COLORADOSCHOOLOFMINES
EARTH • ENERGY • ENVIRONMENT

CSCI 370 Final Report

Strike a Pose

Gordon Dina
Carissa Lee
Akshitha Mudupu
Moozoo Vang
Nico Zucca

Revised December 10, 2023

CSCI 370 Fall 2023

Dr. Iris Bahar

Table 1: Revision history

Revision	Date	Comments
Requirements Document	8/31/2023	Added the high-level description, functional and non-functional requirements, project risks, and definition of done
Design Document	9/17/2023	Added system architecture
Software Test and Quality, Ethical Considerations Document	10/22/2023	Added software quality plan, software tests, and ethical considerations
Results Document	11/12/2023	Added results
Final Report	12/5/2023	Added future work, lessons learned, acknowledgements, and team profile

I. Introduction

Venvee pioneers human-centric analytics, fueled by an unwavering commitment to innovation and a profound understanding of the human experience. Grounded in cutting-edge Computer Vision and Artificial Intelligence, their mission transcends conventional limits to decipher human behavior within spaces. Focusing their expertise on retail media networks, they drive transformative impact in an industry undergoing explosive growth. With advanced technologies, these networks gain unparalleled insights into in-store shopper behavior, revolutionizing audience metrics with accurate data on impressions, actions, and conversions, while tapping into previously untapped intent data. The technology serves as a centralized hub for real-time client data, empowering retailers to drive immediate improvements, facilitate seamless in-store-to-online experiences, and optimize technologies like robotics and inventory AI—all without additional infrastructure. Venvee stands as the bedrock of the physical-to-digital retail transformation, heralding the era of truly prescriptive smart stores. Based in Golden, Colorado, Venvee boasts a team of accomplished experts and strategic affiliations with the Colorado School of Mines, supported by industry-leading partners and a diverse advisory board versed in business development, finance, and innovation.

II. High Level Description

The Strike a Pose field session project includes two efforts: to deliver a fully trained pose classification model that introduces a temporal aspect, and to produce a store action dataset that Venvee can train future pose classification models with. To deliver the pose classification model, the team used datasets provided by the client to train the model to take spatial and temporal information and output what actions a person is taking. To produce the store action dataset, the team utilized synthetic data generation methods and tailored the data to Venvee's requirements.

III. Functional Requirements

The functional requirements outlined by Venvee are included below.

- The model must be able to identify physical actions a person could take, such as walking or standing
- The model must differentiate between a store product and a non-store product
- The model must return outputs that are granular, time-based, and highly descriptive
- The dataset must contain the keypoints along with the images and synthetic footage used to generate the data
- The dataset must include keypoints in both COCO and CrowdPose format
- The dataset must include common actions that a person does within a retail space
- The dataset must include poses from several angles

IV. Non-Functional Requirements

The non-functional requirements outlined by Venvee are included below.

- All code, including the pose classification model and the generated dataset, must be pushed to the provided GitLab repository
- All research and work must be thoroughly documented
- The model must be available for commercial use
- The model must be able to incorporate seamlessly into the Venvee pipeline

V. Risks

The risks outlined by the team are included below.

- The team will be reliant on the subject matter experts at Venvee to provide machine learning knowledge and resources
- The team will be reliant on the Venvee employees to provide data and access to resources in a timely manner

VI. Definition of Done

The pose classification model will be considered done when the team delivers a product that meets the following criteria. The model should match the required formats for Venvee's Edge pipeline, in order to seamlessly place into the overall pipeline with little to no edit. The model should be able to identify different in-store actions that a person could take, such as walking or standing, with a specific focus on product interaction. The product interaction class will ideally be split into more granular classes, such as reaching out, holding, or putting back. Additionally, the model should be fortified against identifying non-product interactions, such as a person pushing a cart. Preferably, the model will have a high FPS without sacrificing accuracy. Finally, along with the model, the team will deliver clear and understandable documentation on the model, its specifications, and its implementation for later use.

The dataset will be considered done when the team delivers a product that meets the following criteria. The dataset should include the images and synthetic footage used to generate the pose data, pose data itself (keypoints for a person), and the annotations (action/class). The keypoints format must be in COCO format and CrowdPose format. The actions within the dataset must include common actions within a retail space, including shelf interaction, basket and cart interaction, checkout interaction, walking, standing, and phone interaction. The actions must include a variety of angles, and be as granular as possible to improve the quality of the classification.

VII. System Architecture

The following flowcharts were generated using Mermaid Chart (https://mermaid.live) and are included to display the internal and external components of the system. Figure 1 shows the data input and processing, which includes pose data being read into the system, processed, and output with the updated time values.

