**55-708252 - AI Research and Development Project**

**Fall Detection From Video Footage**

**Reece Wareham**

**30024660**

**Problem Description**

Fall detection is an important aspect of healthcare and artificial intelligence, particularly for those who are older or have mobility issues. Falls are one of the most common causes of injury and injury-related death among elderly adults and adults with mobility issues and are a major cause of hospitalisation and long-term care especially when not treated hastily as the longer it takes for treatment to begin, the higher likelihood the damage is permanent therefore, detecting a fall swiftly and accurately is critical in helping keep the quality of life of vulnerable people at a high standard.

A solution to help mitigate this problem is to implement a machine learning system that can detect if a person falls over using video footage to allow for a quicker response time to the fall, helping prevent permanent damage and even death. This system will also help provide important data on fall patterns and risk factors which can be used to help prevent future accidents and improve their quality of life. It will also be able to notify an emergency contact when the system detects that a person has fallen.

**Solution Implemented**

The solution that I’ve aimed to implement to solve the proposed problem for this project is to create a machine learning algorithm that has the capability to detect a person falling from video footage and notify an emergency contact of said fall. This solution is made up of many different parts from gathering a dataset to training a machine learning model and is very complex.

The first important aspect of the project was the dataset. Due to the limited amount of large and suitable datasets of people falling, a new dataset needed to be created before any other parts of the project could be completed. The dataset we created consists of various different videos of people falling in both real life and in-game footage from different camera angles and also different environments. This dataset is made up of 49 videos with an average length of 5 seconds each. This part of the project I worked on extensively as I gathered and annotated a large majority of the overall dataset which can be seen on the Trello board.

The second part was the annotation of the dataset. All of the data in the dataset needed to be annotated frame by frame with either fall or no fall for use in the machine learning algorithm. To complete this process, we used CVAT (Computer Vision Annotation Tool) which is a tool used to annotate each frame of a video. This process was done manually for all 49 videos of the dataset and was very time consuming. Once a video is annotated, the annotations are exported as an XML file which we load in python and extract the data that we need. During the project, I got very familiar with this process as I annotated a large majority of the videos (GTA V clips) that I gathered for the dataset.

The next part is the pose detection that is used to recognise a person in a video. The pose detection framework used in this project is Mediapipe developed by Google. This framework accurately recognises the poses of a person and extracts the coordinates of each individual body part landmark. These landmark coordinates are the main data that is used to determine whether or not a person has fallen or not. The landmarks that are deemed important for detecting a fall are the nose, shoulders, hips, knees, feet, wrists and elbows. Combining these landmarks with the annotations creates the complete dataset that is used for the project.

The final main part of the project is the machine learning algorithm. A support vector machine is the chosen algorithm for this project which has been trained on the dataset that has been created. This algorithm can successfully detect a fall with 93% accuracy. Expanding off this algorithm, we managed to implement this model in real time using camera systems to input a video stream into the model and make predictions in real time. A notification system is also implemented that notifies the user when a fall has taken place based on the model’s predictions. I was the main contributor to this process due to my past experience in this area and my familiarity with support vector machines.

**Justification**

For this project, a support vector machine was chosen as the machine learning algorithm of choice because of a few main reasons. The first reason is that an SVM is very simple and easy to implement. This is important because it means that most of our time can be dedicated to other areas of the project that may require more of a focus. Also due to the nature of the data we were using in this project, using more complicated algorithms such as neural networks is unnecessary and will only require more time and resources to perfect for little performance gain. Also due to the data only being numbers, it removes some algorithms out of the decision process such as convolutional neural networks which would not be useful in our project as they mainly focus on analysing video frames. Last but not least, the goal of the project is to create a classifier that can predict falls therefore, a classification algorithm needs to be used which makes a SVM very suitable for this task.

Also for this project, the framework that was chosen to detect the poses of people was Mediapipe. This is because Mediapipe is also very easy to implement alongside the SVM making it a suitable option compared to some other pose detection methods such as OpenPose. Mediapipe is also very accurate and shows similar results to more complex methods like OpenPose. It is also optimised very well for on-device performance making it able to run on many devices and has more extensive platform support unlike OpenPose. Lastly, Mediapipe is very configurable which allowed us to easily toggle the parts that we needed for example, which landmarks we wanted to detect. An example of the differences between OpenPose and MediaPipe can be seen in the conference paper [**Nguyen et al (2022)**](#_nyzyvpt9t1qv) . In this paper, the decision in the project was made to use MediaPipe due to the previous reasons and due to that project being similar to the fall detection project in terms of pose detection, the final decision was made to use MediaPipe.

