature Extraction	Use a pretrained network as a feature extractor by using the layer activations as features. You can use these activations as features to train another machine learning model, such as a support vector machine (SVM). For more information, see Feature Extraction. For an example, see Extract Image Features Using Pretrained Network.
Transfer Learning	Take layers from a network trained on a large data set and fine-tune on a new data set. For more information, see Transfer Learning. For a simple example, see Get Started with Transfer Learning. To try more pretrained networks, see Train Deep Learning Network to Classify New Images.

Related Work

A lot of work on emotion recognition in the human voice. Timbre, loudness, pitch, and tone are very important for detecting emotion. These features vary across different emotions [17]. Even though the variations of these variables depend on personal vocal characteristics, it was observed that in the nervous or panicked emotional state, mean values of pitch, tone, the time between words and timbre ascend increases [17]. Various models ranging from HMM (Hidden Markov Models) to RNNs (Recurrent Neural Networks) have been used to detect human emotions. Convolutional Neural Networks have tackled image classification problems and they can also be used for emotion recognition in the human voice.

Spectrograms are frequently used with timeseries data [23]. As we mentioned in the abstract above, the audio data is transformed into Mel Spectrograms and then is fed to Convolutional Neural Networks [25-26]. Mel Frequency Cepstral Coefficients are widely used for speech recognition, and they can also be applied for solving the emotion recognition task [18], [20], [25-26]. As Mel Frequency Cepstral Coefficients are time-series data they need to be transformed and framed, so that they can be fed to Convolutional Neural Networks.

The results differed for these different models. Some researchers decided to combine both speech features and transcriptions. The results showed that the combined CNN models based on both text and speech features had the highest overall accuracy, namely, 75.1% (Text & Spectrogram) and 76.1% (Text & MFCC) [25]. Several different models have been compared: LSTMs, CNNs, HMMs, DNNs and respectively different input data was fed to them: Log Mel Spectrograms or MFCCs, Log Mel Spectrograms (LSTMs), Log Mel Spectrograms or Pure Audio or MFCCs (HMMs) and MFCC and Deltas (DNN). The best results were achieved using CNNs with Log Mel Spectrograms' features [26]:" On the 14-class (2 genders × 7 emotions) classification task, an accuracy of 68% was achieved with a 4-layer 2 dimensional CNN using the Log-Mel Spectrogram features." Simple ANNs used and neural network pattern recognition were used for supervising training and testing processes. The data contained German sentences and 7 different emotions. Feeding as an input MFCC + Cepstrum+ Frequency scaled MFCC showed the best results (85.7%). But the model completely failed in recognizing the sad emotional state. [20]

Detecting emotions is very important to improve IoT voice-enabled services. From another perspective, voice signals convey important information about the user that can compromise user privacy. So, it is important to maintain user privacy as well as to improve IoT voice-enabled services using emotion recognition. GANS (Generative Adversarial Networks) can be used to solve this issue using voice conversion [14]. Detecting emotions in the human voice can improve various applications. Humans' emotional state can change HRV (Heart Rate Variation). Human speech characteristics, including emotional state, can be used to measure HRV [18]. Classical machine learning models have very poor performance when it comes to emotion detection [18]. Some studies focused on detecting emotions using both vocal and visual features (multi-cue approaches) [22-23].

In order to recognize emotions in a human's voice several datasets have been developed. The emotion labeling process can be quite difficult and mislabeled data can lead to poor performance. Datasets must be well-prepared in order to achieve good results when using them for training models. Some of these datasets are the Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS) [21], Suppers Brain Lab Psychotherapy EEG Dataset [16][24], and Indian EmoSpeech Command Dataset [15][19]. All these datasets are available for free [19], [21], [24].

METHODS AND DATA

Model

Since convolutional neural networks had the best results for identifying emotions, for training model 2D Convolutional Neural Networks were used. In order to answer the second research question, we decided to test both max pooling and average pooling layers after Convolution layers and compare their results. The Convolutional Neural Network consisted of 3 or 4 convolution blocks. The ReLu, ELU, and SoftMax activation functions were used for convolutional, activation and output layers respectively. More precisely, in the first two layers the ELU activation layer was used after the batch normalization and no activation was used in the convolution layer while on the rest of the layers no activation layer was used and activation in convolution layer was set to ReLu. In the case of 4 block Convolutional Neural Networks, for the 2nd and 3rd blocks dropout rate was set to 0.15 and after the 4th block to 0.3. 3 block CNN was the exact copy of 4 block CNN from which the 1st block was removed.

The SoftMax activation function is widely used for classifying multiple classes and this function was used as an activation in the final dense layer. In order to implement the models, Tensorflow framework is used.

The models for the speech and song datasets were almost the same. The only difference was in the output layer since the song data contained only 6 emotional states while the speech data contained 8 different classes.

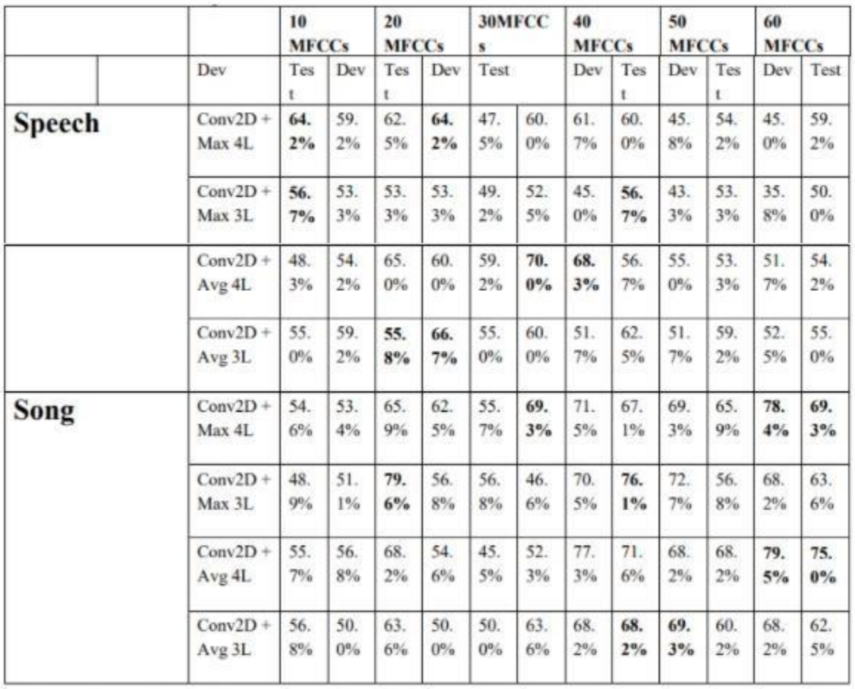
All models were trained for 100 epochs and in the case of the song data if more than 68% validation accuracy was achieved, training was stopped and in the case of the speech data it was stopped if more than 64% was achieved.

