LIGHTWEIGHT SKELETON-BASED PERSON RE-IDENTIFICATION FOR REAL-TIME SECURITY APPLICATIONS ON RASPBERRY PI

ECE 482 Final Report

Prepared by:

Aubrey McKinney (CpE)

Antonio Mendoza (CpE)

Noor Humphreys (EE)

Seth Myers (CpE)

Faculty Advisor:

Dr. Khan Iftekharuddin

Old Dominion University

Department of Electrical and Computer Engineering

Spring 2026

ABSTRACT

The objective of our project is to create a lightweight skeleton-based person re-identification system that runs on edge computing devices like Raspberry Pi. Many existing camera-based person identification frameworks rely on unreliable characteristics like clothing and accessories. A skeleton-based identification method allows us to provide more robust tracking by ignoring unnecessary information and relying only on body dimensions and poses. For our project, we investigated two separate methods for obtaining skeletal data. The first method uses an RGB-D camera to infer joint locations in 3D space from a depth map and RGB video data. The second method utilizes a long-wave thermal imaging camera and machine learning to extrapolate 3D joints from a 2D thermal map. Both methods employ a convolutional neural network to process the obtained skeleton data and provide an identification result. This project covers a wide range of topics including image processing, machine learning, hardware integration, and programming.

NOMENCLATURE

RGB-D	RGB + Depth
CNN	Convolutional Neural Network
IEEE	Institute of Electrical and Electronics Engineers
ISO	International Standards Organization
IEC	International Electrotechnical Commission
NI	Natural Interaction
NITE	Natural InTeraction Engine (stylized as NiTE)
API	Application Programming Interface
HPC	High Performance Compute
GUI	Graphical User Interface
KNN	K-Nearest Neighbor
SVM	Support Vector Machine
SDK	Software Development Kit

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

INTRODUCTION

Design Problem

This project demonstrates a design for a lightweight skeleton-based person re-identification system. The goal of our project is to intelligently identify and track a person in real-time using skeleton data recorded by an RGB-D or thermal imaging camera. The system must reliably recognize individuals regardless of visual factors like clothing, hair, or accessories. The identification network runs on Raspberry Pi 4, a low-cost edge computing device, while skeleton tracking is handled by an external Windows device running camera-specific software. Multiple members from our team and the ODU Vision Lab will be used for testing, and the final system will allow additional person IDs to be added and stored after the initial training.

Function of the Design

Skeleton identification works by analyzing the unique movement style and body dimensions of an individual. The process begins with skeleton extraction, which analyzes video frames recorded by a camera. Camera-specific software estimates the position of a person within the frame and creates joint coordinates to form a skeleton. This approach discards extraneous information like color, allowing the system to process the data stream more efficiently. The recorded skeletal data is then fed into a convolutional neural network (CNN) that compares it against a database of known users. CNNs apply a series of convolutional filters to the input data, performing pooling operations between each step to reduce the number of parameters and recognize specific features [1]. After filtering is complete, the CNN flattens the result into a single identifier that corresponds to the person identified by the skeleton.

Quantitative Performance Objectives

The PrimeSense camera operates at a maximum frame rate of 30 FPS, and the PrimeSense NiTE middleware can track 15 skeleton joints per frame. Therefore, our skeleton extraction script must be able to process and display 450 sets of joint coordinates every second to achieve real-time skeleton tracking performance. While NiTE can detect multiple skeletons in one frame, the Python bindings library that we are using does not support this feature. Therefore, the system will only recognize one person at a time.

The FLIR component, however, operates within a different functional context and therefore utilizes its own qualitative performance criteria. The system achieves real-time operation with a skeletal extraction frame rate of 10 FPS using the FLIR camera. This requirement is determined by the performance limitations of OpenPose, which caps the frame rate due to the computational intensity of 25 key points despite the FLIR camera's native 50 FPS capability. OpenPose also supports tracking multiple skeletons per frame, however we reduce tracking to a single person to create our training data.

The overall goal is to achieve ≥80% recognition accuracy on our test set with the final post-training model. We chose this accuracy goal based on previous results with a network of the same architecture created and tested by Joseph Zalameda, a PhD student at the ODU Vision Lab. During his initial research and testing, he was able to achieve 80% recognition accuracy. However, several factors can affect this metric, including the length and quality of training data, positioning of the camera, and model hyperparameters. Our overall goal is to modify the given neural network to achieve the highest possible accuracy, but given the project's time constraint, 80% accuracy is a realistic quantitative metric.

DESIGN APPROACH

Design Constraints

Our original goal was to run skeleton extraction and person identification simultaneously on the Raspberry Pi. However, we found that skeleton extraction on ARM hardware is a complex task that could warrant a senior design project of its own. The design alternatives section goes into more detail about the issues we encountered during the skeleton extraction phase. Therefore, we are using a revised design that offloads skeleton extraction to an x86-based Windows machine. Even with the overhead gained by offloading skeleton tracking, performance remains our primary constraint. The Raspberry Pi 4 that we are using only has 4 GB of memory, which could limit the performance of our neural network. Another performance-related constraint of the Raspberry Pi 4 is the CPU. The Pi's quad-core Cortex-A72 (ARMv8) 64-bit processor, running at 1.8Ghz, lacks the core count and clock speed of standard laptop or desktop counterparts [2]. A further constraint is data transfer between devices. To maintain stability and reduce vulnerability, our goal is to use a wired connection rather than a wireless network.

An additional design constraint involves the FLIR A655SC thermal camera, which is prone to heat accumulation during extended periods of operation. Continuous use without adequate cooldown intervals may lead to thermal stress or potential damage to the camera's internal components. Proper temperature monitoring and operational cycling are therefore required to maintain performance stability to ensure long-term hardware reliability.

Design Challenges and Alternatives

Our original design approach involved running skeleton tracking and extraction algorithms alongside the identification network on Raspberry Pi to maximize cost-effectiveness

and simplicity. However, due to performance and compatibility issues, we chose to offload skeleton tracking to an external Windows device.

The primary issue with the PrimeSense approach is the age of the camera and its software. Because PrimeSense stopped operating independently after it was acquired by Apple in 2013 [3], it no longer provides updates and support for its devices and drivers. Our Raspberry Pi 4's 64-bit ARM architecture exacerbates this issue. PrimeSense released old versions of NiTE that are compatible with 32-bit ARM devices, but not 64-bit ARM devices. In addition, they were compiled without hardware floating-point support, which all modern operating systems require. Emulating this older system would be slow, and we found that the Pi's processor is not powerful enough to run emulated software alongside our CNN.

The NiTE software also has other limitations that we had to work around during this project. It is prone to memory leaks, meaning that the client cannot run unattended for long periods of time without restarting. Although this can be solved using an automatic restart script, it reduces the quality of the user experience. Another issue with NiTE is how it tracks an active user. It takes between 10-20 seconds to recognize that a user has left the frame, meaning that there is a significant delay before it can start tracking a new person. Finally, since PrimeSense as a company no longer exists, the software will not receive any future updates. This is not ideal for security applications, because it means that vulnerabilities will not be patched.

However, the third-party alternatives we researched were not any better. Skeleton tracking with depth data is not a trivial task, so there are few libraries available. Even fewer run on ARM devices and none provide functionality or performance on par with NiTE. The most popular method for skeleton tracking is to use a 2D image with a machine learning library that can extrapolate depth data for the joints. We tried several of these, including OpenPose and

MediaPipe, but ran into performance issues. In addition, these frameworks would require us to add the depth data from the PrimeSense camera manually to construct a 3D skeleton. This defeats the purpose of using a depth camera for this task and requires extra performance overhead. Therefore, we stuck with NiTE for our approach, although we note in future work that a priority for future teams should be creating a custom skeleton tracking algorithm.

Design challenges for the FLIR system pipeline center around data and inference. While much of the software has the potential to work together, such as FLIR Spinnaker and OpenPose, inside the process of data acquisition and re-identification of the subject, there is an unresolved issue. Since OpenPose was trained off RGB images and people in motion, data obtained while standing and in thermal greyscale will have joint localization noisier than if using a traditional RGB camera. This was tested through running small scripts designed to check centered means, normalized features, raw coordinates, and body features. All the tests reported consistent results that FLIR 2D reidentification would be less stable. Results include a standard deviation of 1-3 pixels on standing data, means within 0.025 of each other, and a signal to noise ratio of 1:9. Standard deviation would represent how measurements vary from frame to frame for one person, means would affect the average body proportion and joint position, and signal noise affects classification.

Alternative designs for the current FLIR component of the project included evaluating different cameras such as the Teledyne FLIR Hadron 640R and the Xenics Bobcat DCL-S. Initial plans were to also use the Hadron 640R as the primary sensor due to its dual-band visible and thermal configuration. However, its integration required additional hardware, firmware, and costs that exceeded the project's allocated budget. Furthermore, its dual-sensor architecture would have required a separate calibration and integration pipeline that would have been

impacted by waiting for part acquisition. This would have strained time considerations and complicated the software implementation process. The Xenics Bobcat DCL-S was also considered as a potential thermal imaging alternative. The Bobcat is a short-wave thermal infrared (SWIR) camera designed with InGaAs (Indium Gallium Arsenide) infrared detectors and 20 um pixel pitches [4]. This means that instead of operating in the 8-14 um range like long-wave infrared does, short-wave uses infrared bands of around 1-3 um. While it offers accommodating sensitivity, high dynamic range, and GigE interfacing, its short-wave thermal infrared technology is a direct contrast to the long-wave infrared technology used by the FLIR A655SC.