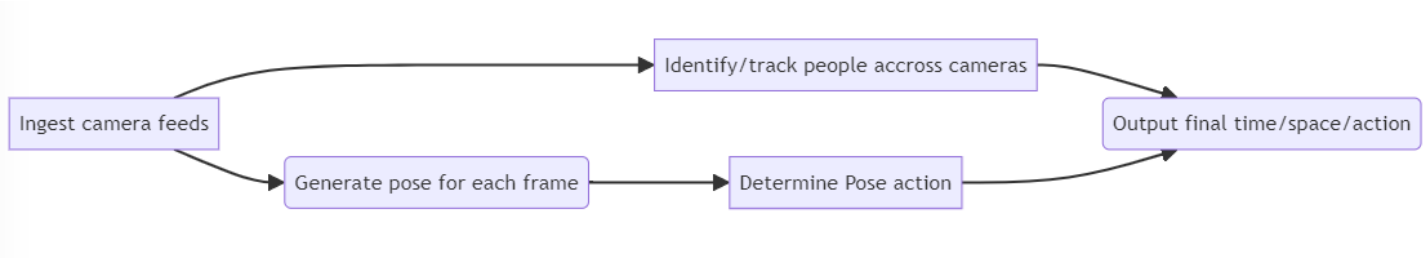

Figure 1: Data Input and Processing

Figure 2 represents the inputs and outputs to the model, as well as specific actions the machine learning algorithm will implement.

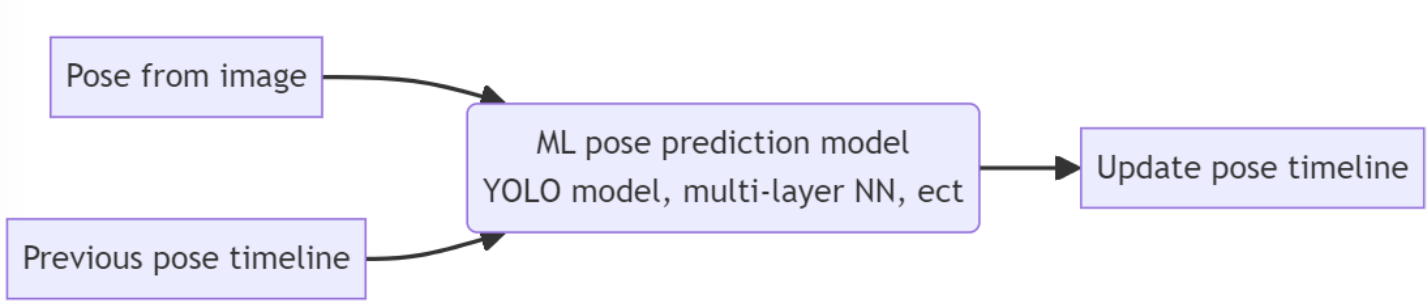

Figure 2: Model Inputs and Outputs

Lastly, Figure 3 represents the full system architecture, including the raw data input from camera feeds, specific actions from the machine learning model, and time-series pose classification data output.