Dataset

In this paper, the RAVDESS dataset was used. To the best of our knowledge, in the previous model song data wasn’t used for predicting emotions, and in this paper, we decided to use both speech and song datasets. The speech dataset contains 8 emotions: happy, fearful, calm, disgust, neutral, sad, surprised, and angry. In the dataset 12 actors and 12 actresses repeated two separate sentences for these different emotions. The song dataset contains 6 emotions: happy, fearful, calm, neutral, sad, angry. [21]

Data Preparation

In order to answer the second research question, we decided to try out a different number of MFCCs: 10, 20, 30, 40, 50, 60. Each sample for each actor/actress was trimmed (all sound below 30 decibels was considered as silence), then MFCCs are extracted, and they were padded with 0s to maintain the same size (4.3 seconds for speech and 5 seconds for song dataset samples). The data was divided based on the emotions and the gender of the actors/actresses wasn’t considered. The data was split into train/validation/ test sets in the following way: 10 actors and 10 actresses’ voices were used for the training set and 1 actor, and 1 actress were used for both validation and test sets.



		10 MFCCs		20 MFCCs		30MFCCs		40 MFCCs		50 MFCCs		60 MFCCs	
	Dev	Test	Dev	Test	Dev	Test		Dev	Test	Dev	Test	Dev	Test
Speech	Conv2D + Max 4L	**64.2%**	59.2%	62.5%	**64.2%**	47.5%	60.0%	61.7%	60.0%	45.8%	54.2%	45.0%	59.2%
	Conv2D + Max 3L	**56.7%**	53.3%	53.3%	53.3%	49.2%	52.5%	45.0%	**56.7%**	43.3%	53.3%	35.8%	50.0%
	Conv2D + Avg 4L	48.3%	54.2%	65.0%	60.0%	59.2%	**70.0%**	**68.3%**	56.7%	55.0%	53.3%	51.7%	54.2%
	Conv2D + Avg 3L	55.0%	59.2%	**55.8%**	**66.7%**	55.0%	60.0%	51.7%	62.5%	51.7%	59.2%	52.5%	55.0%
Song	Conv2D + Max 4L	54.6%	53.4%	65.9%	62.5%	55.7%	**69.3%**	71.5%	67.1%	69.3%	65.9%	**78.4%**	**69.3%**
	Conv2D + Max 3L	48.9%	51.1%	**79.6%**	56.8%	56.8%	46.6%	70.5%	**76.1%**	72.7%	56.8%	68.2%	63.6%
	Conv2D + Avg 4L	55.7%	56.8%	68.2%	54.6%	45.5%	52.3%	77.3%	71.6%	68.2%	68.2%	**79.5%**	**75.0%**
	Conv2D + Avg 3L	56.8%	50.0%	63.6%	50.0%	50.0%	63.6%	68.2%	**68.2%**	**69.3%**	60.2%	68.2%	62.5%

Figure 1: Validation and test accuracy of every model

	Average Pooling 4L		**Max Pooling 4L**		**Average Pooling 3L**		**Max Pooling 3L**	
	Dev	Test	Dev	Test	Dev	Test	Dev	Test
Speech	**57.92%**	58.07%	54.45%	**59.47%**	**53.62%**	**60.43%**	47.22%	53.18%
Song	65.73%	63.08%	**65.90%**	**64.58%**	62.68%	**59.08%**	**66.12%**	58.50%

Figure 2: Average values of the model accuracies

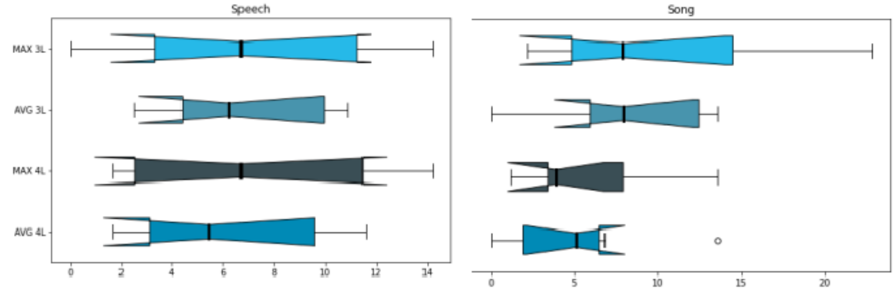


Figure 3: Boxplots of differences between validation and test accuracies

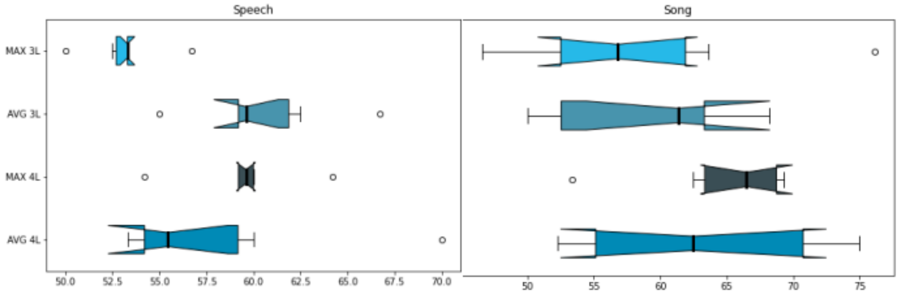


Figure 4: Boxplots of test accuracies

RESULTS AND DISCUSSION

Research Question 1

For every different number of MFCCs 4 different models were tried: 4 Layer 2D Convolutional Neural Networks with average or max pooling and 3 Layer 2D Convolution Neural Networks with average or max pooling. The results are shown in Figure 1. For the speech and song datasets, we had different outcomes. In the case of the speech dataset, the best development set accuracy was 68.3% when 40 MFCCs and 4-layer Convolutional Neural Network with average pooling was used and the best test accuracy was 70% when 30 MFCCs and 4 Layer Convolutional Neural Network with average pooling were used. As it is clear from the table, there was a tendency that the best results for all these 4 models were achieved when the number of coefficients was less than or equal to 40.

For the song dataset, the best test set accuracy was 76.1% when 40 MFCCs and 3-layer Convolutional Neural Network with max pooling were used and the best development set 79.4 when 20 MFCCs and 3-layer Convolutional Neural Network with max pooling were used. It can be noticed that generally the best result for all these models was achieved when the number of MFCCs was more than or equal to 30.

