Final Project: Machine Learning for Interactive Systems(CS4242)

Novel Interaction Paradigms

Muadh Muhsin Zibiri[[1]](#footnote-1)

1st year B.Sc Artificial Intelligence & Machine Learning,University of Limerick; [22235302@studentmail.ul.ie](mailto:22235302@studentmail.ul.ie)

1. Introduction

In Science, a paradigm is defined as a distinct set of concepts or thought patterns, including theories, research methods, postulates, and standards for what constitute legitimate contributions to a field. The word paradigm is Greek in origin, meaning "pattern", and is used to illustrate similar occurrences. Interaction paradigms are the different ways in which people can interact with computers and other digital devices. In the past, most interaction paradigms involved using a mouse, keyboard, and monitor to input and receive information. However, with advances in technology, there are now many new and exciting ways to interact with computers.

Novel interaction paradigms are all about finding new and innovative ways to interact with computers and other digital devices. These new paradigms can make technology more accessible and intuitive, allowing users to interact with digital devices in ways that feel more natural and seamless. My application of novel interaction paradigm was to use the FaceOSC software together with Max/MSP and Wekinator to generate a series of entertaining and interactive experiences.

FaceOSC is an open-source software toolkit that allows for real-time tracking of facial features from live video input, such as a webcam. It uses machine learning algorithms to analyze facial expressions and movements, and can detect a range of facial features including the position and orientation of the face, the position and rotation of the eyes, and the position of the mouth among others which add up to a total ot 45 focus points on the face. This data is then sent to Max/MSP, a visual programming language used for creating interactive music and multimedia applications. In Max/MSP, I used this data to control various parameters in real-time. Combining FaceOSC and Max/MSP with Wekinator, an open-source machine learning tool, made this project even better. Wekinator allows me to train a machine learning model to recognize specific facial expressions or movements, and then uses this model to control various parameters of a multimedia performance. Overall, the combination of FaceOSC, Max/MSP, and Wekinator provided a powerful toolset for creating interactive multimedia experiences that respond to human facial expressions and movements in real-time.

1. Research

FaceOSC was the key component for my project and as such I had to research into the ethics of facial recognition and its future implications. Facial recognition technology has become increasingly ubiquitous in recent years, with many companies and governments using it for a wide range of applications, from security and surveillance to marketing and personalization. While facial recognition technology has many potential benefits, it also raises important ethical concerns that must be addressed.

One key concern being privacy. Facial recognition technology can be used to track people's movements and activities without their knowledge or consent, potentially violating their autonomy and privacy. There are also concerns about data security and the potential for facial recognition databases to be hacked or misused. Another key issue is bias and discrimination. Facial recognition technology has been shown to have higher error rates for women and people with darker skin tones, which can lead to discriminatory outcomes. I experienced this when working on this project with my partner as some of the training data kept giving wrong outputs when the input was my face yet there was no such issues with my partner’s face(he had a way lighter skintone than me). There are also concerns about the potential for facial recognition technology to be used for discriminatory purposes, such as targeting people based on their race or ethnicity.

Additionally, facial recognition technology raises questions about consent. Individuals may not be aware that their image is being used for facial recognition, or they may not understand the potential implications of their data being collected and analyzed. There are also concerns about the lack of transparency and accountability in facial recognition technology. Many facial recognition algorithms are proprietary and not subject to independent scrutiny, making it difficult to evaluate their accuracy, bias, and potential for harm.

Overall, the ethical implications of facial recognition technology are complex and multifaceted. More research is needed to better understand the potential risks and benefits of facial recognition, and to develop ethical guidelines and best practices for its use. It is important that developers, regulators, and users of facial recognition technology consider these issues and work to ensure that it is used in ways that are ethical, transparent, and respectful of individual rights and freedoms.

1. development process

The main system was a combination of FaceOSC, wekinator and Max/MSP. The brain of the project was a max patcher called FaceGames which would contain other subpatches with various games. It emphasized on 15 of the 45 focus points. The data point were received on port 8338 by default. There were 6 sub patches within FaceGames, viz; Stoneface, ColourMatch, TheMatrix, Gestures, StareContest and sound.

The main algorithm used for this project was Support Vector Machine. It is a popular and powerful supervised learning algorithm used for classification and regression analysis. It is based on the concept of finding a hyperplane that separates different classes of data in a high-dimensional space. The main idea is to find the best hyperplane that maximally separates the data points from different classes. It works by mapping the input data into a higher-dimensional space, where it becomes easier to separate different classes using a hyperplane. The hyperplane is chosen to maximize the margin, which is the distance between the hyperplane and the closest data points of different classes. The points closest to the hyperplane are known as support vectors, and they determine the position and orientation of the hyperplane. This algorithm was chosen because of its effectiveness in handling high-dimensional nonlinear and complex datasets. It can be used for binary and multi-class classification and is less prone to overfitting than other algorithms. Overall, SVM is a powerful algorithm that can provide accurate and near-perfect classification results when used properly.

Stoneface uses the mouth-width and mouth-height to detect the time a person takes to make some sort of facial expression similar to a grin, smirk, smile or laugh. The aim of the game is to see which of the player(s) has the most time before making a laugh-like expression as other player(s) try to make them laugh. The training data had two classes, one was a stone-faced(no expression) face while the other was several laughlike expressions.