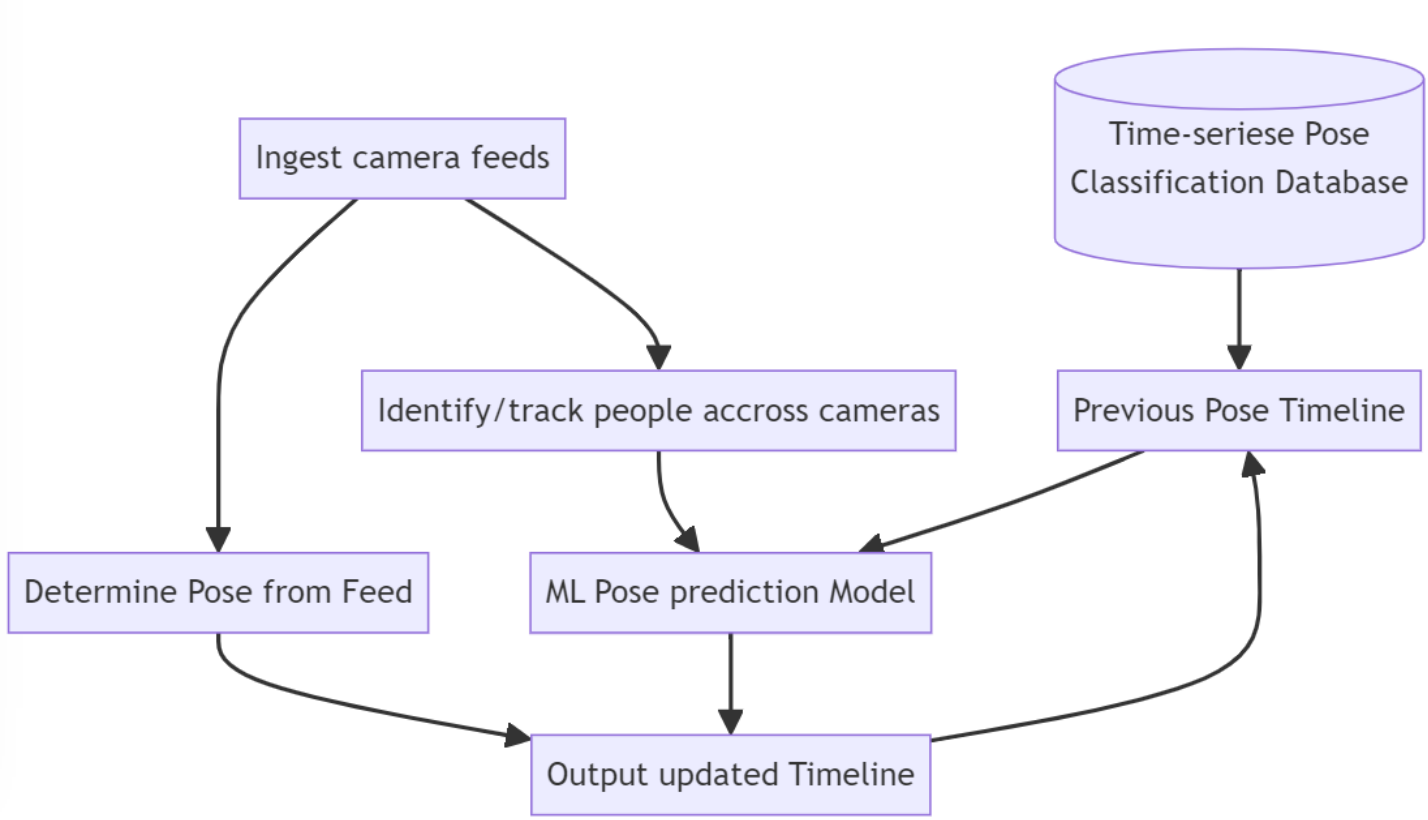

Figure 3: Full System Architecture

VIII. Technical Design

The first model analyzed was the K-Nearest Neighbors model. KNN is based on the concept of similarity, and uses groups or clusters of data to classify points nearest to each other. It is a form of supervised machine learning, and can process different formats of data as input to output cluster predictions. The model takes in a dataset and a cluster value, k, to find k nearest neighbors to each point based on how far apart the next point is.

The team utilized a simple data processing and KNN implementation to start with. Since KNN is a fairly simple model, heuristics are commonly used to tailor the model to the specific needs. After researching various heuristics, an angle heuristic was decided upon. The goal of the angle heuristic was to calculate points based on their angles and feed that into the KNN processor, rather than the given data points themselves. The following figure shows the code for the angle heuristic.

```python
def calculate_angle(point1, point2, point3):
    vector1 = (point1[0] - point2[0], point1[1] - point2[1])
    vector2 = (point3[0] - point2[0], point3[1] - point2[1])

    dot_product = vector1[0] * vector2[0] + vector1[1] * vector2[1]
    magnitude1 = math.sqrt(vector1[0] ** 2 + vector1[1] ** 2)
    magnitude2 = math.sqrt(vector2[0] ** 2 + vector2[1] ** 2)

    cosine_theta = dot_product / (magnitude1 * magnitude2)
    angle_in_radians = math.acos(cosine_theta)
    angle_in_degrees = math.degrees(angle_in_radians)

    return angle_in_degrees


def classify_angle(frame_object, point_list, angle_list):
    for point in point_list:  # for each set of points from the given list (can modify)
        angle_in_degrees = calculate_angle(frame_object[point[0]], frame_object[point[1]], frame_object[point[2]])
        angle_list.append(angle_in_degrees)

    return angle_list
```

Figure 4: Angle heuristic implementation

While the team was able to implement the angle heuristic successfully, its performance was not able to be fully tested due to the lack of data. However, the team decided not to move forward with the KNN model based on the tests performed, which are discussed in the results section of this report.

After looking at the performance of the KNN model, a second model was created, which was a neural network model. Neural networks are graphs made up of neurons, where the data is being passed from one neuron to another. The type of neural network that was programmed was a feedforward neural network. This type of model passes data through multiple nodes until it reaches the output node. Data only moves in one direction and is classified by different activation functions. These activation functions determine if a neuron should be activated by comparing the input value to the threshold value.