As mentioned earlier, our current design uses a convolutional neural network to perform re-identification. To attempt to boost accuracy and reduce model biasing, we also tested alternative learning models like K-nearest neighbor (KNN) and non-linear support vector machine (SVM). These differ from CNNs because they are single-layer shallow models, not multi-layer deep learning models. KNN works by using Euclidean distance to find the nearest neighbors of data points and classifies data based on the similarities of nearby data points. While we were able to match the training accuracy of our CNN results, we found that the KNN models after training were extremely biased towards one individual and resulted in lower full-pipeline accuracy.

To understand how a non-linear SVM works, it's easier to start by understanding linear SVMs. Linear SVMs work by finding a straight line that can separate a hyperplane in n-dimensional space and can split the data into categories based on the groupings split by the line. Non-linear SVM works by trying to find a spot in n-dimensional space using kernel functions to

create curved boundaries between the data, allowing classification of more complex, non-linear data. The figure below highlights the key differences between linear and non-linear SVMs.

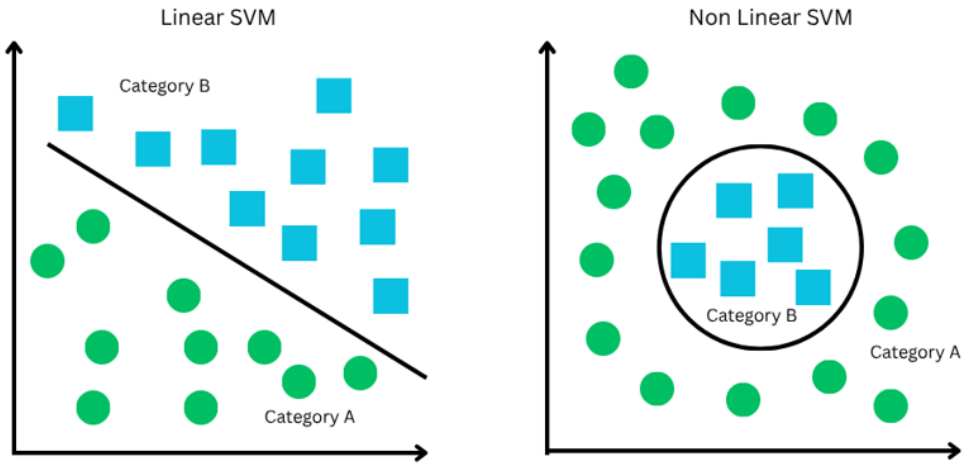


Figure 1: Linear vs Non-Linear SVM

An overview of key conceptual differences between the linear support vector machine and nonlinear support vector machine.

After implementing nonlinear SVM, we found that the resulting accuracy was equivalent to random guessing, meaning that the SVM was not correctly learning features from the training data. After determining the biasing issue with KNN and the accuracy issues with SVM we found that it was better to focus our attention on continuing to implement and fine-tune our CNN.

PROJECT DELIVERABLES

Team Responsibilities

Because we split the project into two separate approaches, we also split into two teams to work on both approaches in parallel. The first team, consisting of Seth Myers and Aubrey McKinney, focuses on the PrimeSense RGB-D approach. They are responsible for interfacing the camera through OpenNI, obtaining a skeleton through NiTE, and transforming the joint data into a PyTorch-compatible format. The other team, consisting of Antonio Mendoza and Noor

Humphreys, is working with the FLIR long wave thermal camera. They are responsible for interfacing the camera through VS Code and FLIR specific SDKs such as Spinnaker and extrapolating depth and skeleton data using machine learning through OpenPose. Both teams are jointly responsible for training and running a convolutional neural network, customized to match their camera's data format. Both teams also collaborated to create a user interface, implement a training data recording pipeline, and complete all required course assignments.

Timeline

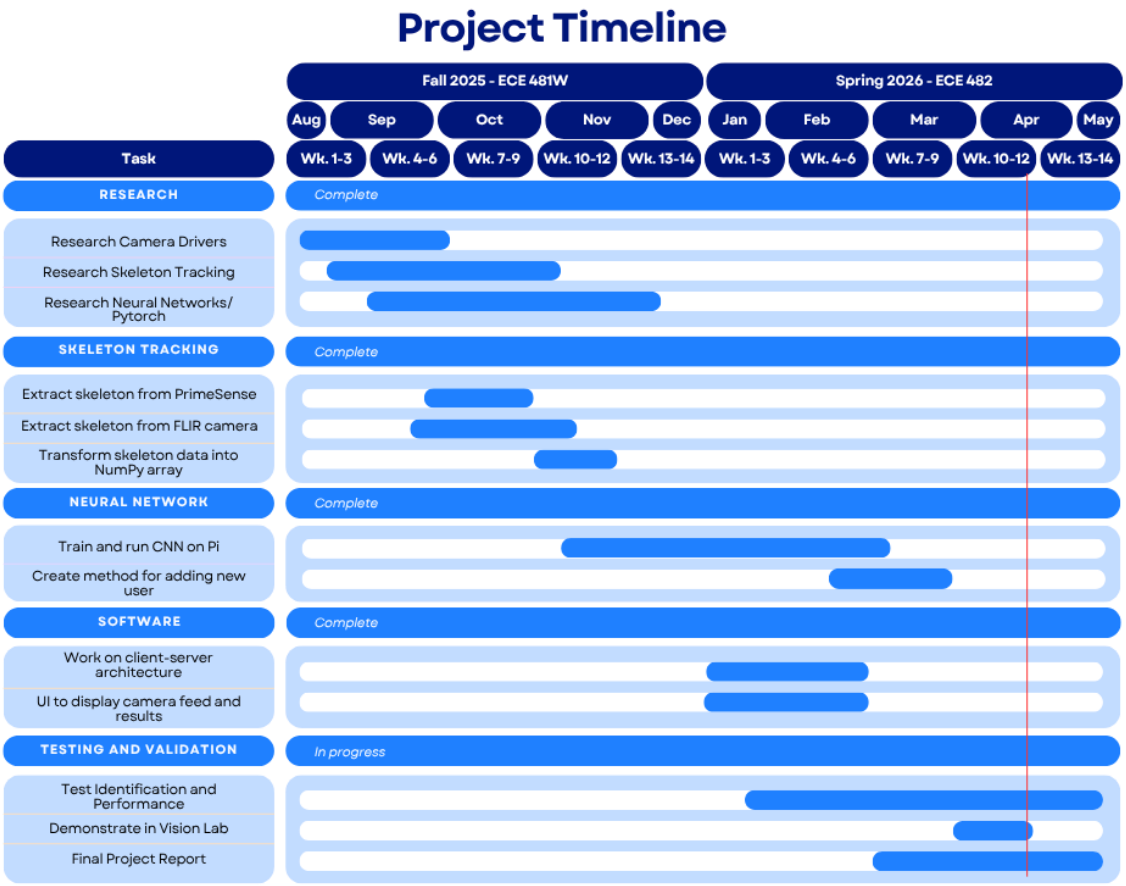


Figure 2: Project Gantt Chart

An overview of our schedule for both project semesters as of April 2026. It includes research, design, implementation, testing, and demonstration. Most of the first semester was spent on research and design of the skeleton extraction system, with the second semester focusing on implementing and training the neural network. We are currently working on refining our CNN and improving model performance.

DESIGN SPECIFICATIONS

Details of the Engineering Design

Our project and team are split into two separate approaches, although many of the design processes are shared. One method uses an RGB-D camera manufactured by PrimeSense, while the other uses a FLIR long-wave thermal camera. The main differences are related to interfacing with the cameras and extracting skeleton data, and our goal was to create an identification pipeline that is functionally equivalent for both approaches.

The PrimeSense RGB-D camera uses an infrared projector and RGB camera to produce a 640 x 480 depth map of its view [5]. The most popular use of this technology is Microsoft's Kinect 360, which utilizes a PrimeSense-created sensor to run skeleton tracking for gaming applications [6]. The PrimeSense camera connects to an x86-based Windows machine via USB. This machine runs PrimeSense's OpenNI drivers and proprietary NiTE skeleton tracking middleware.

To extract skeleton data from the PrimeSense depth map, we create a NiTE UserTracker object during the recording process. This object encapsulates all functions and data structures related to a tracked skeleton, including individual joints and their 3D coordinates. For visualization, we established a set of bones that connect two or more joints. We use a combination of OpenNI's depth-conversion methods and OpenCV to flatten the joint coordinates into a 2D image, which is overlaid on the camera feed with each frame. However, we also store a copy of the original 3D joints to send to the Raspberry Pi for identification.