ColourMatch is a slightly complex game where players uses 3 inputs to try to match a tertiary colour from a [[2]](#endnote-1)set of curated colours on a RGB swatch. The sound input controls the amount of blue while the green value is controlled by the mouth height. The red input is controlled by a dragged-mouse processor sketch trained in wekinator. This game can be played individually, where one person controls all 3 inputs or as group where each player controls at least 1 input.

TheMatrix, similar to its name, is based on a 3x3 mirrored grid of the FaceOSC screen. A timer is set for 60 second and players try to position their face in the position of a randomly generated number(from 1-9) on the screen. The players scores a point on each successful match and the game can be played between multiple players to see who gets the highest score. This game was trained in wekinator based on the somehow accurate position of the face in the 3x3 mirrired grid, therefore it had nine classes.

Gestures is a game which unlike the others uses Dynamic Time Warping. The main player has to make a specific facial gesture(one of 6 possible) that matches the number the other player draws on the DTW processing sketch. It uses 11 datapoint which were selected based on their worthiness of capture in the process of making a gesture.The wekinator project was trained to recognize, to an extent, the 6 different numbers made on the processing sketch which the player has to match with a facial gesture. 1 = neutral (essentially stone face) 2 = surprised (open mouth and raised eye brows) 3 = sleepy (head tilited with eyes closed) 4 = happy (smile - cresent up) 5 = sad (opposite of smile - cresent down) 6 = cringe (harder to describe but looks like a sour face with a sense of dissaproval)

StareContest is similar to the Stoneface game but is different in that it uses 4 data point (the right-eye, left-eye, left-brow and right-brow) to detect the openness of the eye and lack thereof. It is very similar to a staring contest between 2 people but players have to try to see who can last longer before blinking.

Sounds is mainly there as an aesthetic. It makes 2 sounds based on whether a face is detected or not. The first sound, made when a face isn’t detected, is similar to that made on game shows(specifically Family Feud) when a participant gives an incorrect answer. The second sound, made when a face is detected, is similar to that from the ICONIC Super Mario game when a player wins an item that is deemed good.

1. conclusion & future work

This project led me to explore a new algorithm(SVM) which hadn’t been taught previously. I also explored the slight advantages of the other taught algoritms in relation to the superior SVM for better understanding. There were issues regarding the detection of a face of people of different facial structures or skin tones and such isuues were correct by training examples simultaneous and/or individually.

In conclusion, FaceOSC is a powerful tool for creating novel and interactive projects that leverage the facial expressions and movements of the users. It enables a wide range of research and applications, including facial expression recognition, human-computer interaction, biometrics, and augmented reality. The ability to track and analyze facial features in real-time opens up new possibilities for creating more intuitive and natural interactions between humans and computers. However, it is important to consider the ethical implications of using facial recognition technology and to ensure that these projects respect privacy and human dignity. Overall, FaceOSC provides a rich platform for exploring novel interaction paradigms that have the potential to transform how we interact with technology and each other.

In the future I hope to develop a similar software to FaceOSC and or work on FaceOSC and enhance its capabilities and potential and eventually make an application that uses facial recognition alone or in conjunction with other interaction paradigms to develop a gaming/entertainment software.

1. references

Ethics of Facial Recognition: Key Issues and Solutions by Katam Raju Gangarapu (22-01-22) :

<https://learn.g2.com/ethics-of-facial-recognition>

Wekinator:

Rebecca Fiebrink. (2011). Real-time human interaction with supervised learning algorithms for music composition and performance. Ph.D. thesis, Princeton University. Retrieved from <https://rachelbree.com/research/Fiebrink_Dissertation.pdf>

Adam Stark and Rebecca Fiebrink. (2010). Using machine learning to enable new interfaces for musical expression. Proceedings of the 2010 Conference on New Interfaces for Musical Expression (NIME), Sydney, Australia, pp. 164–167. Retrieved from <https://www.nime.org/proceedings/2010/nime2010_164.pdf>

Wekinator project website: <http://www.wekinator.org/>

FaceOSC:

Kyle McDonald and Arturo Castro. (2014). FaceOSC: A tool for prototyping face-based interaction. Proceedings of the 2014 ACM Conference on Designing Interactive Systems (DIS), Vancouver, Canada, pp. 367–370. Retrieved from <https://dl.acm.org/doi/10.1145/2598510.2598584>

FaceOSC project website: <https://github.com/kylemcdonald/FaceOSC>

MaxMSP:

Cycling '74. (2021). MaxMSP. Retrieved from <https://cycling74.com/products/max>

V.J. Manzo. (2017). Max/MSP/Jitter for music: A practical guide to developing interactive music systems for education and more. Oxford University Press.

1. Project conducted in partnership with Eryk Marcinkowski: 1st year B.Sc Artificial Intelligence & Machine Learning,University of Limerick; [22374248@studentmail.ul.ie](mailto:22374248@studentmail.ul.ie) [↑](#footnote-ref-1)
2. [↑](#endnote-ref-1)