Dwight Look College of
ENGINEERING
TEXAS A&M UNIVERSITY

Team 72: Hand Gesture Recognition
Bi-Weekly Update 4

Samuel Oncken, Steven Claypool
Sponsor: Prof. Stavros Kalafatis
TA: Pranav Dhulipala

Project Summary

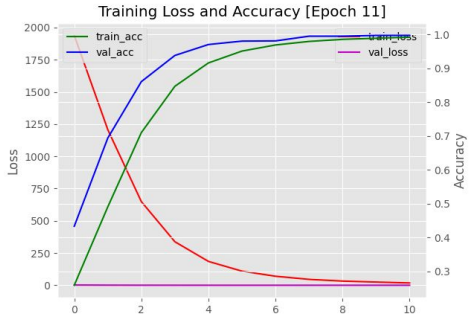
Problem Statement

- Training a hand gesture recognition neural network requires **large amounts** of data
- Collecting this data is **time consuming** and **resource intensive**
- **Numerous participants** must be hired to perform gestures and **expensive** camera equipment is required to record gesture data

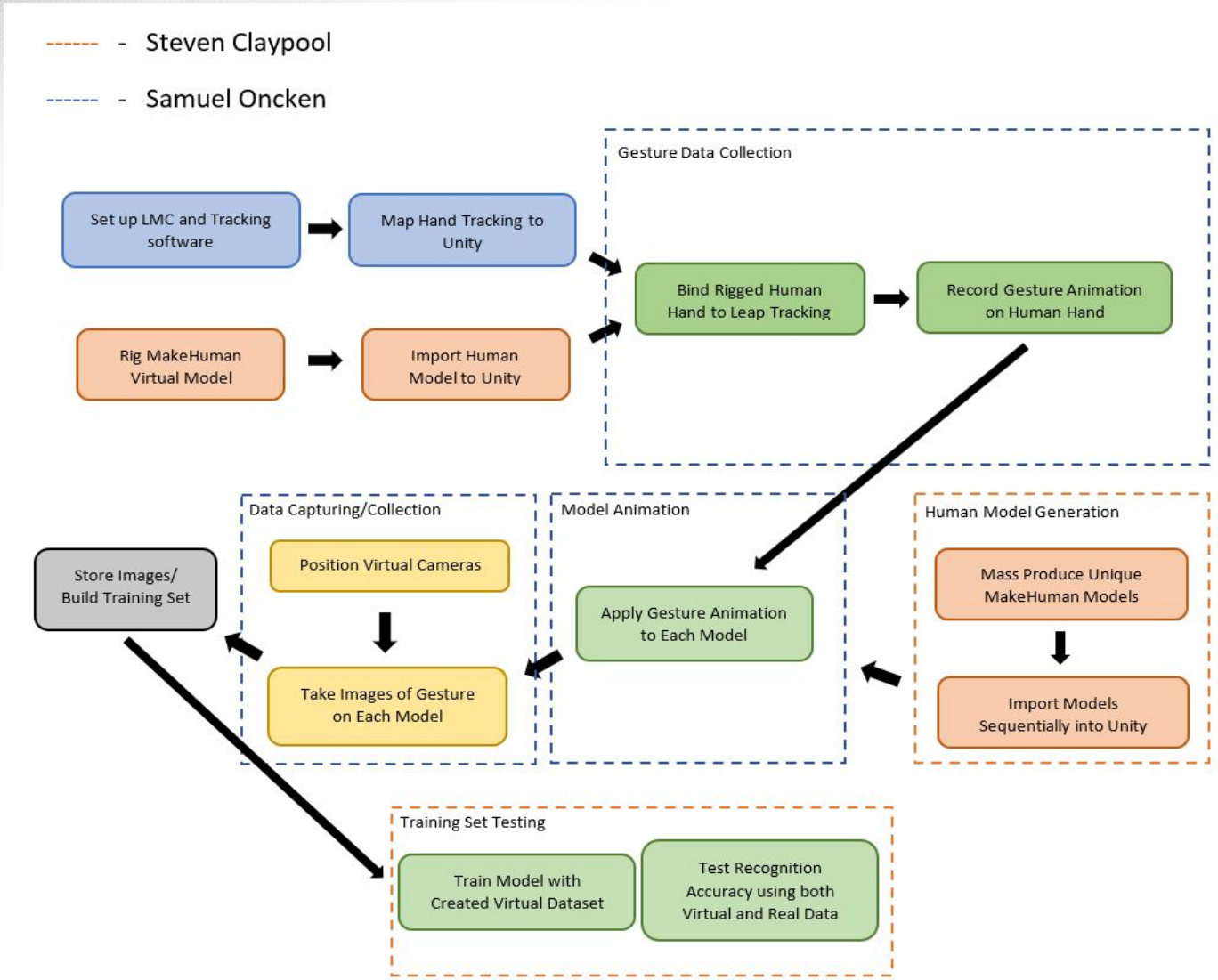
Our Objective

- Achieve similar if not improved gesture detection and object localization accuracy using **virtual (synthetic) data** in comparison to benchmark real datasets
- Provide a **proof of concept** to extend virtual data usage to our graduate students robotic arm training project

ASL - Letter I



Project/Subsystem Overview



Project Timeline