Although OpenNI and NiTE expose C++ and Java APIs for accessing skeleton-related functions and data members, they do not provide a Python equivalent. Python is the most convenient language for our skeleton extraction pipeline because it allows us to integrate directly

with the identification pipeline. Therefore, we will be using a third-party open-source library to provide Python bindings for the C++ functions. It provides functions with similar syntax to the original C++ API, so we were able to utilize the original OpenNI API documentation [7]. We verified the functionality of the library with an OpenCV example program provided by the library author. During this preliminary test, we encountered several version conflicts that are important to note for reproducibility. We found that the OpenNI Python library requires an older version of Python that supports instantiating abstract classes. We chose to use Python 3.6.8 because external packages like OpenCV and NumPy are still available for this version. To avoid conflicts with newer Python installs, we created a Python virtual environment to work in. It's important to note that because of the dependencies on instantiating abstract classes in Python 3.6.8, newer versions of Python will not work with OpenNI. We also temporarily installed the older 3.4.3 release of CMake, which is required to install the older version of OpenCV. We only used CMake during the OpenCV installation process, so it is not required at runtime.

We designed our client GUI with Qt5, a widget-based application framework. This modular design allows us or a future team to add new features to the interface. We also chose Qt5 for its asynchronous signal-based updates, in which the UI only changes when it receives a predetermined signal from another part of the program. This allows parts of our software to operate independently from one another. For example, the RGB camera feed needs to update at a constant rate of 30 frames per second. Therefore, the client emits a "frame ready" signal whenever it receives a frame from the camera, regardless of skeleton data. However, if a recognizable user is in frame, their skeleton visualization is superimposed on each frame before emitting the "frame ready" signal. Once the skeleton batch queue is full, a separate "skeleton ready" signal signifies that the batch is ready to be sent to the Raspberry Pi. Whenever the client

receives a result from the server, a “result ready” signal tells the UI to update the displayed person ID prediction. This separation of updates prevents one waiting process from blocking the entire UI functionality.

For networking, the easiest and most reliable solution is a direct Ethernet connection between our Windows client and Raspberry Pi. The client-server architecture itself is implemented using asyncio, allowing the client and server to send and receive independently. To avoid writing to the server constantly, we use a buffer to accumulate a specified number of skeleton frames before sending the batch to the Raspberry Pi. At the same time, the client listens for identification responses from the server. Since the inference time may vary for each batch, we utilize a queue to store batches while waiting for a reply from the server. A request ID system allows the client and server to keep track of which result matches each batch.

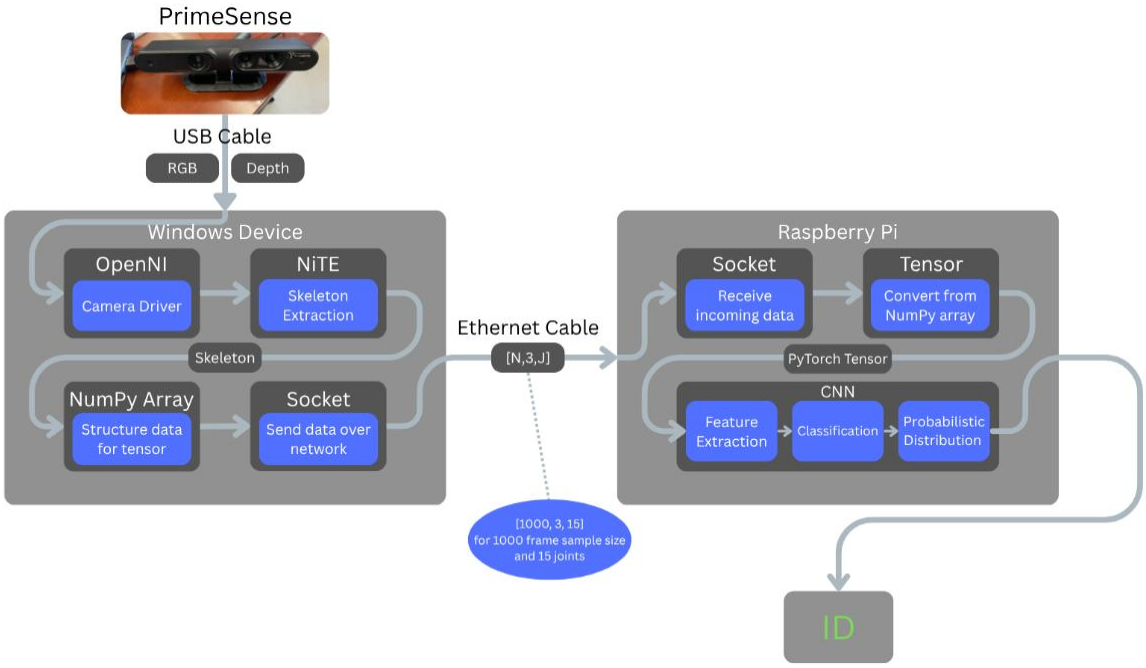


Figure 3: RGB-D Person Re-identification Process Flow Chart

This chart shows a basic overview of the skeleton re-identification process with the PrimeSense camera. It includes raw depth capture, skeleton tracking, neural network processing, and the result.

The FLIR design process begins with the unique nature of the camera. The FLIR A655SC is a long-wave infrared camera, providing a 2D 640 x 480 resolution and a GigE connection [8]. The camera will utilize the GigE connection to a Windows-based machine, which in turn will process skeleton image data through a script integrating OpenPose and FLIR software. The script includes multiple libraries, including OpenCV for image processing, FLIR Spinnaker SDK for camera control and frame acquisition, OpenPose for neural-network-based pose estimation, and standard C++ and Python libraries for data handling and post-processing.

During the skeleton extraction pipeline, thermal image data from the FLIR camera is first obtained in 16-bit Mono16 format. Due to OpenPose expecting an 8-bit, 3-channel imagery, the thermal frame is normalized using a percentile-based conversion (2-98%) and reduced to an 8-bit grayscale image. This grayscale image is then converted to the 3-channel BGR format using OpenCV to match OpenPose's input requirements. Once formatted correctly, the frame is wrapped into an OpenPose datum object, which serves as the container for both images and output pose data. The datum is passed through OpenPose's neural network-based pose estimation pipeline, where body parts are detected and 2D skeletal key points are generated.

In the current implementation of the FLIR skeletal extraction method, only the first detected person is used. For that individual, the (x, y) coordinates of each joint are extracted. The script has potential to be expanded into detection and processing of multiple people; however, we are currently focusing our efforts on obtaining accurate and reliable data regarding a single individual. For user interfacing with the FLIR thermal camera, it is developed in a similar architecture to that of the PrimeSense. It uses a modular system implemented using Qt5 to allow for process separation and synchronization. The C++ main file has been edited to become a backend application responsible for hardware acquisition and processing. After the frames are

captured from the FLIR camera and converted, the skeleton data and visualization are then overlayed and streamed over a TCP socket to the Qt5 python user interface script. The UI would run on its own loop and receive the frames and key point arrays through the stream client. Qt signals are used internally to allow the UI to include frame updates, skeleton buffers, and status messages across the stream without blocking rendering. Due to the FLIR camera occupying the system's primary Ethernet interface, a USB-to-Ethernet adapter or Wi-Fi connection is required when including the Pi for additional networking and data communication. The process is once again similar to the PrimeSense team's process to communicate with the Pi to keep the project consistent across both cameras.

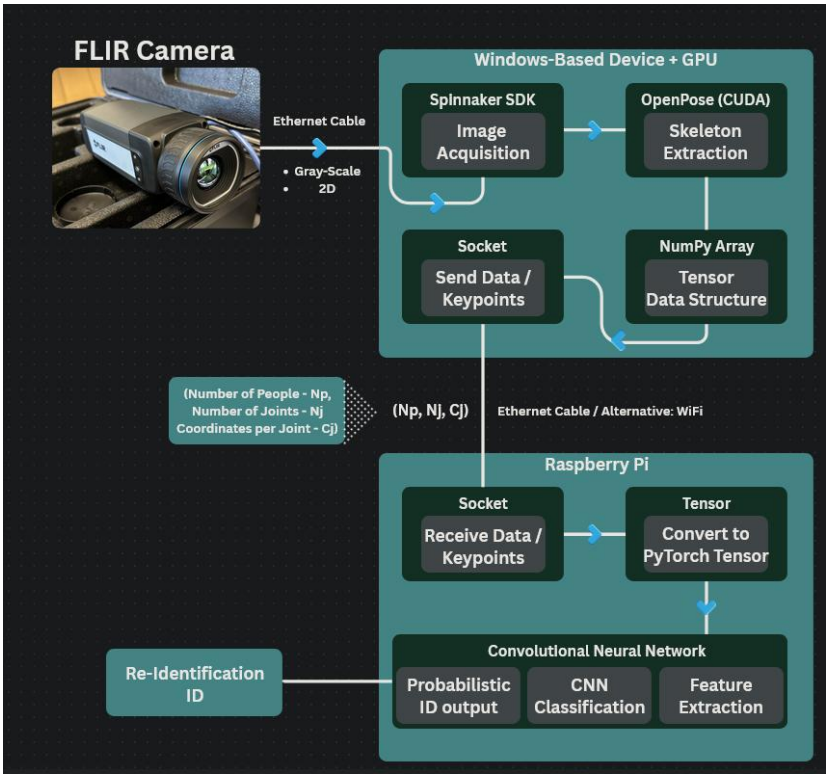


Figure 4: FLIR Person Re-identification Process Flow Chart

This chart shows a basic overview of the skeleton re-identification process with the FLIR camera. It includes raw thermal capture, skeleton tracking, neural network processing, and the result.