**Contribution Log**

| Name of Task | Description of Task | Contribution Percentage | Team Contribution Percentage | Time Spent on Task | Outcome of Task |
| --- | --- | --- | --- | --- | --- |
| Research machine learning algorithms | To research machine learning algorithms and determine the best one to use. | 100% | 0% | 22/01/24 - 29/01/24 | Researched many different machine learning algorithms for this project. Determined the best algorithm to use for this project to be a support vector machine (SVM) due to its simplicity and usefulness with the type of data we will be using. Other algorithms such as neural networks are not viable and unnecessary for a project of this size. |
| Research existing fall detection datasets | To research existing fall detection datasets and decide if they are useful for the project. | 50% | 50% | 05/02/24 - 12/02/24 | Researched some existing datasets that will be useful in creating a system to detect falls. However, these datasets are synthetic and don't contain natural falls. They are also very small in size. I learned how to find datasets relevant to the project. |
| Research pose detection libraries and frameworks | To research pose detection libraries and frameworks to display a skeleton model around a person in a video. | 50% | 50% | 12/02/24 - 19/02/24 | Found some existing libraries and frameworks such as Mediapipe, OpenPose, YoloV7 and DeepPose. |
| Annotate the datasets from research | To annotate the videos from the datasets we gathered online. | 33% | 66% | 19/02/24 - 26/02/24 | Successfully annotated my portion of the datasets gathered using CVAT. Annotated all frames with either fall or no fall. I learned how to annotate and label video footage. |
| Write the code for importing our dataset into python and for creating and training a SVM. | To write the code used to import our dataset and for code for creating and training a support vector machine. | 100% | 0% | 26/02/24 - 04/03/24 | Successfully created a Jupyter Notebook that imports the dataset for training and testing as well as creating a SVM trained on 80% of the initial dataset. Outcome is a SVM with 65% accuracy at making fall predictions. This is impressive for the small amount of data. |
| Create powerpoint demonstration ready for progress presentation | To create a powerpoint explaining the project in its current state and what we will do in the future. | 33% | 66% | 04/03/24 - 11/03/24 | Created a powerpoint presentation in collaboration with the others in the group. Contained introduction, our progress so far, what techniques we are using and what we are going to do in future. |
| Research different methods of getting synthetic data. | To find new ways of making synthetic data to expand the dataset. | 50% | 50% | 11/03/24 - 18/03/24 | Found the best way to get synthetic data that is similar to real life data is using GTA V to make clips of characters falling using a ragdoll mod. |
| Gather in-game footage of people falling | To gather in-game footage of people falling for use in the dataset. Main source will be GTA V. | 100% | 0% | 18/03/24 - 25/03/24 | Gathered a total of 33 video clips of people falling from GTA V. This was done using a mod to help ragdoll people in-game to get realistic falls. Learned how to mod GTA V for the first time. |
| Annotate the in-game footage | To annotate all the in-game video footage for use in the main dataset. | 100% | 0% | 25/03/24 - 01/04/24 | Annotated all the videos gathered from GTA V. All videos were annotated using CVAT. These annotations were also combined with the pose landmarks to create the final dataset. |
| Train a new SVM on the expanded and final dataset. | To train a new support vector machine on the final dataset. | 100% | 0% | 01/04/24 - 08/04/24 | Trained a final SVM on the final dataset with an accuracy of 93%. This is much better in comparison to the original which had an accuracy of 65%. |
| Create a GUI for testing the system. | To create a simple user interface that can be used to test the full system. | 100% | 0% | 08/04/24 - 15/04/24 | Created a user interface using TKinter that shows the video being tested, the prediction made by the model and the coordinates of each landmark detected. Learned how to create a GUI in Python TKinter for the first time. |
| Create powerpoint demonstration ready for final presentation | To create a powerpoint explaining the project, our progress, future work etc. | 33% | 66% | 15/04/24 - 22/04/24 | Created final powerpoint presentation in collaboration with the others in the group. Contained introduction, proposal vs prototype, future work and videos of our prototype in action. |

**Personal Reflection**

To reflect on this project, overall the project was very successful as a lot was accomplished throughout the duration of the project. Looking back at the established aims from the project proposal, there were four main goals that we needed to accomplish for the project to be a success.

The first goal was to gather enough video footage for a dataset. This goal was one of the most important goals as without a dataset, there would be no possibility of training a machine learning algorithm and fortunately we managed to create a medium sized dataset made up of 49 total videos. This was enough data for us to achieve an accuracy of 93% making it a success.

The second aim was to properly label and annotate the data we gathered in preparation for machine learning algorithms. To begin, I was very unsure of how to annotate data as I had no previous experience in video annotation. However, after watching some tutorials and experimenting with CVAT, I finally was able to annotate a large majority of the dataset that was used in this project. Over half of the dataset is made up of GTA V clips which were all annotated by me. This goal was also successful as all the data that was gathered as part of the dataset were annotated and labelled with a large majority being done by myself.