Create Virtual Training Set Generation Unity Environment (Completed 11/20)	Generate Alphabet and Numbers Datasets – 12,000 Images/Gesture (Completed 12/12)	Complete Gesture Classification Analysis using Generated Datasets (To be complete by 03/31)	Complete Object Localization/ Detection Analysis using HANDS Generated Dataset (To be complete by 03/31)	Complete Robotic Application Testing/Training Virtual Environment (To be complete by 04/16)	Demo and Final Report (To be complete by 04/28)

ML Analysis Progress

Pranav Dhulipala, Steven Claypool

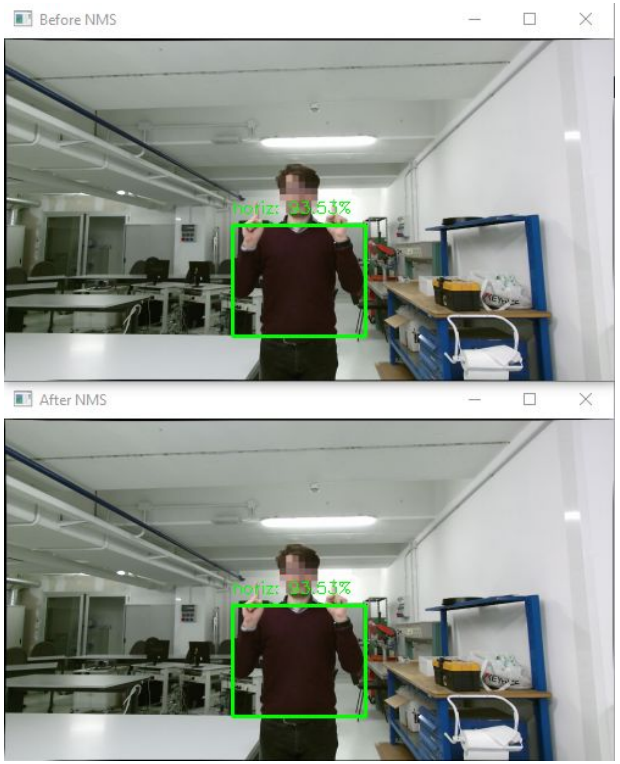
Accomplishments since last update 20 hours of effort	Ongoing progress/problems and plans until the next presentation
Identified issues in both the CNN and region proposals - Model is overconfident on irrelevant regions - Region proposer creates too many irrelevant regions	Continue troubleshooting and testing different architectures/training methods - Region proposal network - Keypoint Detection(?)