As for data handling, both approaches transform their respective coordinate data into a NumPy array for later processing into a PyTorch tensor. In general, this array has the dimensionality [N, D, J]. N represents the number of frames being recorded, D represents the dimensionality of the joint coordinates, and J represents the number of skeleton joints. For PrimeSense, this array will have a final size of [N, 3, 15] because NiTE provides 15 joints with 3D coordinates. For the FLIR camera, the array will be [N, 2, 24] because FLIR software provides 24 joints with 2D coordinates. This means that although the training and identification pipelines are similar, we still utilize two different training scripts and networks for each approach to accommodate these different tensor shapes.

For the identification task, we implemented a CNN created by PhD student Joseph Zalameda in the ODU Vision Lab. We trained this CNN on skeleton data recorded by our team using our skeleton extraction pipeline. Our current training procedure is partially based on a procedure used to evaluate the BIWI RGBD-ID skeleton dataset [9]. We record two sets of skeleton data for each person we plan to add to the network. During the first recording, the participant will stand completely still. The second recording will involve the participant walking in a straight line to and from the camera. Both recordings have a final length of 1000 frames, although we typically record extra frames at the beginning for calibration purposes. These extra frames are pruned from the array by our training script.

The training script also performs other important functions like normalizing our dataset, performing cross-validation, and saving a class map. Normalization is the process of reducing the scale of data to a given range without changing its distribution. This is important in machine learning because very large values can often overpower smaller ones, causing the model to become extremely biased. We use the minimum and maximum values from the entire dataset to

normalize between 1 and 0, where the largest value will be 1 and the smallest will be 0. This means that all values in the dataset are sized relative to the dataset itself, allowing for a more balanced comparison. We also save statistics like the minimum value and range for each dimension to a file during training. These statistics will be loaded by the final model running on the server. This provides a consistent normalization environment for real-time data from the skeleton extraction pipeline. Cross-validation is a method of training that uses part of the training data for testing, providing an early indicator of the model's performance. It will be discussed in more detail in the analysis and testing section of the report. Finally, the class map is a simple data structure that maps integers to class names, which are person names in our case. This is needed because the neural network can only accommodate numbered classes. The class map allows us to work around this issue and provide a better user experience by storing person names separately to be displayed in the UI.

Parts and Software List

Part/Software:	Purchased/Acquired from:
Raspberry Pi 4 – 4GB model	Provided by Vision Lab
PrimeSense Carmine 1.09 RGB-D camera	Provided by Vision Lab
5 V / 3 A USB-C power supply	Provided by Vision Lab
FLIR A655SC Long-wave Thermal Camera	Provided by Vision Lab
FLIR Spinnaker SDK	Downloaded from https://www.teledynevisionsolutions.com/support/support-center/software-firmware-downloads/iis/spinnaker-sdk-download/spinnaker-sdk--download-files/
OpenPose	Downloaded from https://github.com/CMU-Perceptual-Computing-Lab/openpose

OpenCV	Included with OpenPose, used in FLIR image conversion.
x86-based Windows machine	Provided by group/Vision Lab
OpenNI 2.0 software	Downloaded from https://structure.io/openni/
PrimeSense NiTE 2.2 software	Downloaded from https://bitbucket.org/kaorun55/openni-2.2/src/master/
Python 3.6.8 (for OpenNI drivers)	Downloaded from https://www.python.org/downloads/release/python-368/
Python 3.11 (for PyTorch)	Downloaded from https://www.python.org/downloads/
CMake 3.4.3	Downloaded from https://cmake.org/files/v3.4/
Python package:	**pip package name:**
PyTorch	pytorch
OpenCV (for visualization)	opencv-python
OpenNI Python Bindings	openni
NumPy (for creating PyTorch tensors)	numpy
Qt5	PyQt5
SciKit Learn	scikit-learn

Table 1: List of required software and hardware components.

Most of the hardware was provided by the ODU Vision Lab. In addition, there is a list of Python packages that we used and their pip package name.

Analysis and Testing

To perform initial testing on the skeleton extraction portion of our project, we established the camera configuration, shown in Figure 5, which allows us to record data from both cameras in the same environment. Next, we created a standardized process in which frames were taken of the subject standing still at a fixed distance from the camera, walking along a defined forward-and-back path, and posed in various orientations and distances. The joint coordinates and video frames for each pose were also stored for quantitative evaluation. Figure 6 displays an example

of skeleton tracking obtained through both the FLIR and PrimeSense cameras. We can see that the tracking correctly fits a single individual as expected. This also allowed us to develop and test our training data recording pipeline, which uses the same methodology. This approach allowed us to observe a wide range of skeleton joint configurations and find limitations in each camera's capabilities. The primary limitations we found are distance, environment, and the number of individuals that can be tracked. The PrimeSense camera has a short range, losing consistent tracking when the subject is standing more than 12 feet from the camera. The FLIR camera works at a much longer range of 24 feet, but subjects standing closer to the camera have joints out of frame. Finally, although both cameras are capable of tracking multiple people in the frame at once, the software and our pipeline design limit tracking to one person at a time.

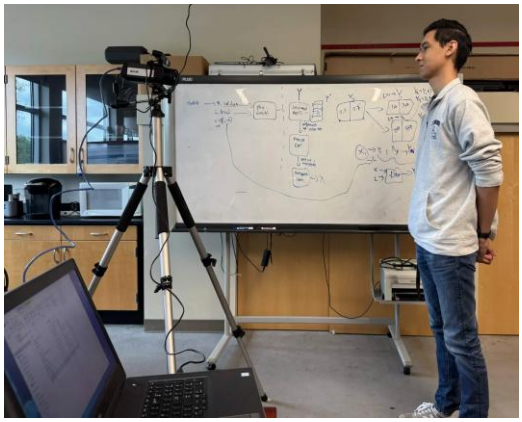


Figure 5: Camera Set Up for Data Acquisition

The figure above demonstrates the current camera equipment set up. Both cameras will be attached to the same tripod. The person will stand at a fixed distance from the camera and/or walk in predetermined patterns.

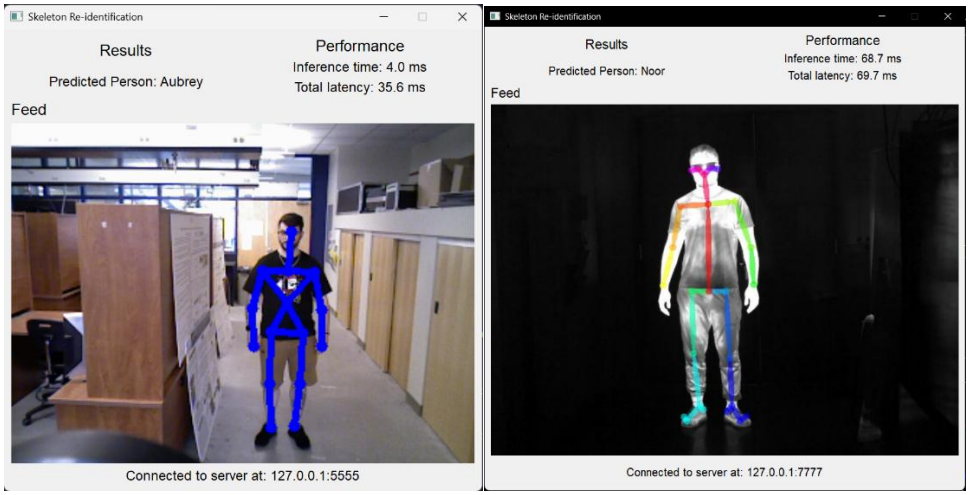


Figure 6: Tracked skeleton on PrimeSense and FLIR cameras respectively.

The figure above shows some preliminary tracking results we were able to achieve with each camera. The key points represent skeleton joints, and the lines represent bones.

The necessity of different poses stems from testing a variety of data metrics. The standing pose typically gives the most accurate joint data because there is no variation, so most of our training data uses the standing pose. The walking pose allows us to gather quality metrics about skeleton accuracy during motion and turning at each end of the walk tests temporary occlusion of joints. We evaluated the tracking quality quantitatively by creating data plots of the training data and looking at the deviation. For the standing recording, the plots should be constant with low deviation. Walking recordings tend to produce a sine wave-like plot as the subject gets closer or farther from the camera. Spikes in either plot show uncertainty or erratic joint movement. Figure 7 shows an example of standing and walking data for a single joint, taken directly from a recording of someone in our training dataset. We can see that plots match the expected patterns for walking and standing data. This metric is augmented by qualitative analysis of video frames from the recording with superimposed skeleton data. Qualitative frame analysis allows us to match spikes in the graph to factors like occlusion and distance limitations.