Before creating the various layers of the model, the team started with preprocessing the data. The dataset was given in a json file. The figure below was the code used to preprocess the dataset.

```python
# Load the json file
dataframe = pd.read_json(json_path)
df_to_process = dataframe.copy()

# Extract the labels
y = df_to_process.pop('action')-2

# Extract the list of class names
classes = y.unique()

# Convert the remaining data into a bunch of columns
df_to_process = pd.DataFrame(df_to_process["pose25d"].tolist())


# Convert the input features and labels into the correct format for training.
X = df_to_process.astype('float64')
y = keras.utils.to_categorical(y)
```

Figure 5: Preprocessing the dataset to a json file with landmarks and action numbers

The json files created extracted the array called “pose25d” as well as the corresponding action. The numbers in the array represented landmarks of the poses. The action given was a number between 1-8.

The neural network model used Keras, which was used to create the different layers in the model. This model uses 5 layers consisting of 2 dropout layers. The figure below shows the code to generate those layers.

```
inputs = tf.keras.Input(shape=(51))
embedding = landmarks_to_embedding(inputs)

layer = keras.layers.Dense(128, activation=tf.nn.relu6)(embedding)
layer = keras.layers.Dropout(0.5)(layer)
layer = keras.layers.Dense(64, activation=tf.nn.relu6)(layer)
layer = keras.layers.Dropout(0.5)(layer)
outputs = keras.layers.Dense(len(class_names), activation="softmax")(layer)
```

Figure 6: Layers used to create the neural network model with inputs, and different activation functions for the layers

The inputs variable is taking in data of a one-dimensional array with 51 elements. The two dropout layers are applied to prevent overfitting. Using these layers, the model outputted the accuracy and a confusion matrix with the dataset used.

IX. Software Test and Quality

As software developers, an important practice to keep in mind early and often is software quality. Software quality is the assurance that the code being developed is correct, thoroughly tested, and sufficient to meet the customer's requirements. This process includes gathering data, forming requirements, and designing tests, all before any actual coding is completed. Within Agile, there is a large emphasis on implementing quality as early as possible. This means that the design must be frequently checked against its requirements and verified with the customer's expectations. The Strike a Pose project aligns with the high standards of software quality, as the implementation needed to be constantly checked by the customer in order to be seamlessly incorporated into the Venvee pipeline. The team chose to implement the following practices to verify the software quality.

1. Pair Programming - this quality assurance concept is used to confirm that all the logic behind the code makes complete sense. With pair programming, the team hopes to improve the defect detection aspect of code quality. Having numerous people work on the same bit of code allows for the mitigation of any defects discovered and the improvement of coding standards. Pair programming is also used to ensure the cleanliness of code so that when the team provides the clients with the finished product, they too will be able to understand everything that is going on.

2. Code Reviews - meetings are scheduled between the team and the client so that the team can demonstrate the current product to them. The aspects of code quality the team improves with this test are validation and verification. Code reviews allow for confirmation that the code follows proper guidelines that were set by the client. With these code reviews, the team is able to receive feedback on ways to further improve or alter the code.

3. Integration testing - part of the unique challenge to the project is the requirement to integrate seamlessly with existing pipelines. As a result, it is crucial for the team to have a solid plan for integration testing. The team has been able to modify the method of developing code to assist in this goal. By using accurate data that is representative of the inputs the pipelined application will receive, the team can effectively test pipelining while working! Of course, the team still needs to conduct additional testing with how the process outputs its results and ensure smooth function. This is planned after the first functional prototype is developed.

4. User acceptance testing - due to the nature of the project being internal to an existing pipeline, there is no need to test with a layman or ensure that the handles and nobs of our model are easily understandable by someone

without a background in machine learning. Instead, the user acceptance testing the team needs to conduct ties heavily into integration testing and code metrics to ensure that the client can use the final product.

5. Code metrics - code metrics are the single most principal factor driving overall development. The performance of the application is a large aspect of its usefulness to the client, and thus the relevant metrics (accuracy, precision, processing time, load time, memory usage, etc.) must be monitored to ensure that when the product is delivered, the team understands how it will run on the edge hardware provided.

X. Project Ethical Considerations

The Strike a Pose project brings many ethical challenges into consideration. Several of the issues the team outlined are listed below.

- Use of Artificial Intelligence and machine learning - it is difficult to implement programs using these technologies because it cannot account for things that a human would, and the team's goal to teach a model to classify poses can be complex
- Consent – collecting store data can be tricky, as consent cannot be granted from each customer
- Training data – the team will need to keep natural biases in minds while working with the datasets
- Future uses – there is no way to guarantee how the work the team produces will be used in the future
- Disability support – the algorithm currently does not account for customers with disabilities as more data is needed to account for that

Additionally, the team identified several professional ethics principles pertinent to the project that are listed below.