Research Question 2

From Figure 1, for the speech dataset average pooling had better results with both 4 and 3 layers while in the case of the song dataset max pooling achieved higher accuracies. The mean values for all these models were calculated (for different numbers of MFCCs) and they are displayed in Figure 2. Based on mean values, it can be argued that in the case of the speech dataset average pooling had better results and for the song dataset max pooling.

Since the mean values can’t provide insightful information, the differences between development and test accuracies were calculated in order to measure the generalization ability and stability. Boxplots were used to display the results and the results are shown in Figure 3. Additionally, boxplots of test accuracies were created in order to measure the stability (Figure 4). Based on Figure 3, it can be said that average pooling had better generalization ability than max pooling for both datasets. It isn’t that clear to understand whether the average pooling is more stable than the max-pooling as both for almost all models had 2 outliers (out of 6) for the speech dataset (Figure 4). The same thing can be said about the song dataset.

Conclusion and Future Work

In this paper, we compared the results of CNN models when the number of MFCCs varied. We obtained distinctive results for the speech and song datasets. While in the case of speech dataset our results support the idea that using the number of MFCCs lower than equal to 40 might lead to better performance, in the case of the song dataset it can be argued that using the number of MFCCs more than equal to 30 would be more efficient. For future work we think that it would be useful to use more versatile values for the number of MFCCs and explore them in combination with other feature extraction methods. Additionally, it would be interesting to use different model architectures like LSTMs and GRUs.

We found that max pooling had higher mean values and better results in the case of the song dataset and vice versa for the speech dataset. Overall, average pooling had better ability to generalize. Both seemed to be unstable, and it can’t be implied whether one of them has a superiority over the other. Still, it might be assumed that in case of speech dataset average pooling is mightier to give better results while for the song dataset max pooling.

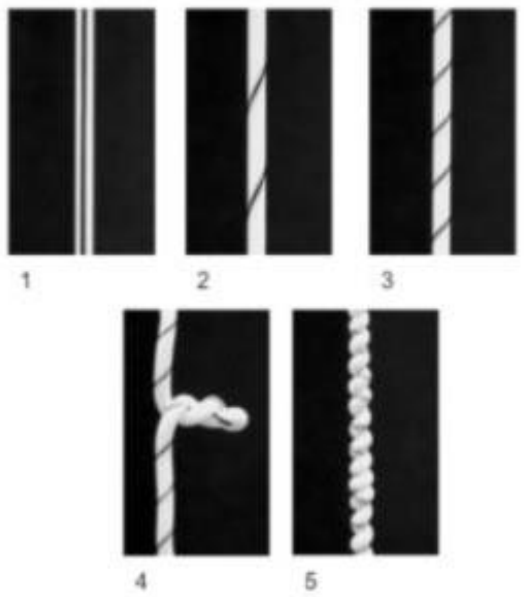
Additionally, we found that even though we didn’t divide samples based on gender, models still managed to learn meaningful features and gave satisfactory results that are comparable and sometimes better than previous attempts.

Since the dataset contained only short audio files, the length ranging between 4-5 seconds, it would be interesting to try to detect emotions in larger files. We consider that in this case, it would be useful to use multimodal architectures in which transcriptions would also be included.

Facial Expressions with Artificial Muscle

Polyfilament Artificial Muscle Fiber

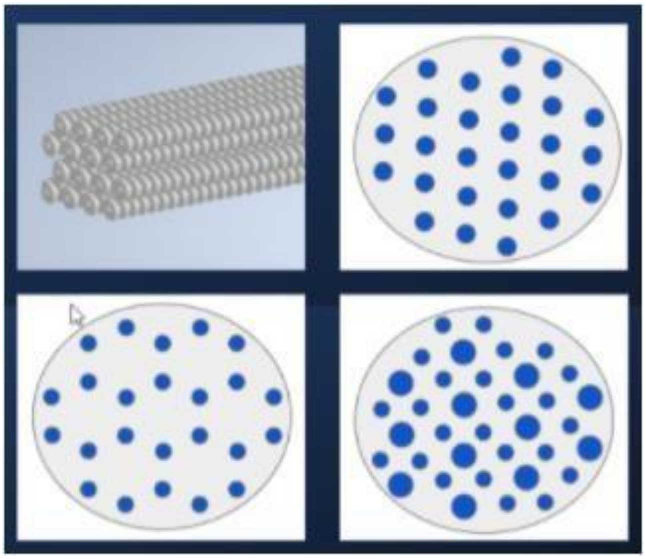
One of the most exciting and innovative sectors of the robotic industry today is the field of artificial muscle fiber. Largely focused on the University and research center space, lightweight artificial muscle fiber offers the promise of matching the characteristics of natural muscles. Historically, artificial muscle fiber came at high cost, limited life cycles, and inefficient energy conservation. In recent years a new class of artificial muscle fiber has been explored based on better selection of materials and design. Monofilament and Polyfilament thermal activated artificial fiber have been shown to be substantially lower cost in manufacturing, deliver 40 times more muscle contraction strength than natural muscle, and significantly longer life cycles. Super-twisted or extreme-twisted designs can contract by 49%, lift loads 100 times heavier than human fiber, and generate 5.3 kilowatts of mechanical work per kilogram of muscle fiber.[27]



Polyfilament thermal activated artificial muscle fiber design begins with a single untwisted synthetic single fiber filament, such as nylon, which has been extruded to form a string of material. The pre-heat-set coil is then twisted with a conductor (1), such as copper, to provide the thermal delivery system of the actuator. The polyfilament material is then super-twisted (2)(3)(4) to form a spring or coil formation. The resulting formation is stretched to the desired design length and then heat-treated to cure and heat set the material. The polyfilament/conductor (5) formation is again stretched to the desired length and heat-set for a second time. The final material form is a polyfilament/conductor coil capable of contraction upon electrical current passing through the conductor.