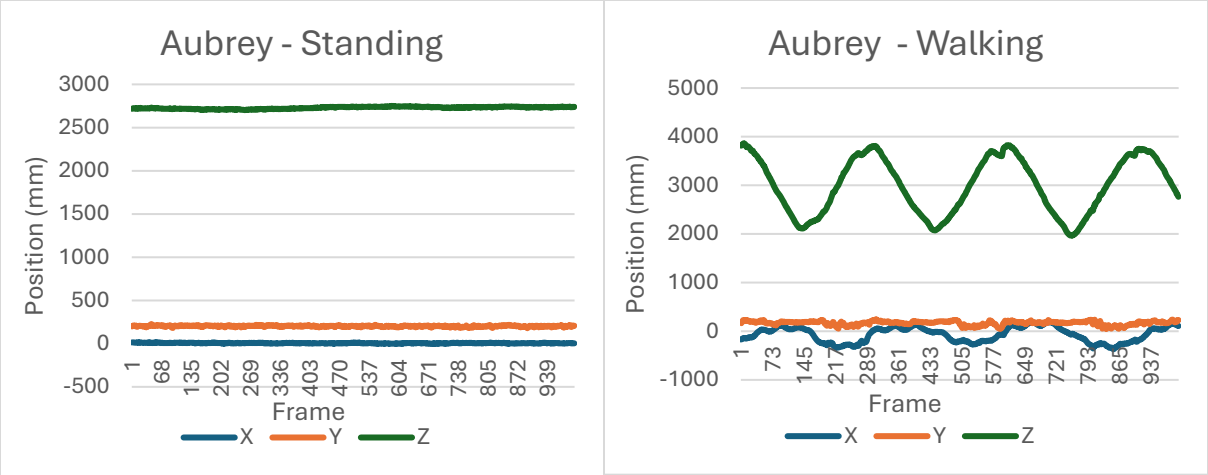


Figure 7: Training data quality metrics for a single joint (3D)

The left plot shows x, y, and z coordinates for a recording where the participant was standing completely still. Therefore, the values are nearly constant with only small fluctuations. The right graph is from a walking recording. We can see small variations in the x coordinate as the subject shifts horizontally when walking, while the y coordinate stays mostly constant. The z coordinate is the biggest differentiator, as it visualizes movement towards and away from the camera.

Our model testing process consists of three stages. Firstly, during model training we use a method called k-fold cross-validation. This involves splitting the training set several times, varying the location of the split, and using the smaller portion for validation. For example, with 5-fold validation you'd take 20% ($1/5^{th}$) of the training data for validation, using a different validation slice for each fold. This provides two quality values in the form of training accuracy and validation accuracy. These values are printed after each epoch, which represents one full pass through the training set. In addition, a confusion matrix is printed after each fold to show more detailed performance and bias statistics of the best performing epoch. This stage presents the best opportunity to catch obvious issues with our data before completing the training process. During the next testing stage, we load each fold's trained model with a separate testing dataset that was recorded in the same environment as the training set. Testing a separate dataset allows us to verify that the model is learning features, not just memorizing the training set. We choose the fold with the highest test accuracy to be our final model. Once we get a model we are

satisfied with, the final stage is a qualitative pipeline test. We load the model onto the Raspberry Pi, have people in the dataset step in front of the camera, and observe the systems' re-identification results.

	PrimeSense	FLIR
Training Accuracy	98-100%	90%
Validation Accuracy	95%	85%
Test Accuracy	65-75%	50%
Pipeline Test Accuracy	75% (3 out of 4 people)	25% (1 out of 4 people)

Table 2: Testing results for both cameras

We can see from the table above that both approaches were able to achieve high training and validation accuracy, showing that the model is learning. Despite this, our test results leave significant room for improvement. In addition, many of our trained models with high accuracy values have biasing issues, where some people are recognized more easily than others. As stated in our performance objectives, we hoped to achieve test accuracy of greater than 80%. However, we believe that our results still show that our solution is viable given more time to investigate model improvements, alternatives, and discrepancies. In the future work section, we will discuss some alternative solutions that we would have liked to try, given more time. These will also present opportunities for continued work on this project by another design team in the future.

Future Work

As we can see from our results, we were not able to meet our accuracy goal within the timeframe of the project. However, given more time there are several options we would like to have tried. The first option is changing our model. As discussed in the alternative designs section, we tried two other models – KNN and SVM, although we could not test them in detail due to the time constraint. We could also modify our CNN, e.g. by using a different activation

function that may better fit our data. In addition to the accuracy issues, the models can also be biased towards recognizing some individuals over others. In some cases, the model may not recognize some people at all, even when the results of test set validation look evenly distributed. To mitigate this issue, we are looking at many of the same options as with the accuracy problem. In addition, the system does not currently have a way to detect if the user does not exist in the dataset. Therefore, they will be identified as the person closest to their dimensions. We believe that this issue could be remedied by adding a confidence metric, where if a certain threshold of similarity is not met, the system displays an unknown value.

As discussed in design alternatives, we also believe that the current skeleton tracking implementations also leave room for improvement. With more time, we would want to create custom skeleton tracking and extraction algorithms that give us more flexibility and control over the system. In addition, this may allow us to streamline the system by running both skeleton extraction and inference on Raspberry Pi. However, this task is nontrivial and may warrant a dedicated senior design project. As discussed in the engineering standards section of this report, there are also security-related features that we would like to implement that fall outside the scope of the project. For example, the connection between the Raspberry Pi and the display client should be encrypted to prevent data interception, especially on wireless networks. This would require significant research outside the main objective of our project.

Another candidate for future work is scalability. While it is our goal to create a modular and scalable system, we did not have the extra hardware, budget, or time to implement and test multiple cameras running on the same network. These limitations are good candidates for future research in this area, and we hope that research into this technology will continue after the conclusion of our project.

ENGINEERING STANDARDS

9868-2025 ISO Standard for Biometric Identification

The ISO/IEC 9868-2025 standard provides recommendations and requirements for the design, development, use, and maintenance of biometric identification systems involving passive capture subjects, including pre- and post-deployment evaluation [10]. When working with the automated recognition of individuals based on unique physical characteristics, consideration into privacy protection, bias, and accuracy is important. This regulation defines high-risk systems as those that pose a risk of harm to the fundamental rights, health, or safety of individuals [10]. Since the skeletal reidentification system uses biometric data to detect and identify individuals based on their individual characteristics, adhering to this standard is crucial to maintain compliance. For example, high-risk systems such as the skeletal reidentification system must comply with this standard as part of the European Union (EU) regulatory requirements to be put on the European market [10]. It is essential to follow engineering standards and guidelines, as this product is designed to be easily scalable and suitable for use in various markets. The ISO/IEC 9868-2025 also provides guidance for security systems related to biases and accurate information. It is also vital that when our system reidentifies an individual based on their skeletal data, the results are accurate and do not misidentify someone. When used in a security application, accuracy is of utmost importance.

7000-2021 – IEEE Standard for Ethical Considerations

Ethical considerations are essential when working with autonomous security applications, making IEEE 7000-2021 highly relevant to our project. This standard supports the design of systems with explicit consideration of individual and societal ethical values, such as transparency, sustainability, privacy, fairness, and accountability, as well as values typically

considered in system engineering, such as efficiency and effectiveness [11]. Machine learning technologies, such as CNN-based identification systems, raise ethical concerns such as wrongful identification, biased decision-making, and data breaches that may involve sensitive information. Additionally, because many publicly available training datasets lack diversity in race, ethnicity, or cultural representation, they can also introduce bias into neural networks [12]. Following this standard means that we must try our best not to limit our dataset to prevent bias from occurring in our convolutional neural network. It is also our responsibility under this standard to limit the amount of personally identifiable information that is stored.

27001-2022 – ISO Standard for Information Security

With cybercrime on the rise and new threats constantly emerging, following ISO/IEC 27001-2022, or at least the principles of this standard, will help mitigate threats to our security system. The goal of this standard is to provide a framework for establishing, implementing, and maintaining an information security management system [13]. This standard applies to the project as we are using raw data collected on individuals to later re-identify them based on their skeletal joint data. Following the principles of this standard ensures the secure handling of sensitive tracking data. We also know that Raspberry Pi can be prone to attacks, and following this standard can help encourage encryption and secure communication protocols between devices. Security practices such as data protection, risk management, testing & validation, and system hardening are key components of this standard. We can maintain these components by practicing ISO/IEEE 27001 security practices, such as using secure file formats and avoiding storing raw identifiable information.

25010-2024 – ISO Standard for Software Quality Characteristics

Although several engineering standards apply to this project, time constraints prevent us from fully implementing all related requirements. ISO/IEC 25010 defines nine software quality characteristics: functional stability, performance efficiency, compatibility, interaction capability, reliability, security, maintainability, flexibility, and safety [14]. Product efficiency represents the degree to which a product performs its function within a specified time and is an efficient use of resources such as CPU, memory, and storage [14]. Although Raspberry Pi's low-power computer already forces us to be efficient, it is still possible to optimize our system further for faster operation and reduced computing power. For other areas of software quality characteristics, we focus on a subset, such as compatibility and safety. For instance, we can ensure the system's ability to exchange information with other products and the neural network's compatibility with any device supporting Python, thereby meeting the compatibility requirements. We also use a generic TCP server to send and receive the skeleton data, meaning that a different client could replace ours in the future without modifying the inference pipeline.