- Association for Computing Machinery (ACM):
    - Principles most pertinent:
        - 2.1 - strive to achieve high quality in both the processes and products of professional work
        - 2.2 - maintain high standards of professional competence, conduct, and ethical practice
        - 2.6 - perform work only in areas of competence
    - Principles most violated:
        - 1.4 - be fair and take action not to discriminate

- Institute for Electrical and Electronics Engineers (IEEE):
    - Principles most pertinent:
        - 2.01 - provide service in areas of competence, being honest and forthright about any limitations of their experience and education
        - 2.02 - ensure that the product delivered meets the licensing requirements
        - 3.02 - ensure proper and achievable goals and objectives for any project
        - 3.03 - identify, define and address ethical, economic, cultural, legal and environmental issues related to work projects
        - 3.10 - ensure adequate testing, debugging, and review of software and related documents
        - 6.08 - take responsibility for detecting, correcting, and reporting errors in software and associated documents
        - 7.06 - assist colleagues in being fully aware of current standard work practices including policies and procedures for protecting passwords, files and other confidential information, and general security measures
    - Principles most violated:

- 1.03 - approve software only with a well-founded belief that it is safe, meets specifications, passes appropriate tests, and does not diminish quality of life, diminish privacy or harm the environment; the ultimate effect of the work should be to the public good
- 1.07 - inclusion of people with disabilities
- 3.13 - be careful to use only accurate data derived by ethical and lawful means, and use it only in ways properly authorized

The Michael Davis Tests were developed to test ethical principles related to a product. The team identified several tests pertinent to the project below.

1. Legality Test - the product doesn't violate the law or policy unless the videos of the customers are taken without permission. Most retail stores already have cameras throughout the store/building for security purposes. Walking into a store means that you give consent to being recorded. The product does not require any personal information of an individual and are simply classifying the different poses a customer has in a retail store. The data that is collected in store is processed by our product, and we return that processed data to the customer (the store--like Kroger, etc.) to use as they please, likely for analysis of their ad campaigns. However, the team must consider each customer's privacy in the act of taking a video.
2. Harm Test - the harm of this product is the way the data is being obtained. It is not ethical to take videos of customers that are not for store use without getting the customer's permission. However, that is not feasible. The benefit of this product is that it can help retail stores improve their overall business by seeing what products customers are buying. Overall, this product will not harm individuals, but will be helpful for the pose classification model which will then benefit retail businesses.

XI. Project Completion Status

The goal of the project was to deliver a pose classification model and a dataset that future models can be trained on. The following sections define the team's project completion status.

XI.I Model Results

- Features implemented:
  - The team has managed to implement multiple pose classification algorithms, as well as classify their performance. The team used these algorithms to evaluate Venvee's internal dataset, and provide real steps that the company can take to improve their output moving forward.
  - The team conducted extensive research on existing pose classification datasets, which Venvee can use to augment their proprietary data. In addition, the team discovered and tested synthetic data generation using text-to-motion models to further augment Venvee's dataset. The team anticipates that this will allow future work to successfully integrate their model or another model based on the new data.
  - Unfortunately, using the initial company provided dataset, the team was not able to generalize the results to untrained data. This was due to the data skew and quantity, both of which can be fixed with the synthetic data generation methods the team developed. As a result, the team did not complete the planned pipeline integration, and instead had to leave that work well documented for the next team or for a Venvee internal team.

- Performance testing results:
  - The team implemented a neural network model using the data given, which were formatted into csv files. The results achieved are shown below:

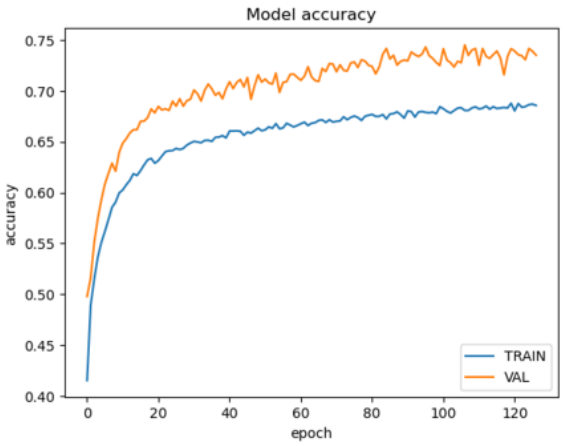

Figure 7: Model accuracy with validation set

- This is a graph of model accuracy over time. Validation is higher than training because dropout layers are used while training, but not on validation. This is a pretty good accuracy, which can also be seen in the confusion matrix.