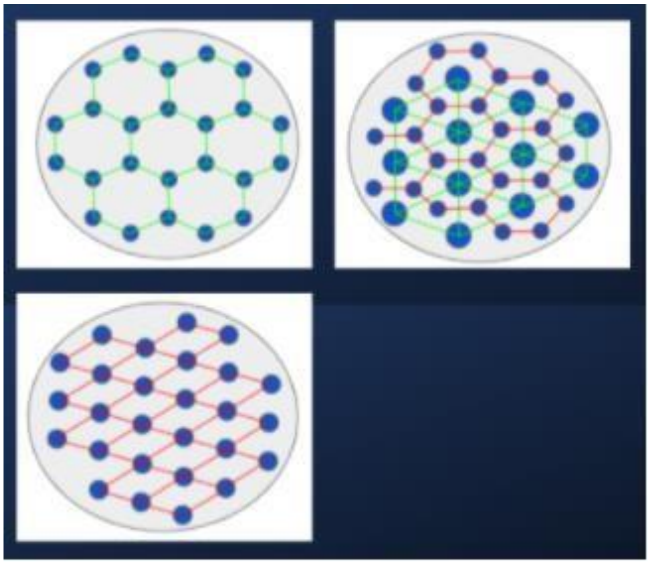
Polyfilament Actuator Matrix

One of the large advantages of polyfilament muscle fiber is that the matrix configuration can be constructed to emulate the form and motion of human muscle fiber. Length, diameter, and specific structure can of cast into specific form. For future study and development, specific polyfilament size (length and diameter) and material type will be developed to mimic specific human muscle type. The polyfilament actuator is activated by running current through the coiled conductor. The amount of current will determine the speed and distance of contraction of the actuator. Forming complex matrices of polyfilament actuators can result in differential speeds and motion which can emulate typical human muscle.



In the addition to the number and physical form of the specific polyfilament fibers, the differing matrices of actuators can also be actuated independently.

In the example below, a smaller hexagonal matrix of polyfilament fibers is wired to form a single circuit. A second formed larger stacked matrix is wired in a second circuit. These two matrices are combined into a single muscle type but wired independently to increase controls of features such as speed and direction.



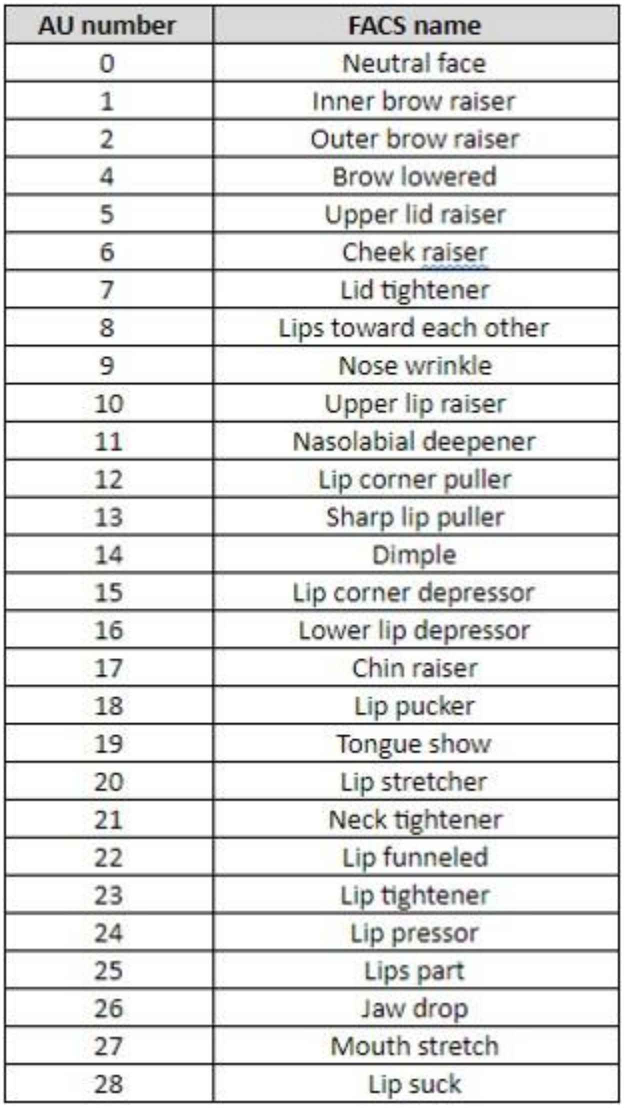
Silicone Porous Resin Substrate

The artificial muscle fibers are configured to the desired matrix and the fibers are then set with a LMS silicone or similar porous substrate. Specific configurations are based on the type of slow-twitch or fast-twitch muscle they are intended to emulate and perform. The specific matrix is set into a die ready for the application of the LMS silicone. The die is filled with a mixture of the unformed LMS silicone and a solvent. The formation of pore structure in the membrane is related to the phase separation and thus the phase separation process of the casting solution.[28] The solvent is evaporated with specific heat and timing to leave the polyfilament and LMS silicone is the desired formation. Additionally, mechanical and thermal properties can be taken advantage with specific weave designs of the polyfilament actuators. Thermal displacement features can conserve and efficiently convert energy through the porous design. More importantly, the LMS porous silicone foam substrate will increase the performance of the artificial muscle returning to the natural state after actuating. The die casts for the formation of each artificial muscle will emulate the size and shape of the related human muscle.

Facial Action Coding System (FACS)

Facial Action Coding System (FACS) is a system to taxonomize human facial movements by their appearance on the face, based on a system originally developed by a Swedish anatomist named Carl-Herman Hjortsjö. Movements of individual facial muscles are encoded by FACS from slightly different instant changes in facial appearance. It is a common standard to systematically categorize the physical expression of emotions.[29][30]

The specific facial expression is characterized by a combination of individual FACS name types such as lowered brow, sharp lip puller, or jaw drop. These combinations form emotional action units to express facial representations of emotions. For example, happiness is expressed by the combination of action units 6 and 12, which are characterized by cheek riser (6) and lip corner puller (12). The Facial Action Coding System does not specifically assign action units to specific muscle groups within the human face however, the Destiny development team will detail and determine specific artificial muscle groups, which mimic actual human motion and form, to actuate during the expression of specific emotions. This development of assigning specific muscle groups will certainly be a significant contribution to the existing FACS framework.