ML Analysis

Pranav Dhulipala, Steven Claypool

VGG16 model original result

Inception model result



Dwight Look College of ENGINEERING
TEXAS A&M UNIVERSITY

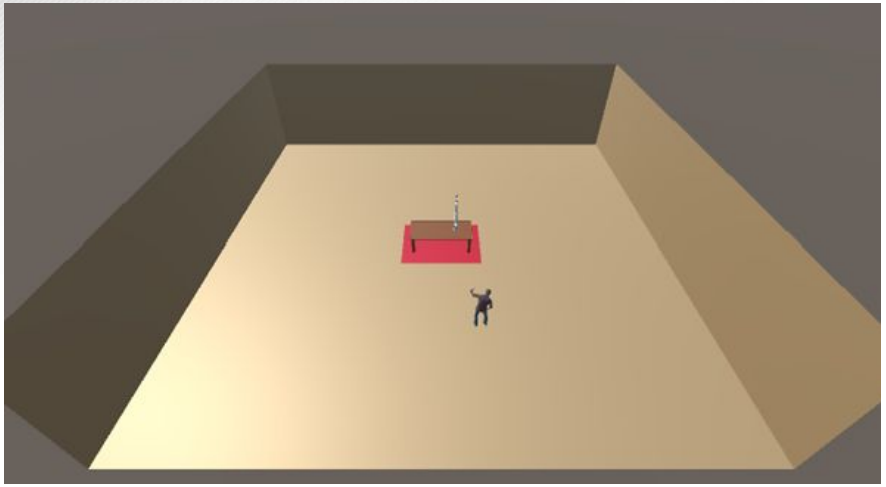
Robotic Application Extension Progress

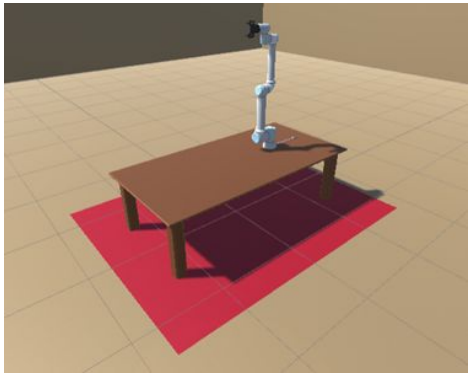
Samuel Oncken

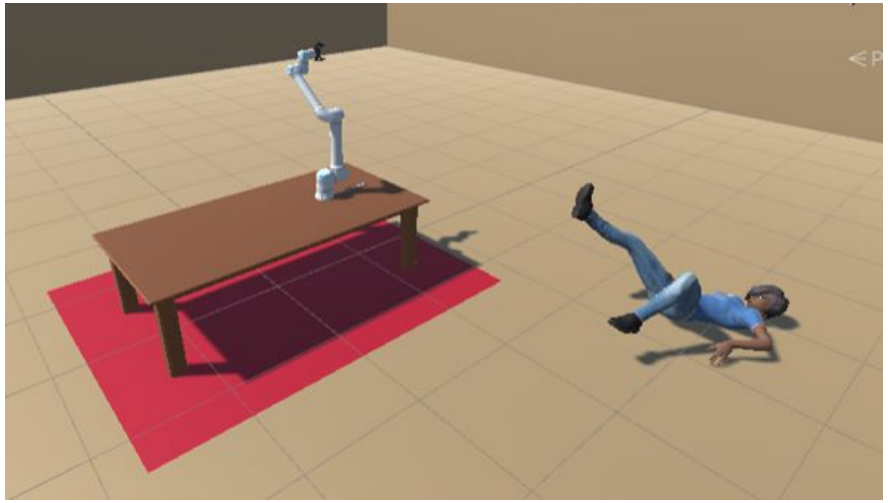
Accomplishments since last update 14 hours of effort	Ongoing progress/problems and plans until the next presentation
Edge case detection/User enhancement - Addition of a “Test Mode” for user to spawn models and create behavior chains manually. - Error throwing upon incorrect user inputs All work uploaded to Github with detailed instructions for use. Integration into Pranav’s Virtual Robotic Arm environment is complete - All test scenario functionality behaves as expected within Robotic Arm environment - Robotic Arm movement functioning as expected in merged environment.	- Continue to implement new functionality as desired by Pranav. We are not involved in the training process - miscommunication Instead, goals for the coming months include: - Optimizing testing environment - Enhancing user experience - Validating functionality - Potentially working to resolve virtual robotic arm gripper movement issues

