2020

Machine Learning Assisted Gait Analysis for the Determination of Handedness in Able-bodied People

Hugh Gallagher

Technological University Dublin

Recommended Citation

Gallagher, H. *Machine learning assisted gait analysis for the determination of handedness in able-bodied people*. Masters Dissertation. Technological University Dublin. DOI:10.21427/zt7p-s632

Machine learning assisted gait analysis for the determination of handedness in able-bodied people.

Hugh Gallagher

D17126644

A dissertation submitted in partial fulfilment of the requirements of Technological University Dublin for the degree of M.Sc. in Computer Science Advanced Software Development

2019

I certify that this dissertation which I now submit for examination for the award of MSc in Computing Advanced Software Development, is entirely my own work and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the test of my work.

This dissertation was prepared according to the regulations for postgraduate study of the Technological University Dublin and has not been submitted in whole or part for an award in any other Institute or University.

The work reported on in this dissertation conforms to the principles and requirements of the Institute's guidelines for ethics in research.

Signed: Hugh Gallogher

Date: **27th January 2020**

ABSTRACT

This study has investigated the potential application of machine learning for video analysis, with a view to creating a system which can determine a person's hand laterality (handedness) from the way that they walk (their gait). To this end, the convolutional neural network model VGG16 underwent transfer learning in order to classify videos under two 'activities': "walking left-handed" and "walking right-handed". This saw varying degrees of success across five transfer learning trained models: Everything – the entire dataset; FiftyFifty – the dataset with enough right-handed samples removed to produce a set with parity between activities; Female – only the female samples; Male – only the male samples; Uninjured – samples declaring no injury within the last year.

The initial phase of this study involved a data collection scheme, as a suitable, pre-existing dataset could not be found to be available. This data collection resulted in 45 participants (7 left-handed, and 38 right-handed. 0 identified as ambidextrous), which resulted in 180 sample videos for use in transfer learning and testing the five produced models.

The video samples were recorded to obtain the volunteers' walking pattern, head to toe, in profile rather than head on. This was to allow the models to obtain as much information about arm and leg movement as possible when it came to analysis.

The findings of this study showed that accurate models could be produced. However, this varied substantially depending on the specific sub-dataset selected. Using the entire dataset was found to present the least accuracy (as well as the subset which removed any volunteers reporting injury within the last year). This resulted in a system which would classify all samples as 'Right'. In contrast the models produced observing the female volunteers (the gender which also provided the highest number of left-handed data samples) was consistently accurate, with a mean accuracy of 75.44%.

The course of this study has shown that training such a model to give an accurate result is possible, yet difficult to achieve with such a small sample size containing such a small population of left-handed individuals. From the results obtained, it appears that a population has a requirement of >~21% being left-handed in order to begin to see accuracy in laterality determination. These limited successes have shown that there is promise to be found in such a study. Although a larger, more wide-spread undertaking would be necessary to definitively show this.

Key words: *CNN, Machine Learning, Video Analysis, Gait, Laterality*

ACKNOWLEDGEMENTS

I would like to express my sincere thanks to my supervisor, Dr. Emma Murphy, who has been extremely patient with me over the course of my dissertation. She gave a lot of her time in providing invaluable guidance in undertaking my research as well as its documentation.

I would like to thank, as well, my work colleagues; Yvonne Keily and Ryan Lynch, for their time and effort in helping me organise data collection in my place of work.

I would like to thank my girlfriend, Emma Graunke, for help in proof-reading drafts (despite this research being outside of her field of study). As well as my family for the support they have given me over the months this undertaking took. With special thanks to my father, who helped me to organise data collection in his place of work.

TABLE OF CONTENTS

TABLE OF FIGURES

TABLE OF TABLES

1. INTRODUCTION

1.1 Background

This research study is rooted in human gait analysis, for the intent of determining hand laterality in humans. There are several small-scale applications for the end result of this research, an example of which is that it could be used in large scale workplace analysis of employees. Knowing the hand laterality of a workforce could be used to determine if any special actions or accommodations are necessary for workers with differing hand laterality to carry out adequate work.

Of particular interest would be in applying the technology to those who are nonverbal in nature. Many of those across the autism spectrum would fit this category (Bardikoff & McGonigle-Chalmers, 2014). Being able to determine the handedness of these individuals could greatly help should they require any support or technologies which differ based on a user's handedness.

A lot of work has gone into gait analysis as a breakdown of physiological factors and movements. This can be used to determine the laterality of a person. However, most instances of this have been in determining foot/leg laterality rather than hand laterality, as most studies involve additional equipment to measure forces exerted by the individual on the ground (Maupas, Paysant, Datie, Martinet, & André, 2002).

In any of these cases, machine learning has not been abundantly applied to breakdowns of laterality, be they hand or foot. It is a method which could speed up identification, and the exploration of such an application of machine learning could prove useful in other areas of gait analysis.

Initially a Support Vector Machine (SVM) model was elected to be trained for this issue of machine learning, and so the research question in the following section was based around this model. Following the commencement of this undertaking, a Convolutional Neural Network (CNN) was found to be more suited to the task. This change will be further explored in the discussion section of this document.

Additionally, sensors other than a simple camera, such as those used to measure kinetic force of footfalls or limb movement (Begg & Kamruzzaman, 2005; Abellanas, Frizera, Ceres, & Raya, 2009), are typically used in research studies involving gait. For a study observing such a specific and precise aspect of gait, solely video analysis has not been utilised.