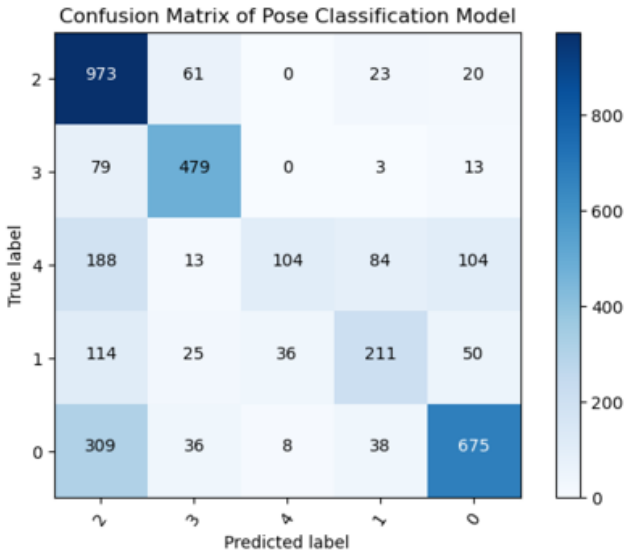

Figure 8: Confusion matrix with validation set

- The confusion matrix shows correct answers on the diagonal, and everything else is incorrect. As the figure demonstrates, the model is a bit messy, but generally has a diagonal trend.

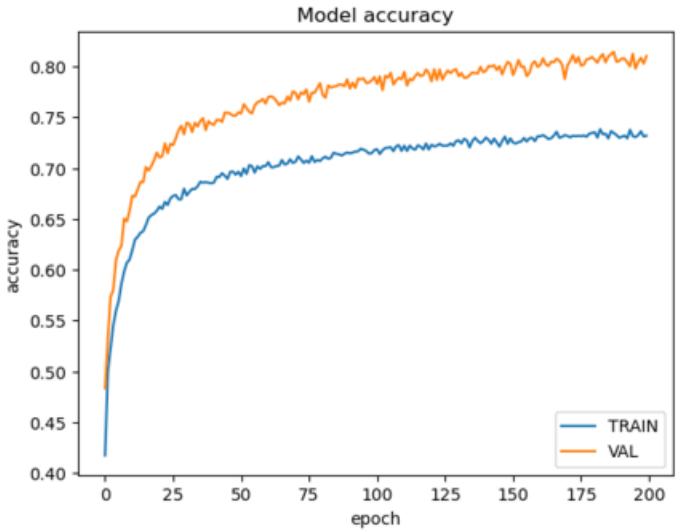

Figure 9: Test dataset without poses from camera 7

- These next two fixgures are the same as above, except that the validation set is now an entire camera dataset instead of randomly pulled from multiple cameras.

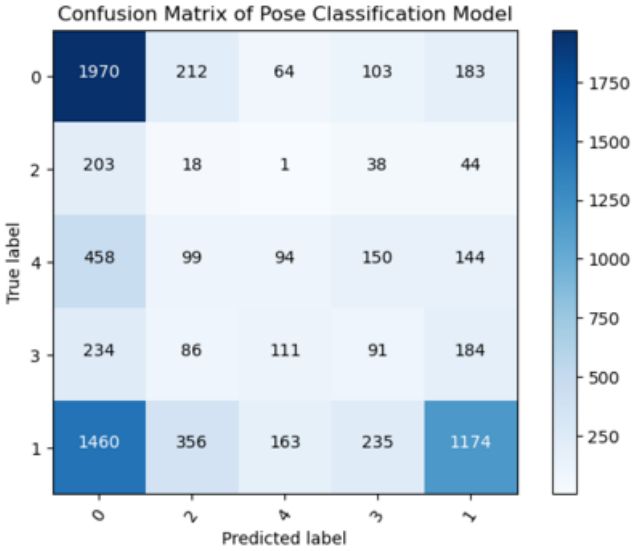

Figure 10: Confusion matrix from testing dataset

- This causes the confusion matrix to be much worse, for reasons already explained. It is clearly seen that the model is overpredicting for class 0, which shows that it is not generalizing poses very well.

Class	Precision	Recall	F1 Score	Support
0	0.46	0.78	0.57	2532
1	0.68	0.35	0.46	3388
2	0.02	0.06	0.03	304
3	0.15	0.13	0.14	706
4	0.22	0.10	0.14	945

Figure 11: Summary of classification results

- The team achieved an accuracy of approximately 43%, which is on the low side. From analyzing these outputs, the team concluded that the validation set performed better with a higher accuracy than using the split dataset. One reason the accuracy may have scored this low is because camera 7 contained more actions, so there were not enough poses left for the model to train on. For example, camera 7 had more of pose 2, which was grabbing from shelf.