PROJECT BROADER IMPACTS

Economic Impact

The development and deployment of a light-weight skeleton-based person re-identification system presents several noteworthy economic advantages, specifically that it provides scalable and cost-effective advanced security solutions. By leveraging affordable hardware, such as the Raspberry Pi 4, and repurposed depth sensors like the PrimeSense Carmine or Xbox Kinect, the system significantly reduces the financial barrier associated with advanced security surveillance technologies. This low-cost skeletal re-identification project allows accessibility to small

businesses, educational institutions, and rural communities that may lack the necessary resources for industry-grade biometric systems. Recent advancements in skeleton-based re-identification highlight the growing availability of precise and economical tracking devices, which simplify mass deployment and reduce infrastructure demands [15].

In economically ravaged regions where security infrastructure is often undeveloped due to limited income and high unemployment rates, affordable surveillance solutions are especially critical. Studies show that in many low-income countries, falling economies and high unemployment rates have left them with unparalleled growth in crime rates [16]. Yet, when faced with poverty, ordinary people fail to see the need to invest in security, and those who do invest barely spare money on efficient forms of security [16]. The cost-effective aspects of this skeletal re-identification system create a more practical approach to a modern security system. By providing enhanced security at a lower cost, we can create more security in poverty-ridden communities. Additionally, the use of open-source software and customizable design minimizes licensing costs and supports flexible development. This further enhances the economic advantages and long-term sustainability of the skeletal re-identification system.

Environmental Impact

Graphics Processing Units (GPUs) play a significant role in real-time processing capability for AI video surveillance, such as facial recognition and person identification in public settings. Modern AI surveillance systems help assist authorities in law enforcement by tracking individual movements and deducing specific behavioral patterns [17]. However, this approach to AI security can be energy costly, which can be threatening to the environment and the electrical grid. The skeleton-based person re-identification system offers a more sustainable approach by using low-power-consuming hardware, such as the Raspberry Pi 4, to perform similar machine

learning tasks to the GPU-based servers, specifically convolutional neural networks. This approach significantly reduces energy consumption compared to traditional GPU-based surveillance systems. Not only does it reduce energy consumption, but it also continues to allow accurate real-time tracking of individual movements and behavioral analysis.

The system's energy efficiency makes it particularly useful for deployment in remote or rural areas, where it can operate in solar-powered or off-grid environments. Furthermore, the reuse of existing hardware, such as the Raspberry Pi 4 and PrimeSense Carmine, helps mitigate electronic waste by using already existing technologies. With the use of low-power components such as the Pi 4, we can lower the system's overall carbon footprint. Also, with flexible architectures like PyTorch, we can provide incremental upgrades without requiring complete hardware replacement. The cost-effective and scalable design choice contributes to a more environmentally responsible approach to real-time surveillance.

Societal Impact

One significant societal impact of the skeleton reidentification system is the ability to enhance public safety by providing real-time tracking in crowded or low-light environments. When trained appropriately, the skeletal reidentification can be used for missing persons identification if the convolutional neural network is trained to track and re-identify that specific individual. Furthermore, skeletal reidentification enhances public safety by providing a security system to rural or low-income communities, as discussed previously.

Security agencies in most developing countries still have stale approaches to forensic investigations, being unable to gather quality data from crime scenes and analyze this data through science and technology-aided methods [15]. This means that even after a crime has been committed, there is a low possibility of narrowing down a suspect via biometric features. The

skeletal reidentification system allows individuals to be re-identified based on their individual movements and can provide better quality data. The skeletal reidentification system also avoids ethical issues such as racial, gender, and cultural biases often found embedded in facial recognition software. This means that this system is more likely to be accepted in communities concerned with surveillance overreach.

ETHICAL IMPLICATIONS

Project Ethics

To uphold our duty to the public, we must design with safety, sustainability, and privacy in mind [18]. For our project, the latter is a paramount issue. Any device that uses a camera to track and identify individuals has an inherent privacy risk if data is mishandled. In the event of a data breach, we must immediately notify affected individuals and investigate the cause. The Code of Ethics prevents engineers from hiding this information from the public [18]. We will also attempt to mitigate this risk by only storing data that is necessary for functionality. This includes extracted skeletal data and a database of associated identities. It may also include the monochrome depth maps produced by the PrimeSense camera. As discussed previously, we avoid storing the color image from the camera since it may expose extraneous information like facial features and clothing. Securing the data that we store is also important. Even skeletal data becomes sensitive when linked to an identity. Bad actors may use this information to harass individuals or even falsely implicate suspects in law enforcement settings. Although the implementation of encryption and other safety measures were outside the scope of this project, it is still important to consider them for future work.

According to the IEEE Code of Ethics, engineers must “treat all persons fairly and with respect, not engage in harassment or discrimination, and avoid injuring others” [18]. This applies to everyone who interacts with our product, including our colleagues. Our goal is to identify all people with equal accuracy regardless of their race, gender, or other features. This is why we believe skeleton tracking is a great approach to identification. Human body dimensions are unique, yet independent of clothing, race, or gender. This allows us to perform identification without storing any information that would allow for discrimination. The Code of Ethics also prevents engineers from engaging in stalking or harassment [18]. We understand that bad actors may still try to use our technology as a tool for stalking and harassment. While this risk is beyond our control, it is important to consider it during design and ensure that we take every precaution against it.

Group Ethics

The above principles apply to everyone working on our team. We must not forget that our colleagues are also members of the public, and we should treat them with the same level of respect and concern. We must support all members of our team by communicating with one another and assigning work equally. We should be able to resolve conflicts amongst ourselves and voice concerns without fear of retaliation. It is the responsibility of every engineer on the team to ensure that their colleagues are acting ethically and to report violations of the Code of Ethics [18]. We will report all ethical violations to the appropriate supervisor or senior team member. If no one is willing to correct these violations internally, we must report them to the proper authorities and regulators privately. We will avoid involving the public in internal team conflicts without first resolving any conflicts of interest. There will always be new ethical concerns that we have not encountered yet, even beyond this project. Therefore, we continue to

evaluate our values as a team throughout our project and communicate new ethical issues to each other. Maintaining ethical awareness enables us to serve the public responsibly and preserve their trust.

LIFELONG LEARNING

This project has taught us many practical skills primarily related to software and hardware interfacing. Our research for this project taught us about the foundations of machine learning, which is the basis for all artificial intelligence. Both teams learned how neural networks make connections between data and the differences between shallow and deep learning models. We also gained experience with programming, structuring data, NumPy arrays, and APIs by working with Python, OpenNI and NiTE APIs. Designing the asynchronous UI and client-server architecture of our system allowed us to gain networking, user interface, and parallel processing experience.

The PrimeSense team has continued to learn about supporting and interfacing with legacy hardware. The challenges we encountered while trying to obtain working drivers allowed us to learn about hardware architectures and build tools. From a hardware perspective, we were able to learn about using Python and OpenNI to interface with our camera. We were then able to learn about data manipulation and visualization as the project progressed into the training data stage.

As for the FLIR team, this part required integration of multiple complex systems such as thermal imaging hardware, different coding frameworks, and deep-learning libraries, all of which reinforced the importance of multidisciplinary learning. Combining independent research and knowledge from computer vision, system embedding, and programming provided us with experience in resolving cross-domain engineering challenges across electrical and computer engineering fields. The FLIR team gained insight into how real-world imaging pipelines operate,

from low-level data gathering through Spinnaker, to preprocessing with OpenCV, to high-level neural network inference via OpenPose. Both teams employed learning strategies such as gathering and reviewing documentation, studying SDK and library references, collaborating through peer discussions, and iteratively debugging and refining implementations. These strategies helped accumulate new knowledge and helped build upon prior experience, contributing to a more complete and robust background in our fields.

Our assignments and reports for this project provided us with more generalized design skills that will be useful as we transition from our academic careers to professional careers. The engineering design process encapsulates design problem analysis, solution brainstorming, research, standards compliance, prototyping, and testing. We also learned how to continually evaluate the global and ethical impacts of our project, which are essential to fulfilling our duty to improve the lives of the public. These concepts will continue to be relevant in any engineering field and will guide us in future projects. Because this project sharpened our skills in so many different areas of the design process, we believe that it was an essential stepping stone to a more well-rounded understanding of the broader field of engineering.

REFERENCES

[1] MathWorks, "What Is a Convolutional Neural Network?," [Online]. Available: https://www.mathworks.com/discovery/convolutional-neural-network.html. [Accessed 27 October 2025].

[2] Raspberry Pi Foundation, "Raspberry Pi 4 Model B specifications," [Online]. Available: https://www.raspberrypi.com/products/raspberry-pi-4-model-b/specifications/. [Accessed 23 September 2025].