1.2 Research Project/problem

The question sought to be answered through this study is: "Is it possible to determine a person's hand laterality through gait analysis, using an SVM model, with a relatively high (>70%) accuracy?"

1.3 Research Objectives

The objective of this research is to investigate H1 (below), while utilising the background work offered by H0, to show that it is possible to quickly, and with reasonable accuracy, determine a person's handedness, aided by a machine learning system.

H0: If a person's gait is analysed through physiological factors and biological movement, then it is possible to determine a person's laterality.

H1: If machine learning is used in gait analysis, then it is possible to quickly determine the laterality of a person's handedness.

1.4 Research Methodologies

Data collection was one of the main methodologies practiced over the course of this study. It was a key component, as a previously existing, suitable dataset could not be found. The data was collected through volunteer crowdsourcing. This data collection came in the form of a series of videos.

Alongside the video data collected, was a survey, which was completed by the previously mentioned volunteers after having been recorded. This became the source of the metadata relating to the videos which could not be found for previously existing datasets.

Data analysis was then carried out on the two previously mentioned sets of data (videos and metadata). This analysis was the main methodology utilised for the second part of the undertaken research.

1.5 Scope and Limitations

The scope of this research extends between machine learning and potential medical applications. More specifically, in terms of machine learning, the use of computer vision and the training in the realm of transfer learning of Convolutional Neural Network models.

The largest, initial limitation facing this study was the lack of a pre-existing dataset to work with.

The data collection methods introduce potential for bias. This was due to the locations (business offices and a hospital) for data collection having been chosen by the researcher. The data in these locations was collected from the staff working there, as opposed to any customers or visitors therein. Being staff, these people were a selective sub-section of the world's population. The staff from these locations accounted for mainly business people, doctors, and nurses. It gave a good division between people who spend much of their day seated, vs those who spend much of it on their feet. In either case the majority of these subjects came from similar backgrounds (education, lifestyle, etc.).

A heavy, secondary source of bias comes in the form of the percentage of younger participants. As can be seen in Table 3-1, more than half of the sample population were between 20-24 years of age.

1.6 Document Outline

The document that follows carries through a background literature review, providing a more in-depth look than is provided above in 1.1 Background. Following this is the methodology undertaken for data collection, and the pre-processing carried out before the machine learning segment of the research project was undertaken.

Following this will be a discussion of the results of the machine learning, as well as an additional outline for a future study which could follow this undertaking.

2. LITERATURE REVIEW

2.1 Video Metadata Extraction Techniques

The background research for this research project focused heavily on work done featuring video analysis through machine learning methods (Nasreen, Vinutha and Shobha, 2017), as well as work undertaken on more classical gait analysis. (Sadeghi et al., 2000). However, not all past studies investigated made use of video analysis techniques (Corballis & Morgan, 1978)

A major factor of modern, non-video driven gait analysis is the use of additional sensor equipment to provide readings of exerted forces from each limb (Abellanas, Frizera, Ceres, & Raya, 2009; Abdulhay, Arunkumar, Narasimhan, Vellaiappan, & Venkatraman, 2018). This additional equipment will be absent from this research.

Of particular interest among video analysis techniques were methods of silhouette extraction (Han Su and Feng-Gang Huang, 2005; Yamada, Ebihara and Ohya, 2002; Agarwal & Triggs, 2004), as well as silhouette analysis for use in machine learning research. (Agarwal and Triggs, 2004; Barnich, Jodogne, & Van Droogenbroeck, 2006). A mixture of methods have been used to implement this in the past (Dadashi, Araabi, & Soltanian-Zadeh, 2009). The most common and straightforward method is to use background subtraction for videos, with no camera movement. This shows a high level of success in extracting a clean silhouette as displayed by Yamada, Ebihara, & Ohya, 2002. Their work showed that this method was also feasible for use in real-time video processing.

2.2 Use Cases of Gait Analysis

It was imperative to find out the most widely used and reliable sub-section of machine learning commonly implemented for the study of human gait analysis. Through various readings this was initially found to be the use of Support Vector Machine models (Dadashi, Araabi and Soltanian-Zadeh, 2009). However, where this work falls short of the aim of this research project is that most instances of silhouette analysis are used for the determination of the presence of a human in a video (Agarwal & Triggs, 2004; Sidenbladh, 2004), rather than any more detailed, specific information about said human, or humans (Begg and Kamruzzaman, 2005; Barnich, Jodogne, & Van Droogenbroeck, 2006).

However, there has been a lot of work done implementing machine learning for the purposes of medical diagnoses. However, most conditions that are sought to be determined have an effect on the body as a whole rather than the specifics of laterality. Abdulhay, Arunkumar, Narasimhan, Vellaiappan, & Venkatraman, 2018 have shown great success in producing a novel system for the diagnosis and, following diagnosis, the severity of Parkinsons Disease in patients. They achieved this through analysis of kinetic footfall data (toe and heel), and used this inconjuntion with the time spent in moving all four limbs to deduce the presence of tremors in patients. These tremors are indicitive of Parkinsons Disease, and the severity of it was used to determine the severityu of the condition.

A similar, medical study looked at the diagnosis of Coplex Regional Pain Syndrome. In this instance the diagnosis was achieved using a Multilayer Perceptron Neural network to examine accelerometer data obtained from short distance walking undertaken by participants. This study showed great results in a non-visual oriented methodology, at an accuracy of 85.7% (Yang, et al., 2012).

While there has so far been little work done on hand laterality determination using machine learning methods, this does not detract from the benefit this knowledge would have.