- User testing results
  - One of the primary methods of testing were client demos. The weekly client meetings always consisted of a summary of work completed and work planned for the next week. The demos that were performed during the weekly meetings were received well, and the client offered feedback in order to improve the team's implementation. As the client acted as the main user of the software, no other user testing was performed.

XI.II Dataset Results

Based on the results gathered, the team decided to generate synthetic data as a way to augment the performance. The team made two attempts, one to address the pose discrepancy, and one to address the camera discrepancy. The pose discrepancy could only be addressed by generating poses that could be used to train. The team was able to get a text-to-motion model working, which generated sequences of poses that describes an action. For example, the text "A woman picking up milk from the shelf" could be inputted, and the model would output a video as can be seen below.

Figure 12: Output of Text-To-Motion

This gives the ability to generate as many movements as it can generate text prompts, which is a huge improvement on manual pose generation. Additionally, many other biases can be corrected in the data by generating prompts that address discrimination. The main problem with the model was that its outputs did not look very human-like. The output human will often stand on 1 leg when bending over or freeze randomly. Due to time constraints, there was no chance to test the output with the model, but the training would not generalize to humans.

The reason testing was not completed was due to the speed of generation. It was very slow (~15 mins) to generate a single inference, which is unacceptable to generate the thousands of inferences needed. If someone were to continue work using this model, it would need to run on a very powerful server.

The team also worked on camera generalization, which is a much simpler problem. A program was written which can take a 3D pose and output millions (as many as you want) of 2D projections of the given pose. This allows a single stereo-camera setup to capture enough data that so a model can be built that generalizes to any camera position and angle. The code has an interface as shown below.

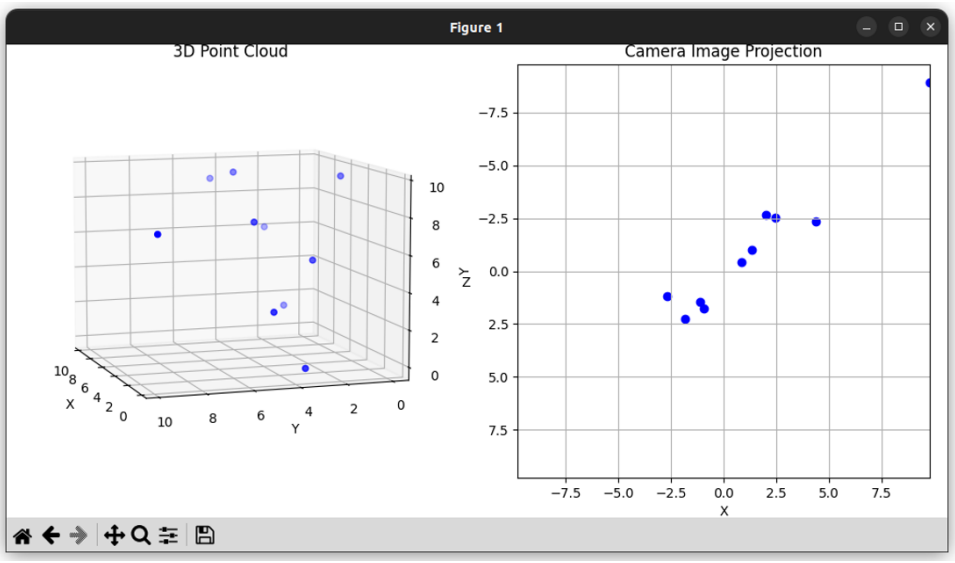

Figure 13: Output of Synthetic Camera Generalization

XII. Future Work

1. **Post-training Quantization/Quantization Aware Training:** When switching to a neural network-based approach for classification, quantization was highlighted as important as this would significantly decrease the computation time for real-time inference. However, quantization can reduce model accuracy, depending on the conversion. Post-training quantization was implemented into our program at the end, but Quantization aware training would have given us a higher accuracy return compared to the former; research was put into Quantization aware training but wasn't fully completed.
2. **Planned Pipeline Integration:** Given the situation of the initial dataset, the team was not able to fully integrate the pose classification program with Venee's pipeline in the given time. Once a new dataset is given, development in pipeline integration will continue.
3. **Increase Accuracy:** Current accuracy score from the initial data set was not ideal. A new dataset made from synthetic data and further testing would provide an improvement in accuracy.
4. **Dealing with Counter Examples:** Our pose classifier program was able to identify a given pose, but the team did not have the opportunity to test the program against counter examples, such as someone looking at their phone rather than a product. The use of synthetic data would have provided this opportunity.
5. **Available/Have Licensing for Commercial Use:** Licensing of our program for commercial use was one of our main requirements, but since pipeline integration was incomplete, this was left unfinished. Continuing development in pipeline integration would kickstart licensing back up.

