# Sign Language Teacher/Interpreter using Xbox Kinect

David Clairmont,

Johnny Doan, Jack Gaither,

Nick Hester, Sam Witucki

# Abstract

- American Sign Language
  - Difficult to learn
  - Many different Platforms but lack ability to keep students engaged

- Objective
  - Develop Sign Language Teacher to improve learning process
  - Make learning ASL easier and more fun

- Approach
  - Train AI using Xbox Kinect by using references
  - References will sign letters and phrases

# Background – Key Concepts

- Digital image processing
  - Computer analyzes an image using algorithms
    - Compression
    - Enhancement
    - Restoration
  - Needed to pick out human body structure from image

# Background – Key Concepts

- Computer vision
  - Computers interpreting images as humans do
    - High level understanding of image
    - Algorithms capable of this are complicated
      - Require AI

# Background – Key Concepts

- AI action recognition
  - AI correctly recognizing an action taking place in an image
    - Subset of computer vision
    - Usually trying to recognize human actions
    - Often requires deep learning and large amount of input data to train the AI

# Background – Key Concepts

- Sign language
    - Will be using American Sign Language
        - Commonly used in English speaking countries
        - Own language separate from English
            - Different grammar rules
        - Unique signs for letters, numbers, words
            - Use hands, face, other body language

# Related Works

- Teaching Introductory Programming Concepts through a Gesture-Based Interface
  - Dr. Lora Streeter
  - Used similar technology to achieve a different goal
- GitHub projects
  - University of North Texas's 2019 Hackathon
    - Created a sign language interpreter with Python
  - Kinect Sign Language project
    - Essentially a working version of what we want to achieve
  - PyKinect2
    - Library for Python that allows developers to easily create applications for Kinect

# Deliverables

- DESIGN DOCUMENT: CONTAINS A LIST OF EACH MAJOR HARDWARE AND SOFTWARE COMPONENT.
- DATABASE SCHEME AND INITIAL DATA
- WEB SITE CODE
- AI, GATHERING/STORING OPTICAL DATA, DIGITAL IMAGE PROCESSING, ETC. CODE (PYTHON)
- FINAL REPORT

# Design – Requirements

- Connect to Kinect API through Python
- Convert Kinect sensory data into trainable datasets
- Allow for extensive training to improve accuracy
- Predefine different sign language letters, words, and phrases into identifiable outputs
- Allow for the user to sign to the Kinect and receive back the identified sign

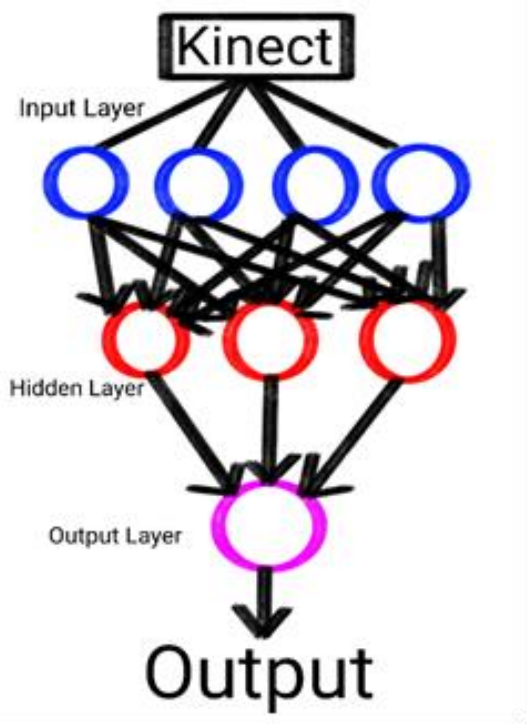
# Design – Use Cases, Design Goals

- Use Cases
  - User runs the application, allowing access to the Kinect camera
  - User can provide a sign and receive the correct answer

- Design Goals
  - Fast, efficient code
  - Easy access for modifying data sets and loading trained models