For individuals with Autism Spectrum Disorder, video games are often a source of enjoyment and mental stimulation (Malinverni, et al., 2017). While the ultimate developmental effects of lengthy time playing video games on this section of the population can be questioned, with som individuals presenting pathological use of video games (O. Mazurek, R. Engelhardt, & E. Clark, 2015) , the fact is that for many of them it is an outlet of significant importance. Many of these individuals tend to develop as nonverbal, and often remain as so through adulthood. It has been shown that significant portions of the poputlation of those with ASD (40%) present as nonverbal (Bardikoff & McGonigle-Chalmers, 2014; Patten, K. Ausderau, R. Watson, & T. Baranek, 2013).

It has been shown that hardware and software built and designed with a left-handed individual's dominant hand in mind can increase their enjoyment of the overall experience (Maubert Crotte, H. Hepting, & Roshchina, 2019). A result of this study would be a system which could correctly identify the handedness of a nonverbal individual and so be given the ideal hardware to enjoy their outlet.

Although they are typically not situations where the individuals cannot self-identify as left or right handed (i.e. when they are verbal), other medical studies have shown the importance of knowledge of the handedness of the individual. It has been observed that in cases of female breast cancer patients in Sweden that the percentage of the effected sample population presenting as left handed (<1.5%) were significantly less than the population of left handed Swedish women (5%) (Olsson & Ingvar, 1991). This presented a likely correlation between handedness and liklihood of developing breast cancer.

A similar study concerning patients suffering with schizophrenia had comparable results when it came to the ratios of left to right handed individuals. In the study by Taylor, Dalton, & Fleminger, 1980 it was found that these patients had a much higher proportion of right-handed individuals than did the control groups.

While using differing mehodologies in laterality determination than those utilised through the work of this study, usablity of smartphone applications and interfaces can be improved upon with knowledge of which hand the user is holding the phone in – likely their dominant hand (Avery, Vogel, Lank, Masson, & Hanae, 2019). When this can be determined, single handed modes of use may be automatically activated, rather than manually as most phones do now. For example, Android devices enter one handed mode when swiping onto the screen from either of the bottom corners.

This use case would generally implement data gathering from sensors within the phone to determine which hand is operating the device, and with an assumption of dominance therefore also determine the lateralityof the individual using the device (Löchtefeld, Schardt, & Krüger, 2015).

This manner of thinking for usablilty of devices could be tied back to workplace applications. Should IoT connected, or similar, specialised devices be necessary for use in the work place, but lacking accelerometer data gathering methods, a mixed implementation of this study and facial recognition techniques could be used to produce more personalised interfaces for each employee. A more user friendly device would no doubt lead to an increase in worker productivity.

2.3 Machine Learning Model Selection

Though initial research suggested the promise of a Support Vector Machine model, this was later found to be insufficient for this undertaking. As such further research was carried out to find a more appropriate machine learning model to use. Following this further research, it was found that a Convolutional Neural Network (CNN) would be more suited to carrying out this particular research endeavour (Ignatov, 2018; Simonyan & Zisserman, 2015). Such machine learning systems have been implemented in a wide variety of image recognition (Gopalakrishnan, K. Khaitan, & Agrawal, 2017), classification (Simonyan & Zisserman, 2015), and activity identification (Ignatov, 2018) projects.

Initially it was planned to train a full model, however due to the expected low number of final data points (given the volunteer nature of the data collection) it was necessary to find an alternative approach. As such the practice of Transfer Learning within the realm of machine learning was investigated. This practice is generally used in situations such as this study, where the dataset being observed has a low number of entries. The main use case of transfer learning is when a dataset of interest is limited in its scope, yet a machine learning model implemented on a similar dataset exists (Pan & Yang, 2010).

The chosen, pre-existing model can be further trained on the dataset of interest in the capacity needed for that dataset's identification or classification (Dai, Yang, Xue, & Yu, 2007). Of particular interest for the researcher of this study was the VGG16 CNN pre-trained model. This model is investigated and discussed by Simonyan & Zisserman, 2015 in their paper "Very Deep Convolutional Networks for Large-Scale Image Recognition".

This model was named after the Visual Geometry Group from Oxford, and employs a 16-weight layer architecture, hence “VGG16”.