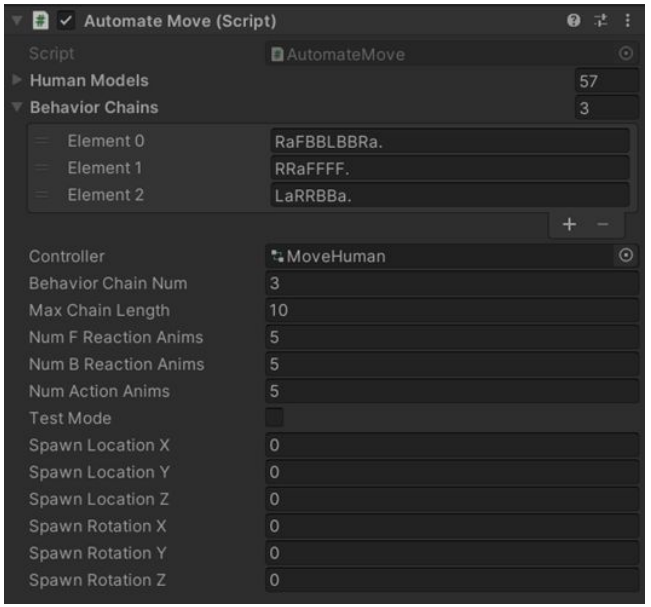
Robotic Application Extension

Samuel Oncken









Execution Plan

Legend

- Analysis Extension
- Robotics Integration
- Final work

Darkened = Completed

Dwight Look College of ENGINEERING
TEXAS A&M UNIVERSITY

403 Validation Plan

Test Name	Success Criteria	Methodology	Status	Responsible Engineer(s)
Benchmark Dataset Training	Gesture recognition neural network can run on our home computer and train using the real dataset. Results quantified	Download the benchmark data set and the code for the CNN. Run the code and confirm similar accuracy to benchmark logs provided.	TESTED - Pass	Steven Claypool
Virtual Dataset Training	Gesture recognition neural network can train using our built dataset and provide accuracy results	Take a final virtual dataset modeled after a real benchmark dataset and use it to train the same CNN as the benchmark. Ensure similar accuracy results.	TESTED - Pass	Steven Claypool
Gesture Recognition Accuracy	Accuracy of gesture recognition is within 5% of benchmark accuracy using our virtual dataset	Train gesture recognition neural network using real and synthetic sets and compare accuracy	TESTED - Pass	Steven Claypool
Synthetic Data on Real Data Accuracy	Test real data with CNN trained on synthetic data and achieve accuracy similar to benchmark.	Train a CNN using different methods/compositions with synthetic and/or real data, and test using real data.	IN PROGRESS	Steven Claypool, Pranav Dhulipala
Unity Hand Mapping	Real hand movement is mapped in Unity	Set up Unity, install Ultraleap plug-ins, map hand motion.	TESTED - Pass	Samuel Oncken
Import Rigged MakeHuman Model	A fully rigged MakeHuman model is imported into Unity	Import model into Unity and confirm appearance and functionality.	TESTED - Pass	Steven Claypool
Virtual Model Unity Hand Mapping	Map hand motion onto an imported MakeHuman model.	Use Hand Binder component/configure settings. Confirm natural motion.	TESTED - Pass	Samuel Oncken
Mounting Stability	Head mounted LMC remains in place during head motion	Mount LMC and plug the device into the computer. Rotate head in all directions and shake head left to right.	TESTED - Pass	Samuel Oncken
Apply Example Animation to Model	MakeHuman model is able to perform an imported full body gesture accurately.	Import an animation .fbx and apply the animation to the rigged human model. Confirm that motion is as expected.	TESTED - Pass	Samuel Oncken
Apply Recorded Gesture Animation to Model	Rigged MakeHuman model can perform a recorded gesture animation.	After recording an animation, apply it to an imported MakeHuman model using the Animator component.	TESTED - Pass	Samuel Oncken
Create and Import Rigged MakeHuman Models to Unity	Minimum 30 MakeHuman models can be generated and imported into Unity	Use MakeHuman “mass produce” function to generate unique character models, each fit with a “Default” rig, with 20% edge cases	TESTED - Pass	Steven Claypool
Data Capture Output and File Type	Virtual camera outputs image data as a .jpg files	Record images of gesture, validate that the data is stored, organized, and is of the desired file type.	TESTED - Pass	Samuel Oncken
Final System Validation	With the press of a button, a large, diverse virtual training set is produced	Run system and validate in output files that each gesture has at least 500 images of gesture performance on differing human models from numerous angles	TESTED - Pass	Samuel Oncken