AU number	FACS name
0	Neutral face
1	Inner brow raiser
2	Outer brow raiser
4	Brow lowered
5	Upper lid raiser
6	Cheek raiser
7	Lid tightener
8	Lips toward each other
9	Nose wrinkle
10	Upper lip raiser
11	Nasolabial deepener
12	Lip corner puller
13	Sharp lip puller
14	Dimple
15	Lip corner depressor
16	Lower lip depressor
17	Chin raiser
18	Lip pucker
19	Tongue show
20	Lip stretcher
21	Neck tightener
22	Lip funneled
23	Lip tightener
24	Lip pressor
25	Lips part
26	Jaw drop
27	Mouth stretch
28	Lip suck

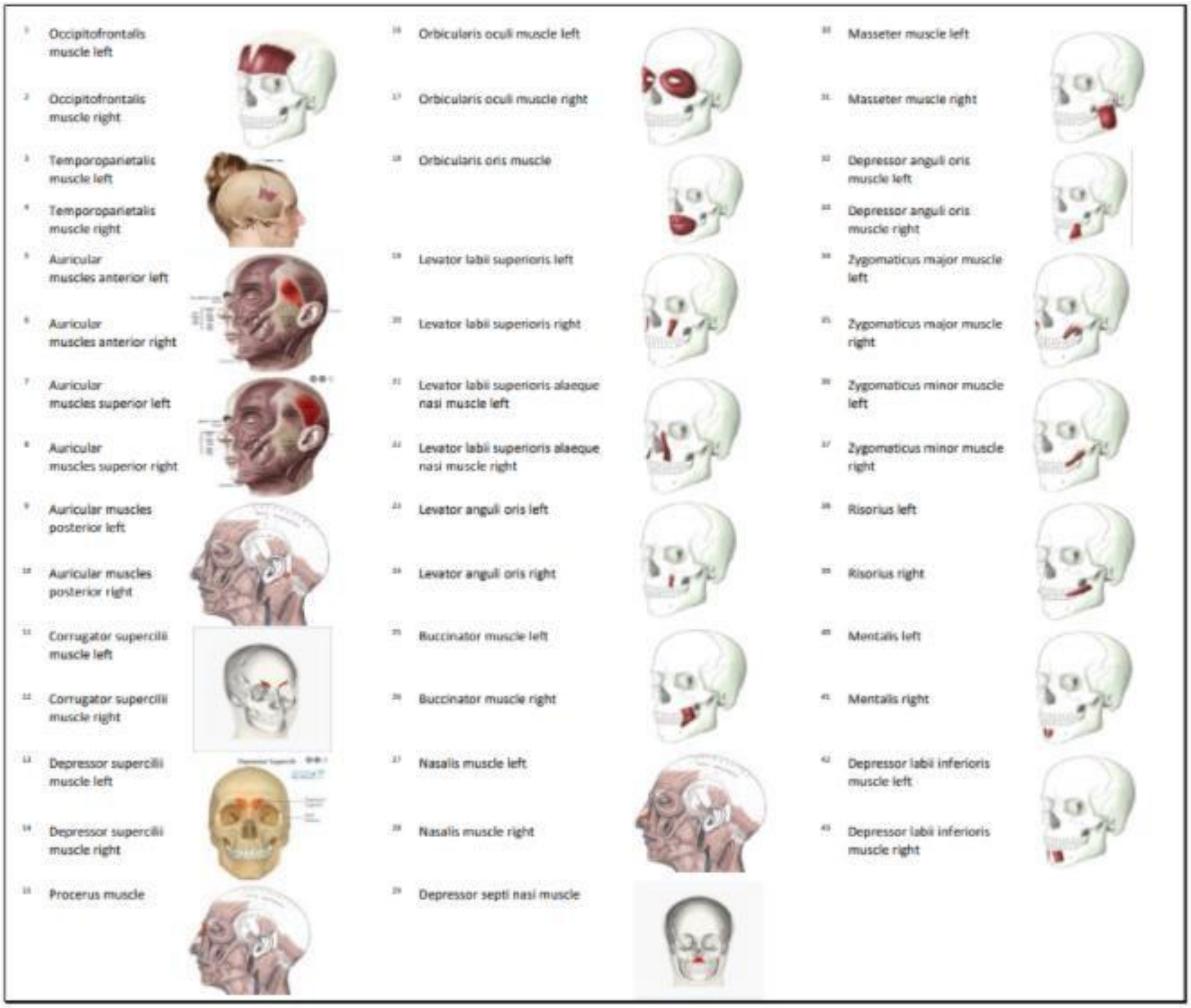
FACS with Destiny Artificial Muscle

The objective of Destiny robot to express accurate human facial expression relies on the Destiny robot to operate muscle groups which mimic human expression. The Facial Action Coding System is the method to characterize muscle groups with human expression of emotions such as happiness and anger. There is not a direct correlation between individual muscles and the action units within the FACS architecture. Therefore, Destiny Robotics will develop a system to correlate the specific action units of the FACS and the Destiny robot muscles. Each polyfilament artificial muscle will be activated by a central controller with applies a voltage across the conductive circuit of each of specific 43 muscles within the Destiny robot face. That voltage can be applied variably between 0V and 5V. A video gathering device will be placed in front of the Destiny robot which will supply the images for a convolutional neural network designed for facial expression recognition. A pre-classified training data set (such as OpenFace) will be used to develop a model for operating the Destiny robot. Initially, a randomized activation of muscles on the Destiny robot will be used to compare the library of classified action units. The CNN will be allowed to train with a framework of reinforcement learning.[31] Using all 43 independent Destiny facial muscles, there are 243 combinations. Each muscle can be activated between 0V and 5V in 0.01V increments. This would result in trial and error 1.4x106472 combinations. Using a reinforcement learning framework with reward feedback, the algorithm provides a method to more quickly aggregate the lessons of determining FACS action units in time.

Silicone Porous Resin Substrate

The Destiny robot will contain 43 individual muscles within the face to express and mimic human facial emotions. Each muscle is constructed in a matrix of polyfilament actuators in a porous silicone resin. Each muscle will also contain a surface mounted deformation sensor.

A surface mounted deformation sensor is made of carbon-black impregnated polymer. This polymer has a resistance of 350 ohms per inch. As the artificial muscle is contracted, the resistive value is reduced. A data acquisition edge device monitors the overall length of the muscle and can determine the distance the muscle contracts. The resulting data value for the contraction of the muscle sensor will be added to the CNN model.



1 Occipitofrontalis muscle left

2 Occipitofrontalis muscle right

3 Temporoparietalis muscle left

4 Temporoparietalis muscle right

5 Auricular muscles anterior left

6 Auricular muscles anterior right

7 Auricular muscles superior left

8 Auricular muscles superior right

9 Auricular muscles posterior left

10 Auricular muscles posterior right

11 Corrugator supercilii muscle left

12 Corrugator supercilii muscle right

13 Depressor supercilii muscle left

14 Depressor supercilii muscle right

15 Procerus muscle

16 Orbicularis oculi muscle left

17 Orbicularis oculi muscle right

18 Orbicularis oris muscle

19 Levator labii superioris left

20 Levator labii superioris right

21 Levator labii superioris alaeque nasi muscle left

22 Levator labii superioris alaeque nasi muscle right

23 Levator anguli oris left

24 Levator anguli oris right

25 Buccinator muscle left

26 Buccinator muscle right

27 Nasalis muscle left

28 Nasalis muscle right

29 Depressor septi nasi muscle

30 Masseter muscle left

31 Masseter muscle right

32 Depressor anguli oris muscle left

33 Depressor anguli oris muscle right

34 Zygomaticus major muscle left

35 Zygomaticus major muscle right

36 Zygomaticus minor muscle left

37 Zygomaticus minor muscle right

38 Risorius left

39 Risorius right

40 Mentalis left

41 Mentalis right

42 Depressor labii inferioris muscle left

43 Depressor labii inferioris muscle right

References

[1] https://www.nia.nih.gov/news/social-isolation-loneliness-older-people-pose-health-risks