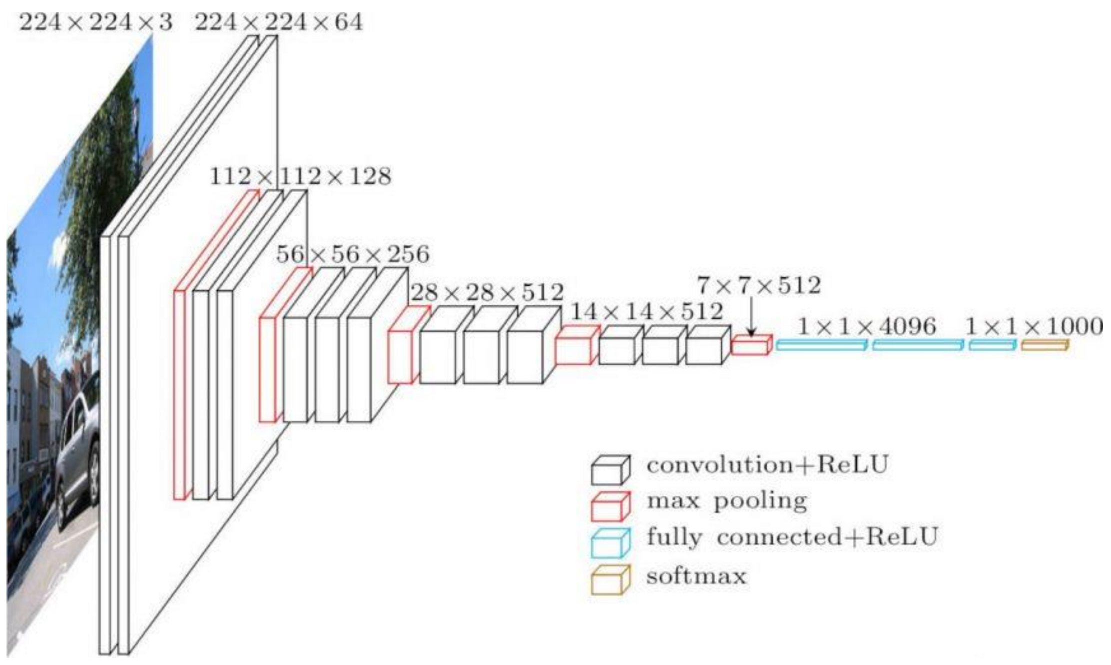


Figure 2-1 VGG16 Architecture Representation

This CNN sees improvements on accuracy over earlier designs (Krizhevsky, Sutskever, & E. Hinton, 2012) due, in part, to its increased depth and differences in receptive fields (Yu, et al., 2016), using a small (3x3) field with a stride of 1.

The main use cases of this model are within the realm of image classification. There are abundant examples across many differing industries. As discussed above, most of these studies made use of the power of transfer learning to augment this model for their specific use cases.

The work of Aqib Haqmi Abas, Ismail, Ihsan Mohd Yassin, & Nasir Taib, 2018 used this model for the calssification of plants, with great levels of success. Using similar methods of transfer learning was the work of Liu, Zhang, Gao, & Chen, 2018, which used the same base model for the purposes of weld defect identification. This was a “non-destrucive testing” method, which was sought to reduce the time spent by skilled technicians who would typically carry out such identification.

With a world-wide data total greatly increasing each year, it is necessary to develop more novel solutions to data analysis. One such novel use case, which shows the robustness of this base model, was the application of using transfer learning for the purposes of malware (malicious software) classification. This was achieved by representing the sample set as byteplot greyscale images. This study presented an accuracy of 92.97% in correct classification of malware (Rezende, et al., 2018).

3. DESIGN AND METHODOLOGY

3.1 Dataset

3.1.1 Dataset Identification

The initial phase of this study was identifying a suitable dataset to perform the necessary machine learning methods on. Visually sufficient datasets for what was required could be found, however they lacked the metadata required for this specific study. As it was not possible to carry out this study with the identified, pre-existing datasets the initial stage of this research became data collection and dataset construction.

This proved difficult to obtain a large volunteer population for data gathering. However, ultimately a baseline-sufficient number of data points were collected. The size and scope of the collected data will be discussed later, throughout this document.

3.1.2 Participants

The main location for data collection was in the offices of a large company, with the employees working there being the volunteers for the study. The secondary location for data collection was within a hospital, with some of the members of staff being the volunteers for the study.

While the video quality of samples obtained through the data collection undertaken in the hospital resulted in cleaner, crisper silhouettes (as discussed in 5.5.1), there were too few left-handed samples obtained (1 in 11) to be able to both train and test an accurate model for the purposes of this research.

This sample gave an almost even split in binary-gender ratio; 54.55% to 45.45% Male to Female, with one participant opting not to answer. It was decided that one participant not answering was not enough data to require a separate category.

15.56% of the sample pool identified as having a left-handed laterality, while only 2.94% identified as having a left footed laterality. One participant identified as having an ambidextrous foot laterality.

The modal age range among participants was 20-24, at 53.33% of participants. 30-34 was the next most represented age range, at 13.33%. Each other age range given was represented at a non-zero percentage.

From the sample population 6 individuals (13.33%) reported having had a leg injury within the last year. This was important to note, as any form of leg injury could affect an individual's gait and alter the perceived laterality of such an individual. Of these individuals 2 were male, 3 were female, and the remaining was the individual who declined to provide their gender.

Age Range	Female Percentage	Male Percentage	Total Count
20-24	0.54	0.42	24
25-29	1.0	0.0	3
30-34	0.33	0.67	6
35-39	1.0	0.0	2
40-44	0.0	1.0	1
45-49	0.0	1.0	1
50-54	0.5	0.5	2
55-59	0.4	0.6	5
60-64	0.0	1.0	1

Table 3-1 Total Volunteer Population

Age Range	Count	Average Height (cm)	Percentage Injured
20-24	13	166.63	7.69
25-29	3	165.85	33.33
30-34	2	177.90	0
35-39	2	159.39	0
40-44	0	---	---
45-49	0	---	---
50-54	1	167.64	100
55-59	2	173.26	0
60-64	0	---	---