[3] Reuters, "Apple buys Israel's PrimeSense for $345 million: report," 2013. [Online]. Available: https://www.reuters.com/article/technology/apple-buys-israels-primesense-for-345-million-report-idUSBRE9AG03G/. [Accessed 26 October 2025].

[4] Exosens, "Bobcat Affordable SWIR Cameras," Exosens, [Online]. Available: https://www.exosens.com/products/bobcat. [Accessed 23 November 2025].

[5] imagefact.de, "RGB-D depth cameras," [Online]. Available: https://www.scanner.imagefact.de/gb/depthcam.html. [Accessed 13 September 2025].

[6] H. Haggag, M. Hossny, D. Filippidis, D. Creighton, S. Nahavandi and V. Puri, "Measuring depth accuracy in RGBD cameras," in *7th International Conference on Signal Processing and Communication Systems*, 2013.

[7] S. Lemaignan, "OpenNI2/NiTE2 Python Bindings," [Online]. Available: https://github.com/severin-lemaignan/openni-python/blob/master/README.md. [Accessed 23 November 2025].

[8] FLIR Systems, "FLIR A655SC High-Resolution Science Grade LWIR Camera," FLIR Systems, 2024. [Online]. Available: https://www.flir.com/products/a655sc/. [Accessed 23 November 2025].

[9] M. Munaro, "BIWI RGBD-ID Dataset," [Online]. Available: https://robotics.dei.unipd.it/reid/reid/index.php/downloads/8-dataset/2-overview-biwi. [Accessed 21 November 2025].

[10] International Organization for Standardization, "Information Technology - Design, Development, Use and Maintenance of Biometric Identifcation Systems Involving Passive Capture Subjects," *ISO,* 2025.

[11] IEEE, "IEEE Standard Model Process for Addressing Ethical Concerns during System Design," *IEEE STD 7000,* vol. 1, pp. 1-82, 2021.

[12] IEEE, "IEEE Standard for System, Software, and Hardware Verification and Validation," *IEEE Std. 1012-2024,* 2025.

[13] International Organization for Standardization, "Information security, cybersecurity and privacy protection - Information security management systems -Requirements," *ISO/IEC 27001,* 2022.

[14] ISO/IEC, "Systems and Software Engineering — Systems and Software Quality Requirements and Evaluation," *ISO/IEC 25000,* 2022.

[15] R. Haocong and e. al, "Recognizing Identities from Human Skeletons: A Survey on 3D Skeleton Based Person Re-Identification," College of Computing and Data Science, Nanyang Technological University, [Online]. Available: arxiv.org/html/2401.15296v2. [Accessed 20 September 2025].

[16] M. Emetere and e. al., "Review on Home Security System in Developing Countries: Affordability or Comfortability," TheSCIpub, 22 Febuary 2022. [Online]. Available: thescipub.com/pdf/jcssp.2023.415.430.pdf. [Accessed 20 September 2025].

[17] C. Fontes, E. Hohma, C. Corrigan and C. Lutge, "AI-powered public surveillance systems: why we (might) need them and how we want them, Technology in Society," *ISSN 0160-791X,* vol. 71, 2022.

[18] IEEE, "IEEE Code of Ethics," [Online]. Available: https://www.ieee.org/content/dam/ieee-org/ieee/web/org/about/corporate/ieee-code-of-ethics.pdf. [Accessed 3 October 2025].

[19] IEEE, "IEEE Recommended Practice on Software Reliability," *IEEE Std. 1633-2016,* 2017.

[20] IEEE, "IEEE Standard for Algorithmic Bias Considerations," *IEEE Std. 7003-2024,* 2025.

[21] IEEE, "IEEE Standard for Biometric Privacy," *IEEE Std. 2410-2021,* 2021.

[22] IEEE, "IEEE Standard for Data Privacy Process," *IEEE Std. 7002-2022,* 2022.

[23] NVIDIA Corporation, "Jetson Nano System-on-Module Data Sheet," [Online]. Available: https://openzeka.com/en/wp-content/uploads/2022/08/JetsonNano_DataSheet_DS09366001v1.1-1.pdf. [Accessed 23 September 2025].

[24] Apple Inc., "Secure Enclave," 2024. [Online]. Available: https://support.apple.com/guide/security/secure-enclave-sec59b0b31ff/web. [Accessed 12 October 2025].

[25] A. Bate, "Thermal testing Raspberry Pi 4," Raspberry Pi, [Online]. Available: https://www.raspberrypi.com/news/thermal-testing-raspberry-pi-4/. [Accessed 23 September 2025].

[26] T. Rice, "Real-time inference on Raspberry Pi 4 (30 fps!)," [Online]. Available: https://docs.pytorch.org/tutorials/intermediate/realtime_rpi.html. [Accessed 13 September 2025].

[27] A. Shehabi, A. Newkirk, S. J. Smith, A. Hubbard, N. Lei and M. A. B. Siddik, "2024 United States Data Center Energy Usage Report," Lawrence Berkeley National Laboratory, 2024.

[28] C. Prince, N. Omrani, A. Maalaoui, M. Dabic and S. Kraus, "Are We Living in Surveillance Societies and Is Privacy an Illusion? An Empirical Study on Privacy Literacy and Privacy Concerns," *IEEE Transactions on Engineering Management,* vol. 70, no. 10, pp. 3553-3570, 2023.

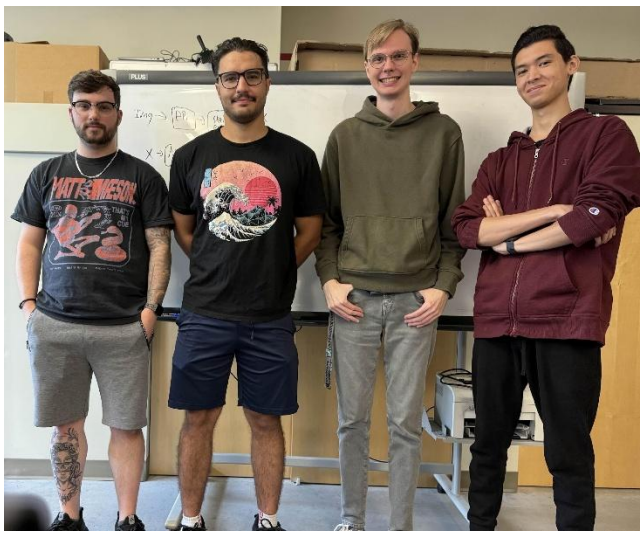
[29] A. Khalil, S. Ahmed, A. Khattak and N. Al-Qirim, "Investigating Bias in Facial Analysis Systems: A Systematic Review," *IEEE Access,* vol. 8, pp. 130751-130761, 2020.

[30] IEEE, "IEEE Standard for Biometric Privacy," IEEE, 2021.

APPENDICES

Appendix A: Executive Summary

The overall development of the lightweight skeleton-based re-identification project was overseen by Dr. Khan Iftekharuddin. To assist with development, secondary management was provided by Joseph Zalameda, a PhD student in the ODU Vision Lab. Group members included in the development of the project are Antonio Mendoza, Aubrey McKinney, Seth Myers, and Noor Humphreys. Aubrey, Antonio, and Seth are undergraduate Computer Engineering students, while Noor is an undergraduate Electrical Engineering student. The figure below depicts the group members, omitting Dr. Iftekharuddin and Joseph Zalameda.



The system is designed to operate through an integrated configuration consisting of a Windows-based client machine, camera, and Raspberry Pi microcomputer-based server. The client computer is responsible for capturing raw image frames and data from the camera, performing skeleton extraction, and structuring the output for transmission. The Raspberry Pi then receives processed skeletal data and utilizes a trained convolutional neural network (CNN) to interpret the information and create a unique identification output. This design approach

addresses the computational limitations of the Raspberry Pi while accommodating the differing capabilities of the connected cameras. By implementing a streamlined data pipeline, the system ensures efficient data transfer, maintains real-time processing capabilities, and preserves accuracy for reliable person re-identification.

Appendix B: Public Slides

ODU
Department of Electrical & Computer Engineering

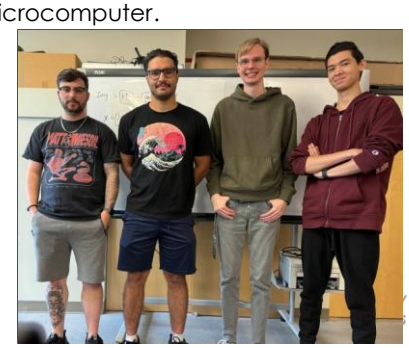
Skeleton-based Re-identification System

Funding Agency: *ODU ECE Department*

Development and Deployment of a Lightweight Skeleton-Based Person Re-Identification System for Real-Time Security Applications onto a Raspberry Pi Microcomputer.