[2]http://web.archive.org/web/20201108110122/https://www.wsj.com/articles/is-technology-making-people-less-sociable-1431093491

[3] https://www.usnews.com/news/blogs/data-mine/2014/02/27/is-the-internet-bad-for-society-and-relationships

[4]https://www.ijcrt.org/papers/IJCRTICGT000.pdf

[5]https://ijssst.info/Vol-22/No-1/paper3.pdf

[6] https://www.researchgate.net/publication/305719211_Robotic_Systems_Architectures_and_Programming

[7]https://kubernetes.io/docs/setup/

[8]https://lordfulfillment.com/pdf/44/PC8008_MicroStrainCatalog.pdf

[9]https://www.microstrain.com/software/ros

missing

[11] https://recfaces.com/articles/how-facial-recognition-works

[12] https://www.linkedin.com/pulse/task-4-face-recognition-using-transfer-learning-priyanshi-garg/

[13] http://papasearch.net/DeepMind/DeepMind11.html

[14] Aloufi R., Haddadi H. & Boyle D. (2019) Emotionless: Privacy-Preserving Speech Analysis for Voice Assistants. https://arxiv.org/pdf/1908.03632.pdf

[15] Banga S., Upadhyay U., Agarwal P., Sharma A. & Mukherjee P. (2019) Indian EmoSpeech Command Dataset: A dataset for emotion based speech recognition in the wild. https://arxiv.org/pdf/1910.13801.pdf

[16] Crangle C.E., Wang R., Perreau-Guimaraesa M.,Nguyena M. I., Nguyena D. T. & Suppesa P. (2019). Machine learning for the recognition of emotion in the speech of couples in psychotherapy using the Stanford Suppes Brain Lab Psychotherapy Dataset. https://arxiv.org/pdf/1901.04110.pdf

[17] Dasgupta P. B. (2017) Detection and Analysis of Human Emotions through Voice and Speech Pattern Processing. https://arxiv.org/pdf/1710.10198.pdf

[18] Davletcharova A., Sugathan Sh., Abraham B. & James A. P.(2015) Detection and Analysis of Emotion From Speech Signals. https://arxiv.org/pdf/1506.06832.pdf

[19] EmoSpeech Dataset. https://emo-speech.web.app/

[20] Lalitha S., Geyasruti D., Narayanan R., Shravani M. (2015) Emotion Detection using MFCC and Cepstrum Features. https://www.sciencedirect.com/science/article/pii/S1877050915031841

[21] Livingstone SR, Russo FA (2018) The Ryerson Audio-Visual Database of Emotional Speech and Song (RAVDESS): A dynamic, multimodal set of facial and vocal expressions in North American English. PLoS ONE 13(5): e0196391. American English. PLoS ONE 13(5): e0196391. https://doi.org/10.1371/journal.pone.0196391

[22] Matthias L., Egger M. & Hanke S. (2019) Evaluating Methods for Emotion Recognition based on Facial and Vocal Feature. https://www.researchgate.net/publication/33736

6576_Evaluating_Methods_for_Emotion_Recogn

[23] Ristea N., Dutu L. C. & Radoi A. (2020) Emotion Recognition System from Speech and Visual Information based on Convolutional Neural Networks. http://arxiv.org/pdf/2003.00351.pdf

[24] Suppes, Patrick, Nguyen, Michelle U., Nguyen, Duc T., Nguyen, Cynthia, Tuckner, Margot, and Perreau Guimaraes, Marcos. Suppes Brain Lab Psychotherapy EEG Dataset. https://exhibits.stanford.edu/data/catalog/mz950kf4667

[25] Tripathi S., Kumar A., Ramesh A., Singh C., Yenigalla P. (2019) Deep Learning based Emotion Recognition System Using Speech Features and Transcriptions. https://arxiv.org/pdf/1906.05681.pdf

[26] Venkataramanan K. & Rajamohan H. R. (2019) Emotion Recognition from Speech. https://arxiv.org/pdf/1912.10458.pdf

[27] https://www.researchgate.net/publication/260372180_Artificial_Muscles_from_Fishing_Line_and_Sewing_Thread

[28] https://pubmed.ncbi.nlm.nih.gov/23448280/

[29]https://en.wikipedia.org/wiki/Facial_Action_Coding_System

[30] https://artsandculture.google.com/entity/facial-action-coding-system/m051n_0?hl=en

[31] https://wiki.pathmind.com/deep-reinforcement-learning

[32] Face Net: A Unified Embedding for Face Recognition and Clustering, https://arxiv.org/pdf/1503.03832.pdf

[33] Comparing classical and deep approaches for face recognition in a smartgym application : https://sergioescalera.com/wp-content/uploads/2017/09/comparing-classical-deep.pdf

[34] Y. Sun, X. Wang, and X. Tang. Deeply learned face representations are sparse, selective, and robust. https://arxiv.org/pdf/1412.1265.pdf

[35] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf. Deepface: Closing the gap to human-level performance in face verification. IEEE

[36] EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks https://arxiv.org/abs/1905.11946

[37] ResNet strikes back: An improved training procedure in timm. https://arxiv.org/pdf/2110.00476.pdf

[38] MobileNetV2: Inverted Residuals and Linear Bottlenecks https://arxiv.org/pdf/1801.04381v4.pdf

[39] AN IMAGE IS WORTH 16X16 WORDS: TRANSFORMERS FOR IMAGE RECOGNITION AT SCALE https://arxiv.org/pdf/2010.11929.pdf

[40] Deep Residual Learning for Image Recognition https://arxiv.org/pdf/1512.03385.pdf

Part II

Cryptanalysis of Robotic Systems and Internet of Things (IoT)

1. Introduction

Karen Panetta, dean of graduate engineering at Tufts University, an Institute of Electrical and Electronics Engineers (IEEE) Fellow and a member of the IEEE Robotics and Automation Society said: "However, developers and users should remember to protect cybersecurity in the rush to respond to urgent needs". The robots from different fields deserved a huge attention. She also said: "Safety assurances for robotics and artificial intelligence are essential to their continued adoption beyond the COVID-19 pandemic". “Right now, with AI and cybersecurity, we’re looking at deep deviations and behavior,” she told The Robot Report. “Hackers have brought down drones by bombarding them with more commands than they could handle.” “Right now, with AI and cybersecurity, we’re looking at deep deviations and behavior,” she told The Robot Report. “It’s a big paradigm shift, with actuators and end manipulators as a key focus for safe and secure interactions,” she said. “Designers need to incorporate more AI to make their systems both more efficient and strong and to inspire confidence [1].”