Table 3-2 Female Volunteer Population

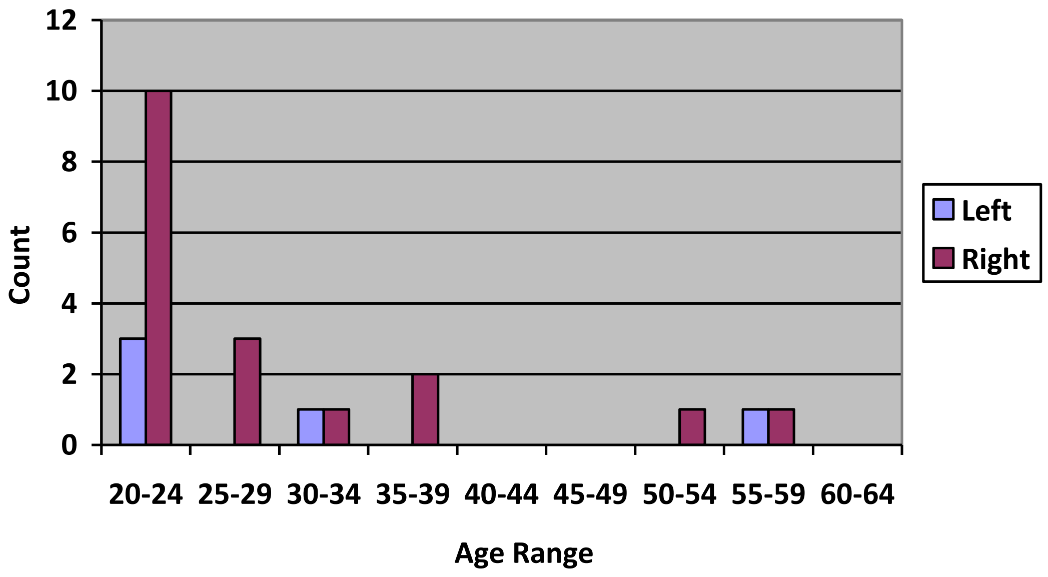


Figure 3-1 Female Handedness Breakdown

The breakdown of female handedness shows that the entirety of the female population observed (23) presented a ratio of 5:18 Left:Right.

Age Range	Count	Average Height	Percentage Injured
20-24	10	184.74	20
25-29	0	---	---
30-34	4	177.5	0
35-39	0	---	---
40-44	1	158.00	0
45-49	1	175.26	0
50-54	1	175.26	0
55-59	3	180.20	0
60-64	1	182.88	0

Table 3-3 Male Volunteer Population

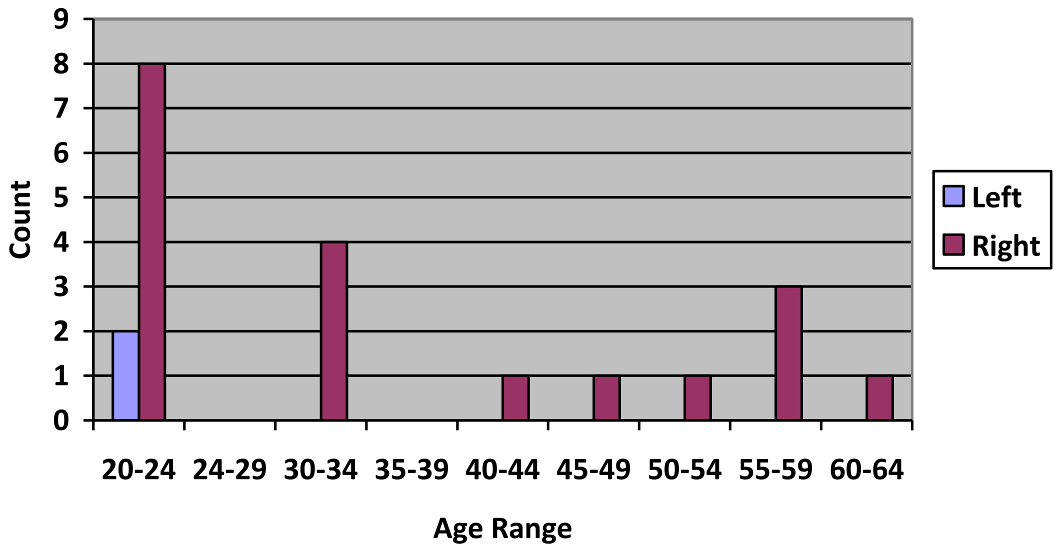


Figure 3-2 Male Handedness Breakdown

The breakdown of male handedness shows that the entirety of the male population observed (21) presented a ratio of 2:19 Left:Right.