404 Validation Plan

RCNN Validation	Create a functional RCNN that takes images as an input and outputs the image with correct bounding boxes	Using an image classifier with ImageNet weights, feed sample images into the RCNN and verify correct output of images with bounding boxes	IN PROGRESS	Steven Claypool
RCNN Object Detection on Real Data	Test the RCNN with the HANDS dataset and reach acceptable accuracy metrics	Feed dataset into RCNN and measure accuracy using Intersection of Union (IoU) to compare the predicted bounding box and the ground-truth bounding box	IN PROGRESS	Steven Claypool
RCNN Object Detection on Synthetic Data	Test the RCNN with a synthetic HANDS dataset and reach similar accuracy compared to the real dataset	Feed dataset into RCNN and measure accuracy using Intersection of Union (IoU) to compare the predicted bounding box and the ground-truth bounding box	IN PROGRESS	Steven Claypool
Import and Use Mixamo Animations	Animation clips from Mixamo FBX imported files can be applied to MakeHuman virtual models,	Download FBX (for Unity) files from mixamo.com for animations we wish to use. Import them into Unity project folder. Make each prefab “Humanoid” and extract and test animation clip on MakeHuman models	TESTED - Pass	Samuel Oncken
Randomly Select “Behavior” Animation	After model is spawned, depending on user input for desired “behavior”, a randomly selected animation clip will play.	Use Unity C# to script the categorization of “behavior” specific animation clips and random selection. Use Animator on human model with an animation controller selected that includes all desired animation clips.	TESTED - Pass	Samuel Oncken
Ability to Chain “Behavior” Animations	While environment is running, user is able to manually create test behavior chains or system is able to automatically randomly generate behavior chains	Use Unity C# to script the choice of a “Test Mode” vs. automatic mode. In Test mode, user can manually create behavior chains which link animations clips one after the other in order established by the user. Automatic mode uses a method in Unity C# to randomly generate chains of behavior animations.	TESTED - Pass	Samuel Oncken
Virtual Robot Unity Environment Integration	Be able to merge the testing environment with the robotic arm simulation environment created by Pranav. Make sure all functionality is still available.	Transfer all necessary code, model prefabs, unity scenes, etc.through GitHub Desktop and integrate all of these into the environment already created by Pranav with included robotic arm model.	TESTED - Pass	Samuel Oncken
Human Model Interaction with Robotic Arm Model	If criteria is met such as human models becoming too close to robot arm, they perform a randomized reaction animation	Unity C# scripting for reaction animations to take place using a trigger collider around the robotic arm prefab. Once a model comes into contact with the trigger, all other behaviors in the behavior chain cease to run and a reaction animation is immediately played.	TESTED - Pass	Samuel Oncken

Thank you. Questions?