Securing robots is a crucial requirement as the robots can be subjected to malicious attacks by a hacker that can control robots and robotic systems. Robots are simply devices with executing code to be compromised as any other device in an organization [2]. Cybersecurity consultancy IOActive recently analyzed and found that most of the robots are using insecure communications having authentication issues, very weak cryptography, no authorization schemes, and weak default configurations. Most of the Robotic systems are using vulnerable libraries and open source frameworks. The Robot Operating System (ROS) has been shown to be vulnerable and sensitive to injection and eavesdropping attacks. This can result in data loss or physical injuries. In most cases, real world industrial robots run firmware written in C and they don't use a real-time operating system (OS) such as Linux.

There is a very high risk of robots to be hacked meaning that extra security measures needed to protect them, such as implementation of facial recognition of the owner. Another real risk is privacy invasion in the case of a robot that has freedom of roaming inside the house. In order to prevent remote hacks the firmware & robot OS must be kept up-to-date, ensure complex passwords and multi factor authentication methods.

Home robots have limited functionalities like our destiny can pose a security through their interface as we can consider a robot as a part of IoT (the Internet of Things) if they are connected to a home Wi-Fi. Main problem is that customers never change the default passwords or use weak passwords that might be broken by hackers allowing them to gain. Building Robot security at an early stage is the best way of securing robots.

2. Robotics security: vulnerabilities

There are several vulnerabilities in robotic systems:

- **Security vulnerability -** without penetration testing the performance of robotic devices and systems can be affected. Penetration Testing aims to detect as many robot vulnerabilities as possible to mitigate them. Security vulnerability includes using inappropriate security measures, lack of programming (coding) skills [3]. Testing gives us the possibility to easily find the errors and re-modify the code to mitigate errors.
- **Platform vulnerability** implies the lack of regular software updates and patches to maintain a secure robotic system. A security patch update covers the holes found in DoS attacks through internet connection which is happening during each upgrade [153]. On the robotic software and database vulnerabilities.
- **Application vulnerability -** without testing applications for bugs can affect the performance of the robotic system. So, testing is essentially required.
- **Network vulnerability -** without network security testing robotic systems may be vulnerable to wired/wireless network attacks including eavesdropping, man-in-the-middle, DDoS, replay, spoofing, sniffing and other attacks.
- **Management vulnerability -** a lack of well-advised planning, security measures and guidelines, policies and procedures.

3. Robotic attacks classification

3.1. Robot Hardware Attacks

These attacks include least dangerous attacks such as phishing and most dangerous attacks like hardware Trojans. They can lead to back-doors implementation where the attacker can gain unauthorized access to the robots [151-152] or a full access to the robot's hardware. Robots are prone to side channel attacks leading to sensitive data loss or robot's system exploitation.

3.2. Robot Firmware Attacks

The Robot's Operating System (OS) is prone to other hand, the applications are running on software programs for performing the required tasks. Software programs are prone to application attacks such as malware that includes software Trojans, viruses, worms, buffer overflow and malicious code injection attacks (Fig.1) [154].

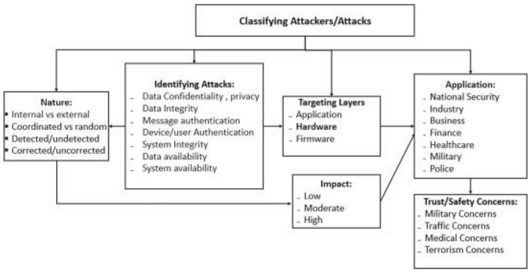


Fig. 1. Proposed attacks classification

4. IoT Security Issues and Capabilities

The definition of the Internet of Things (IoT) by ITU·T - "A global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) things based on existing and evolving interoperable infor-mation and communication technologies" [1]. IoT reference model (Fig.1) con-sists of four layers plus security capabilities and management Capabilities. Four layers of IoT are: 1) application layer, 2) service support and application support layer, 3) network layer and 4) device layer. We focus on IoT Security capabilities such as generic security capabilities and specific security capabilities.

Generic security capabilities include:

a) Authentication, authorization, privacy protection, security audit and anti-virus, Application data confidentiality and integrity protection at the application layer;

b) Authentication, authorization, use data and signalling data confidentiality, and signalling integrity protection at the network layer;

c) Authentication, authorization, access control, device integrity validation, data confidentiality and integrity protection at the device layer;

d) Specific security capabilities are closely coupled with application-specific requirements.

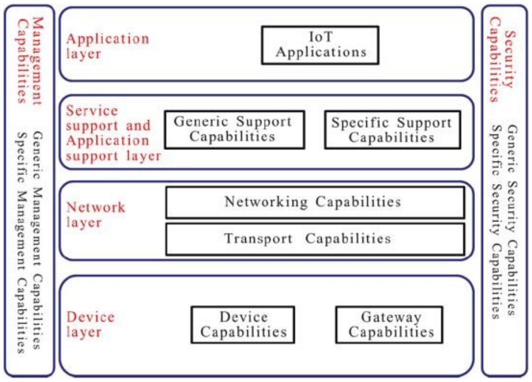

Management Capabilities
Generic Management Capabilities
Specific Management Capabilities
Application layer
IoT Applications
Service support and Application support layer
Generic Support Capabilities
Specific Support Capabilities
Network layer
Networking Capabilities
Transport Capabilities
Device layer
Device Capabilities
Gateway Capabilities
Security Capabilities
Generic Security Capabilities
Specific Security Capabilities

Fig. 1. IoT reference model. (Rec. ITU-T Y.2060 (06/2012))

Securing IoT is the biggest challenge for technology companies. In IoT network, data is collected from external sensors that are placed in public sites allowing anyone to send harmful data to the network. Bring your own device (BYOD) is another case when third-party devices are allowed to access the network [2]. There are most vulnerable areas of IoT given in the table 1:

Table 1. IoT vulnerabilities

IoT Security Requirements	Description
Confidentiality	Ensures that the exchanged messages can be accessed only by the intended users.
Integrity	Ensures that the exchanged messages were not modified by the intruder.
Authentication	Ensures that the sender and the receiver involved in any operation are right identities avoiding a masquerade attack usually targeting this requirement and claiming to be another identity.
Availability	Ensures that the service is not denied avoiding Denial of service attacks targeting this requirement and causing service disruption.
Authorization	Ensures that entities are permitted to do the operation they request to perform.
Freshness	Ensures freshness of the data
Non-repudiation	Ensures that an entity is unable to deny an action that it has performed.
Forward Secrecy	Ensures that after an object leaves the network, it no longer have an access to data exchange process.
Backward Secrecy	Backward Secrecy: ensures that a new object joining the network was unable to access the previous communications.