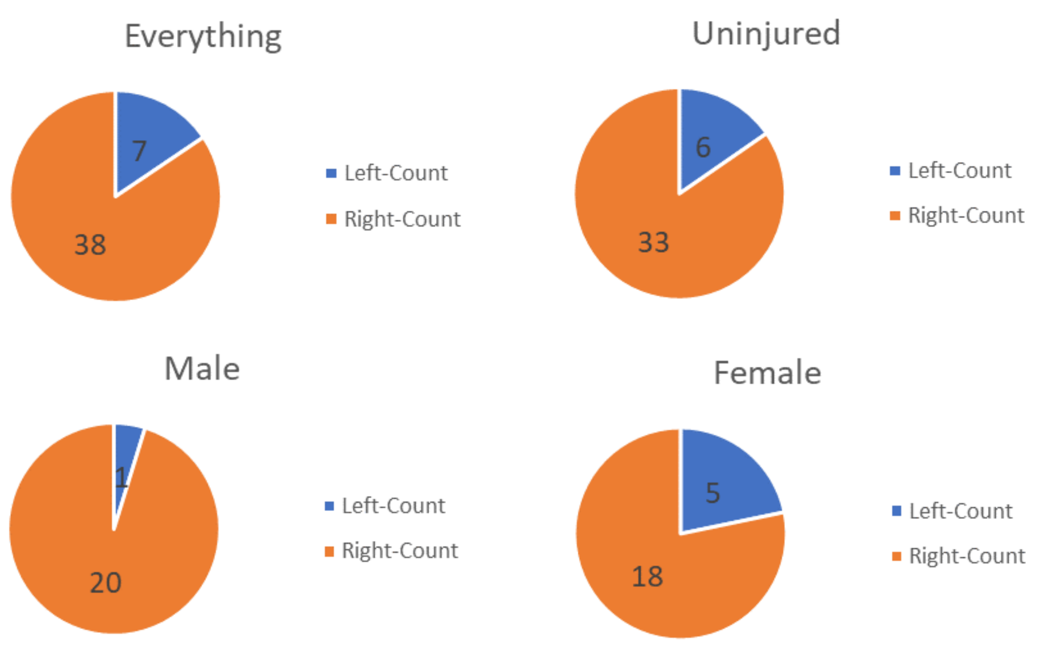


Figure 3-3 Pie-Chart Population Break Downs

The above pie-chart representation of each population considered for a training model has omitted 'FiftyFifty', as this population is artificially reduced to provide a 50:50 ratio and so would result in a half-orange, half-blue pie chart.

3.1.3 Data Collection Methodology

Data collection was carried out using basic, commodity equipment. A Sony Xperia XA2 H3113 phone's rear facing camera was used to capture the walking footage. This phone was running Android Version 9, with a build number of '50.2.A.0.400'. The application 'Open Camera', version 1.47.3, was favoured over the built-in camera application.

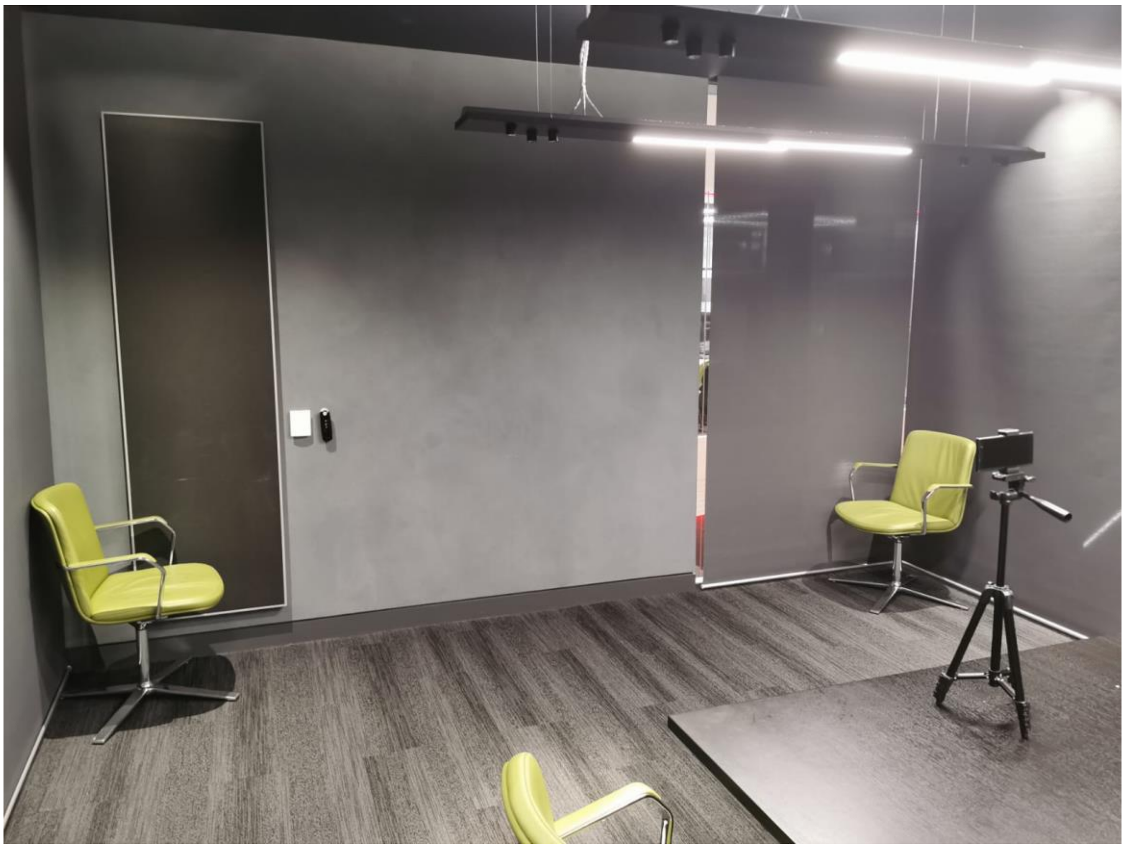


Figure 3-4 Data Collection Setup

The videos produced were saved with the following naming structure:
“MSC_VID[YYYYMMDD]_[hhmmss].mp4”
Where the letters in brackets were replaced as follows:

- YYYY with the current year (2019)
- MM with the month (10 or 11, in this instance)
- DD with the date
- hh with the hour of start of recording
- mm with the minute of start of recording
- ss with the second of start of recording

A protocol script, as can be seen in Appendix A – documents, was created in order to conduct a consistent data collection experience with each volunteer. The script and the accompanying consent form cover all of the basic aspects of the data collection, as well as informing the volunteers how their data will be treated and protected.

Following the introduction to the data collection, volunteers were given the opportunity to ask any questions they had prior to the video recording. At this stage they were simply informed that this was a study on gait, with no specifics about hand laterality provided.

Volunteers were then requested to walk a set path in the room in use. This path followed a back and forth walking pattern across a 3.2-meter distance. The limitation to this distance was set by the size of the space available. The back and forth path was walked three times to allow the volunteers to affect a typical gait. The third walk back and forth were each used as a collected data point. This path was walked twice by each participant.