XIII. Lessons Learned

In our efforts to create a Machine Learning algorithm that did pose classification, the team ran into multiple issues that resulted in the following lessons:

1. **Data Quality Matters:** At the beginning of the semester, the client provided data before the team started any of the actual coding. As a result, the team did not fully understand the data requirements, and began heavily coding before realizing the provided data was not sufficient. By the time more data was obtained, the team ran out of time.
2. **Effective communication with our client:** Early on in the project, the team thoroughly discussed the project with the client. From there, the team assumed full control of the project with the assurance that the requirements were understood. However, the team soon realized there were still various important questions. This same process proceeded throughout the entirety of the semester. As a result, time was wasted from the slow process of back and forth communication with the client.
3. **Have standards:** Initially, the team accepted the client's offerings without seeking specific details, and this approach led to issues later on. The team learned to request information that aligns with project guidelines, providing a smoother development process.
4. **Document everything:** The Strike a Pose project revealed the importance of documentation. Venvee introduced the team to the Jira software, which helped the team to navigate the Agile process and keep track of the requirements and research necessary.

XIV. Acknowledgments

The Strike a Pose project would like to thank their advisor, Iris Bahar, for all her advice and support throughout the semester. Additionally, the team would like to thank Khloe Downie, for acting as a resource and providing the role of a client to the project.

XV. Team Profile

What I worked on:

Nerual Network code: *Worked on NN model*

Text to motion data set: *Experimented with synthetic text to motion data generation*

Quantization Research: *Reasearched Quantization to see it could improve our model*

Bio: Howdy my name is Gordon Dina. I'm a senior studying Computer Science specializing in Robotics & Intelligent Systems.

Role: Researcher, Software Developer

What I worked on:

KNN algorithm: *Created K-NN angle heuristic*

Nerual Network code: *Worked on NN model*

Text to motion data set: *Experimented with synthetic text to motion data generation*

Bio: Hello! I'm Carissa Lee, and I am a senior studying Computer Science with a business focus.

Role: Researcher, Software Developer, Client POC

What I worked on:

Nerual Networks Reaserch: *Reasearched NN algorithm*

Nerual Networks Code: *Found/altered pre-existing NN code that does pose prediction*

Synthetic text to motion data generation: *Experimented with synthetic text to motion data generation*

Bio: Hi! My name is Akshitha Mudupu and I am a senior studying Computer Science with a focus in business.

Role: Researcher, Software Developer

What I worked on:

Nerual Networks Reaserch: *Reasearched NN algorithm*

Nerual Networks Code: *Found/altered pre-existing NN code that does pose prediction*

Coco 17 and Crowd Pose data set rearch: *Looked for data sets in Coco 17 and Crowd pose format that had grocery store poses*

Bio: Hello, my name is Moozoo Vang, and I am a senior studying Computer Science with a specialization in Computer Engineering.

Role: Researcher, Software Developer, Advisor POC

.

What I worked on:

KNN algorithm: *Created K-NN pose prediction model*

Nerual Network code: *Worked on NN model*

Text to motion data set: *Experimented with synthetic text to motion data generation*

Synthetic 2D Pose generatation: *Experimented with 2D synthetic pose generation model*

Bio: My name is Nico Zucca. I am a senior in Computer Science with a focus in robotics. I am currently a perception engineer at Trimble and a researcher at Aria Lab.

Role: Researcher, Software Developer

References

- *Online FlowChart & Diagrams Editor - Mermaid Live Editor*. (n.d.). https://mermaid.live/edit#pako:eNptkLsOwjAMRX_FeKX9gQ4gJAZAYmLNYiUXGikPCIkQqvrvpC1seLqyz7FkD6yjAXf8xKMgaOyt3JJ4FajWzlmNdrNZn2IfOjrAuUhTbqiPL5IEesey_QsvmJYwIdRDEnl80WnWVrSdnbrYLnZDx9mY6aqt_uNHugKObgmSV9ywR_JiTb1imATFuYeH4q5Gg6sUlxWrMFZUSo6Xd9Dc5VTQcLkbyb-jf00Ym2M6L4-Z_zN-AJdmX0k

Appendix A – Key Terms

Term	Definition
ACM	*Association for Computing Machinery*
IEEE	*Institute for Electrical and Electronics Engineers*