# Design – High-Level Architecture

- Create Python file to connect to the Kinect API
  - Allows us to access body tracking and coordinate mapping information
- Create functions to convert coordinate data into trainable datasets and usable inputs for neural network
- Convert ASL letters A – Z into outputs for the network
- Create the neural network model
  - Use PyTorch/TensorFlow to create model and training framework
  - Start training the network
- Add evaluation framework to test models for accuracy
- Optimize the network for the highest possible accuracy



# Design – Risks

| Risk | Risk Reduction |
|---|---|
| Inaccurate Sign Language | Extensive research of ASL |
| Inaccurate Neural Network | Train with many datasets in order to improve accuracy. Modify hyper-parameters and learning rate to find the most accurate model |
| Slow code | Write out algorithms and optimize for the best complexity time |

# Design – Tasks: Preparation Phase

- Equipment/Software Preparation
  - Understand how to work with the Kinect
  - Will determine what software(s) are necessary

- Make necessary installations

- If problems, we either get a replacement and/or change software(s)

# Design – Tasks: Design Phase

- Design an application allowing the user to use the Kinect
  - Work with the Kinect API

- Develop library/database storing programmed ASL words/letters

# Design – Tasks: Implementation Phase

- Implement first 3 letters together
    - Serves as the foundation of our implementation for remaining letters/phrases
    - Allows each person to know how the program works
- Complete remaining letters
    - Will be distributed among members
- Work on other ASL phrases
    - Numbers, time, people, etc.

| Team Member | Set of Alphabet Letters |
|---|---|
| David Clairmont | D – H |
| Johnny Doan | I – M |
| Jack Gaither | N – R |
| Nick Hester | S – V |
| Sam Witucki | W – Z |

# Design – Tasks: Testing Phase

- Run the application
  - Check if application fulfills our expectations

- Test the accuracy of the Kinect for each letter
  - Different angles of the person’s hand with respect to the Kinect
  - Different distances from the person to the Kinect

# Design – Tasks: Documentation Phase

- Report results of application
  - Screenshots
  - Walkthrough of the application

- Record demonstration of the Kinect

# Design – Task Schedule

| Tasks | Dates |
|---|---|
| 1. Equipment/Software Preparation. | 11/16-11/30 |
| 2. Test the Equipment. If there are issues, report and get necessary replacements. | 11/30-12/12 |
| 3. Design a Python application for the Kinect. | 1/11-1/25 |
| 4. Develop the library/database that holds all the ASL words/letters that have been programmed. | 1/25-2/1 |
| 5. Implement the first 3 letters of the alphabet and test the accuracy. Add the letters to the library. | 2/1-2/8 |
| 6. Test the application, ensuring that the library is properly connected and running properly. Make necessary adjustments if needed. | 2/8-2/15 |
| 7. Each person implements their set of alphabet letters. Add letters to the library and test if the letters work properly. | 2/15-3/1 |
| 8. Make necessary fixes if the letters aren't working correctly. If there are minimal to no fixes, then we take this time to brainstorm other terms to include. | 3/1-3/8 |
| 9. Implement ASL numbers. Add them to the library and test if the letters work properly. Make necessary adjustments if needed. | 3/8-3/22 |
| 10. Implement ASL terms relating to time. Add them to the library and test if the letters work properly. Make necessary adjustments if needed. | 3/22-3/29 |
| 11. Implement ASL terms relating to people. Add them to the library and test if the letters work properly. Make necessary adjustments if needed. | 3/29-4/5 |
| 12. Record and report the necessary information for the final document. | 4/5-4/19 |
| 13. Make any final adjustments before submitting the project. | 4/19-4/30 |

# Problem – Why We're Doing This?

- Difficulty
  - Difficult to learn, more difficult to learn it correctly
  - Practicing can be difficult

- Relevance in U.S. alone
  - 250,000 – 500,000 rely on ASL because of their disability
  - People who rely on ASL to communicate with disabled people