The third main goal was to evaluate and compare machine learning algorithms. Throughout the project duration, we tried many different machine learning algorithms to decide which one would be most suitable for the project and came to the conclusion that a support vector machine would be the best fit. This decision was a success as after training the SVM, we were able to get an accuracy of 93% and were able to predict if a person is falling in real time using this model.

The final aim for this project was to implement the machine learning model to detect falls in real time. We successfully achieved this by connecting phone cameras to our Python code using DroidCam and feeding each frame of the video stream into the model. This was able to make predictions in real time with a high success rate. We also implemented a notification system alongside it when a fall is detected. This made our project overall a success as we were able to accomplish the main tasks of the project.

Throughout this project, I participated in nearly every task that was assigned. I mainly worked on the research at the beginning of the project, the dataset creation and annotation as I gathered a large majority of the videos present in the final dataset and I mainly worked on the creation, training and testing of the support vector machine used to predict falls. My work was proven successful as the final accuracy of the model was 93% which led to the success of the project. This can all be seen in the Trello board as I was assigned to all of these tasks and also was the creator of the files seen in the Google Drive.

The key reason for the success of this project is the communication within the group. Communication was very important in this project as it allowed us to know what each other is working on and the progress of each goal. Throughout the duration of the project, we used a mixture of Discord and WhatsApp to communicate with each other with constant messages back and forth about what we are working on and what needs to be done by each deadline. We had two weekly meetings, one scrum meeting and one collaborative meeting where we worked together. This led to us being up to date on all the goals of our project and meant that everything was done before each deadline.

From this project, the experience I have taken away from it is that communication is key when it comes to working on a project in a group and having good communication allows you to collaborate really well with your team and make massive progress very quickly. Also it is useful knowing what everyone is working on and what needs working on next through communication and also through the use of Trello boards. I have also learnt how to annotate and label video footage for future projects that may require it as well as learnt how to implement pose detection in Python through various different libraries and frameworks.

Some ethical implications that we had to be mindful of during the development of our prototype was the type of data we used. Due to the purpose of the project being able to detect when elderly people or people with mobility issues are falling, the type of data we should have been using is data that matches these situations. However, due to the difficulty of accessing this type of data from nursing homes or hospitals for example, we were unable to use this data and had to use generic fall videos instead. The main reason for this is privacy concerns with using footage of people without their permission. This is also something that needs to be taken into consideration when working on it in future as finding footage of different types of falls will be difficult to find usable online and may require the use of private footage.

**Peer Contribution Report**

The peer contribution report that can be seen in [**Annex A**](#_uagrm0i6b760) shows the percentage of each member who participated in the overall project. This report shows that the workload was distributed evenly as all members are at 33% participation. This means that we all contributed equally which can be seen in the Google Drive and Trello as we worked together on a lot of the tasks required to complete the project.

**References**

# Nguyen, P. K., Nguyen, A. T., Doan, T. B., Trung, P. N., & Thi, N. D. (2022, December). Assessing Bicep Curl Exercises by Human Pose Application: A Preliminary Study. In *International Conference on Soft Computing and Pattern Recognition* (pp. 581-589). Cham: Springer Nature Switzerland.

**Annexes**

# **Annex A**

**Peer Contribution Report Form**

# **Peer Contribution Form**

How equal (or not) was each team member's contribution, including yourself, in your opinion?

For example:

* A team where all members contributed equally, consisting of 3 people, each person would get the same mark, so each contribution is exactly 33%.
* A team where one member did most of the work, consisting of 3 people, so one person gets 70% and the other two get 15% each.

| **Group Name: Group 5 – The Fall Guys**  **Your name: Reece Wareham** | |
| --- | --- |
| **Names of ALL members of the group.** | **Contribution**  **(should add up to 100)** |
| 1. Reece Wareham | 33% |
| 1. Danylo Krywyj | 33% |
| 1. Fuhad Owolabi | 33% |
| **Total** | **100/100** |

Please discuss why you think this share of contribution is appropriate below. Don't forget to cover your own work (only briefly):

Throughout the project, all members of the group worked equally together on all the different parts of the project whether it be coding, documentation or research. We all collaboratively worked together on nearly all aspects of the project during weekly meetups to work together. I worked on lots of the research area such as researching pose detection libraries, existing datasets and machine learning algorithms. I also worked on the documentation such as the presentations and the scrum meeting records. Lastly, I worked on writing a lot of the code for this project such as importing our dataset, splitting the dataset into training and testing, training a SVM model, testing a SVM model and also created a GUI to help analyse the fall detection system.