Wi-Fi technology is one of the leading technology in IoT systems and plays a very significant role but main challenge in using Wi-Fi networks is security issues. Wi-Fi uses radio waves prone to eavesdropping. WPA2 protocol security issues discovered recently by Computer Scientists and WPA3 has been developed to improve Wi-Fi security aspects. WPA2 uses a cryptographic - four-way handshake process for validation of the users. Main weakness in WPA2 was found in 2017 using "key reinstallation attacks" (KRACKs) [2]. In a new WPA3-Personal protocol have been found the vulnerabilities allowing intruders to crack Wi-Fi passwords and intercept encrypted traffic sent between the Wi-Fi users.

5. Cryptographic solutions and protocols

Cryptographic protocols can be used to authenticate devices and users. We can differ symmetric and asymmetric encryption algorithms and hashing functions. Well-designed cryptographic algorithm can result in the reduction of the required latency and resources. An efficient authentication protocol can reduce the required communication overhead. This can be achieved by reducing the volume of the communicated message within the authentication process. Improving the key management techniques can help to reach a better security level.

Symmetric cryptographic protocols are preferred to be more lightweight than asymmetric ciphers with the Advanced Encryption Standard (AES) which is faster than

Elliptic-Curve Cryptography (ECC) in [286]. So, symmetric protocols can be considered as more energy efficient when using the optimized AES block cipher. There are different other lightweight ciphers such as KATAN [287], KLEIN [288], mCrypton [289], Piccolo [290], PRESENT [291], TWINE [292], and EPCBC [293].

6. Cryptographic Solutions In Wi-Fi Security And Cryptosystems

For securing Wi-Fi networks the existing wireless protocols must ensure user authentication, message privacy and integrity. Cryptographic approaches are required in Wi-Fi networks for dealing with attacks providing integrity and confidentiality that effect on the Wi-Fi network performance. We can use cryptosystem as a set of three algorithms: for encryption, decryption and a key generation. Wi-Fi Infrastructure incorporates billions of digital devices and users. Every node in Wi-Fi networks must be provided with cryptographic functions like symmetric and asymmetric cryptographic primitives for performing data encryption and authentication. NTRU is a probabilistic cryptosystem that includes a random element that means each of the messages may has many encryptions. Either encryption or description using NTRU are fast enough. Creation of key is easy and fast. NTRU is a probabilistic cryptosystem that includes a random element that means each of the messages may has many encryptions. Either encryption or description using NTRU are fast enough. Creation of key is easy and fast. In comparison to the other public key cryptosystems at equivalent security levels, NTRU can offer: 1) more efficient encryption and decryption in software and hardware implementations; 2) faster key generation that allows to use the disposable keys [17-21].

7. Neural Cryptography

Even though this algorithm has been around for many generations, increasing usage of Artificial Intelligence gives hope and raises a lot of questions about whether we can use it in cryptography as well. Successful AI that can master how to encrypt and decrypt data in a way that it will be only decryptable by authenticated users - this concept is very promising and likeable for computer scientists, especially when Machine Learning (ML) and Deep Learning (DL) algorithms have become very powerful tools against many tasks that seemed impossible to master for classic algorithms. For example, Microsoft's deep learning algorithms can outperform humans in image recognition tasks [5], which again outlines the power of

AI and how useful it might be if used correctly on cryptosystem tasks.

This was the motivation that made us study the concept of Neural Key exchange. Which means using AI algorithms to create secret keys for authenticated users [6]. To achieve this, the elements of tree parity machines are used. In this paper, we will explain how tree parity machines work and all the adaptations that we came up with, that make this algorithm and structure more robust to hacker attacks, while maintaining desired computational cost and speed.

References

1. https://www.therobotreport.com/cybersecurity-remembered-robotics-ai-developers-says-ieee-expert/
2. https://www.roboticstomorrow.com/article/2018/04/securing-the-robots/11719
3. https://cybersecurityforrobotics.com/
4. Jean-Paul A. Yaacoub, Hassan N. Noura, Ola Salman, Ali Chehab. Robotics cyber security: vulnerabilities, attacks, countermeasures, and recommendations. International Journal of Information Security. https://doi.org/10.1007/s10207-021-00545-8. 2021. © Springer-Verlag GmbH, DE
5. https://aliasrobotics.com/robot-penetration-testing.php
6. L. Mirtskhulava, N. Gulua, N. Meshveliani. Iot Security Analysis Using Neural Key Exchange Protocol. GESJ: Computer Science and Telecommunications 2019|No.2(57)
7. L. Mirtskhulava, L. Globa, N. Meshveliani and N. Gulua, "Cryptanalysis of Internet of Things (IoT) Wireless Technology," *2019 International Conference on Information and Telecommunication Technologies and Radio Electronics (UkrMiCo)*, 2019, pp. 1-4, doi: 10.1109/UkrMiCo47782.2019.9165363.
8. Mirtskhulava, Lela; Gulua, Nana; Meshveliani, Nugzar. NTRU Cryptosystem Analysis For Securing Iot. Computer Science & Telecommunications. 2019, Vol. 56 Issue 1, p59-66. 8p. 3
9. Erekle Shishniashvili, Lizi Mamisashvili, Lela Mirtskhulava. Enhancing Iot Security Using Multi-Layer Feed Forward Neural Network With Tree Parity Machine Elements. UKSim-AMSS 22nd International Conference on Modelling & Simulation, Cambridge University (Emmanuel College), International Journal of Simulation Systems, Science & Technology. V.21 N2. https://ijssst.info/Vol-21/No-2/cover-21-2.htm, 2021.
10. Iavich M., Iashvili G., Gnatyuk S., Tolbatov A., Mirtskhulava L. (2021)

Efficient and Secure Digital Signature Scheme for Post Quantum Epoch. In: Lopata A., Gudonienė D., Butkienė R. (eds) Information and Software Technologies. ICIST 2021. Communications in Computer and Information Science, vol 1486. Springer, Cham. https://doi.org/10.1007/978-3-030-88304