The lighting within the video was artificial and natural lighting was blocked out, as any significant changes in lighting would become apparent once the silhouette extraction process was performed on the collected videos. The fluctuations of natural lighting were wished to be avoided, as they would present themselves as random changes when extraction was to be carried out. As the silhouette is generated based on changes between successive frames of the video, random shadows cast due to these fluctuations would make themselves seen as additional silhouettes.

These lighting fluctuations were avoided in the hospital setting as it was a windowless location.

After the volunteers had walked, they were presented with a questionnaire to be filled in. This collected the required data not present in the previously identified datasets. Some additional data were recorded, such as age range, gender, height, etc. The extent of this collection can be seen in Table 3-1.

3.1.4 Data Pre-Processing

Following data collection, the videos collected were processed to extract the volunteers' silhouettes. This was achieved through methods of background extraction. The code for this is provided in Appendix B - Code.

This background extraction method primarily used the OpenCV (version 4.1.2.30) Python library to carry out the silhouette extraction. The raw, pre-extraction footage is opened and read in. The frame size (height and width) were extracted to create the dimensions for the output file. A similar naming convention to that of the raw footage was utilised for the extracted videos. This naming convention replaced the "MSC" in the raw footage with "SIL" in the silhouette extracted footage.

To extract the silhouette the code compares each successive pair of frames to detect change. This change is then shown as a white 'shadow' on a black background.

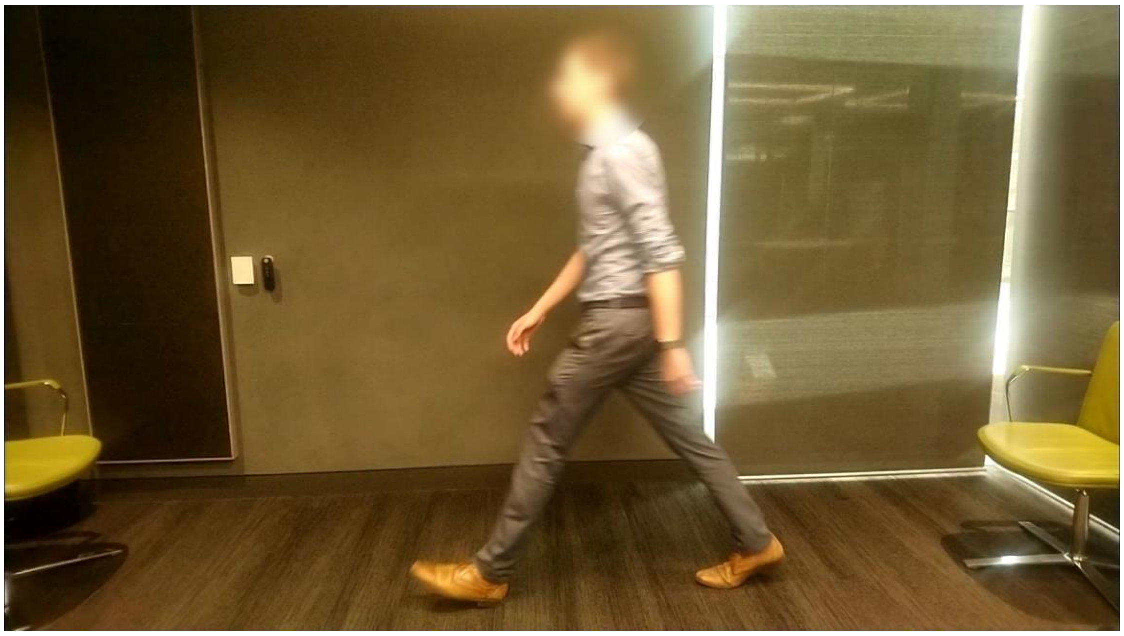


Figure 3-5 Pre-Extraction (Male, Office)

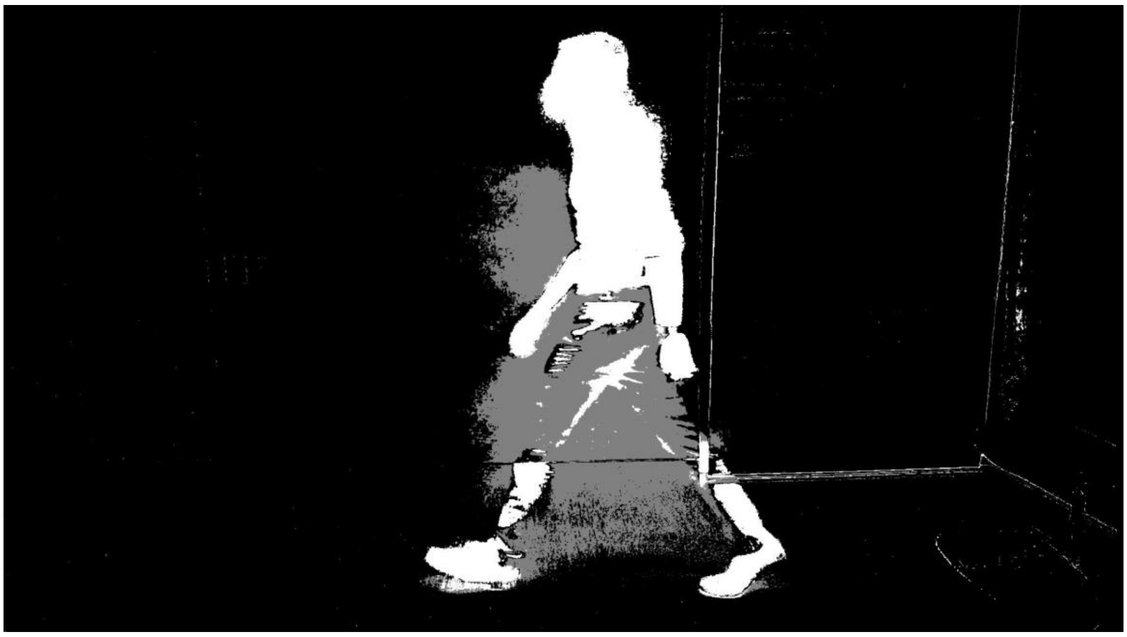


Figure 3-6 Post-Extraction (Male, Office)