Team Members: *Aubrey McKinney, Antonio Mendoza, Noor Humphreys, Seth Myers*

Advisor: *Dr. Khan Iftekharuddin*



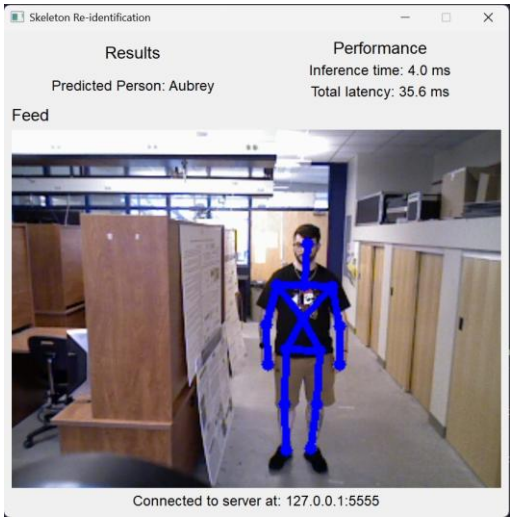
Design Challenge

Run effective skeleton re-identification on a low-power microcomputer that is easily scalable and cost-effective.

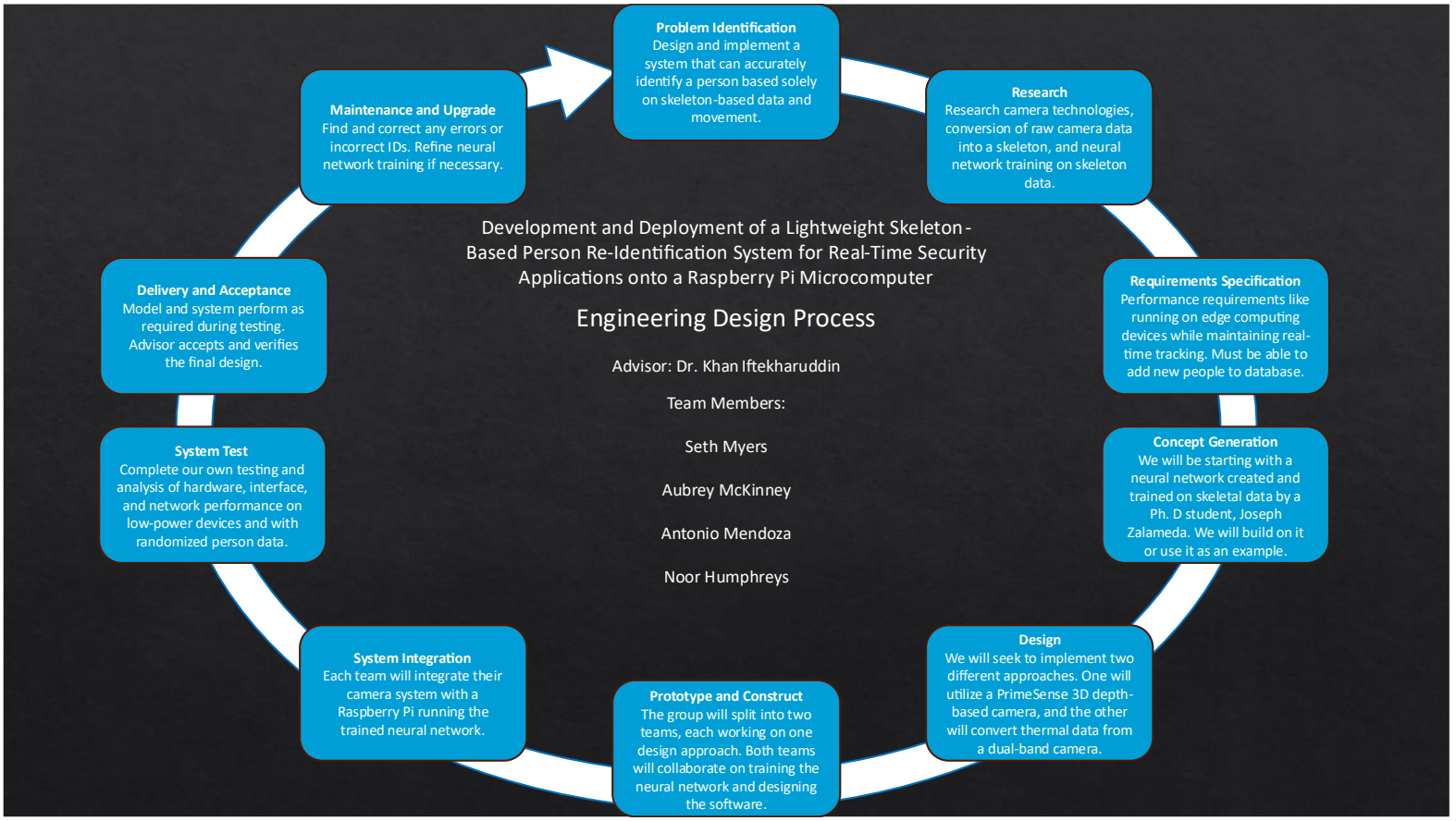
Design Goals

- Implement skeleton tracking with PrimeSense and FLIR thermal camera approaches
- Train a neural network to learn from data samples
- Create a user interface to display skeleton visualization and re-identification results

"Parallel processing machine learning and human intelligence"
-Antonio Mendoza



Appendix C: Original Design Flow Chart



Appendix D: ESPEX Poster

SKELETON RE-IDENTIFICATION ON RASPBERRY PI

Antonio Mendoza, Aubrey McKinney, Noor Humphreys, Seth Myers
Department of Electrical and Computer Engineering, Old Dominion University, Norfolk, VA USA

ODU
Batten College of Engineering and Technology

PURPOSE

- The purpose of this project was to develop an accurate, lightweight, and cost-effective skeleton re-identification system that runs on a low-power microcomputer.

BACKGROUND

- Skeletal Re-identification works similarly to other forms of biometric data, like fingerprints or facial scans.
- Skeleton Extraction: detects joints and position of a subject in the frame
- Convolutional Neural Network (CNN): A deep learning model that classifies feature patterns like distances between joints or pose variations

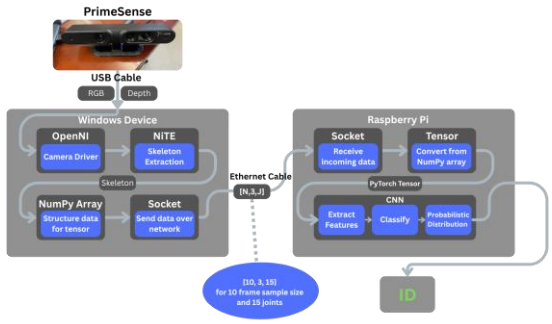
METHODS

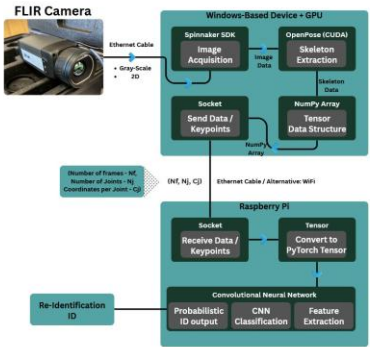
- **Two Approaches:** PrimeSense RGB-D (3D) or FLIR thermal (2D) cameras provide positional data to a Windows client machine
- **Extraction:** The cameras' respective software extracts skeleton joint coordinates from the frames
- **Preprocessing:** The extracted joints are normalized, batched into a NumPy array, and sent to a Raspberry Pi-based server
- **Identification:** The server hosts a convolutional neural network that analyzes features like the unique distances between joints and matches them to existing data
- **User Interface:** A GUI displays information like a camera feed, identification result, performance statistics, and connection status

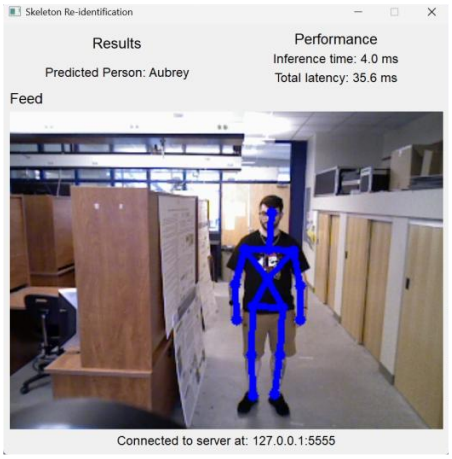
ACKNOWLEDGEMENTS

We would like to thank our advisor, Dr. Khan Iftekharuddin, and our project mentor, Joseph Zalameda, for providing design guidance. In addition, we would like to thank the ECE department and the ODU Vision Lab for providing the project hardware and funding.

DATA PIPELINE AND DESIGN







TESTING AND RESULTS

We have three stages of model testing – during training, post-training, and pipeline testing. We measure accuracy by the number of correct guesses the model makes for each set.

PrimeSense:

- **Training Set Accuracy:** 95-100%
- **Post-training Test Set Accuracy:** 65-70%
- **Full Pipeline Test (see left example):** 75% (3 out of 4 people)

FLIR:

- **Training Set Accuracy:** 90%
- **Post-training Test Set Accuracy:** 50-55%
- **Full Pipeline Test:** 25% (1 out of 4 people)

FUTURE WORK

Given more time, we would have liked to:

- Meet our original accuracy goal of >80%
- Test alternative designs with different cameras, a different model, or a custom skeleton tracking solution
- Implement more security features, like encrypted data transfers between the Pi and client machines

OLD DOMINION UNIVERSITY