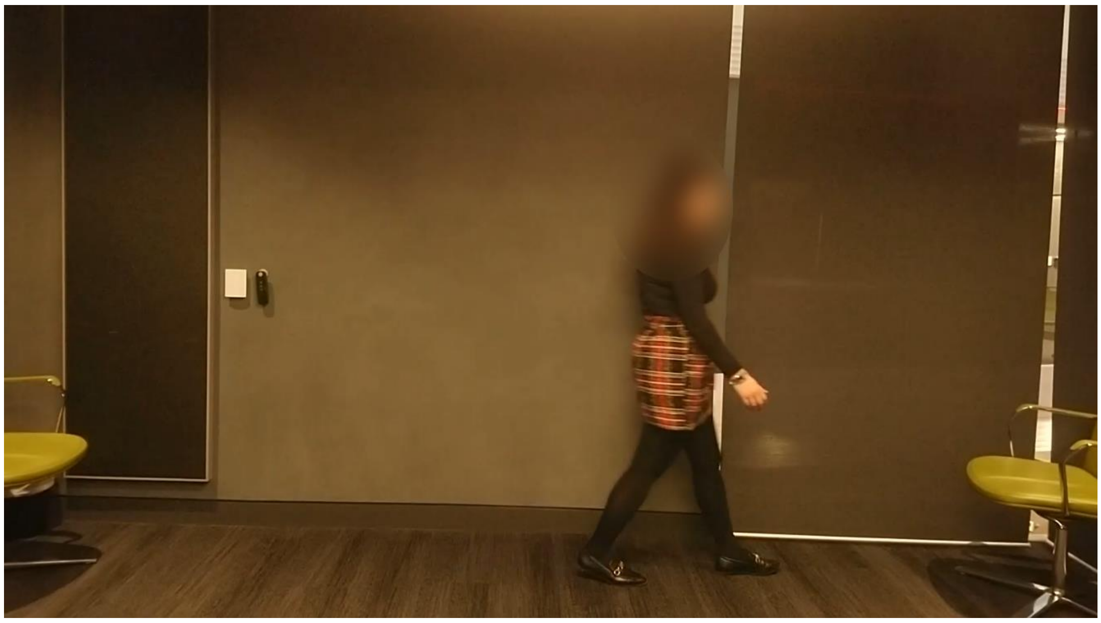


Figure 3-7 Pre-Extraction (Female, Office)

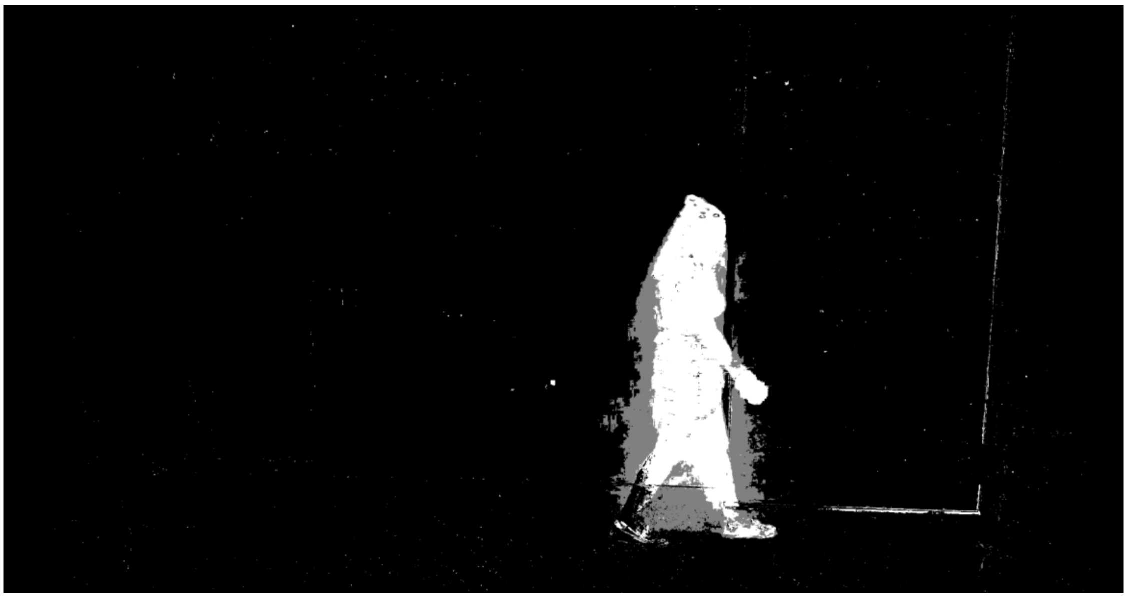


Figure 3-8 Post-Extraction (Female, Office)

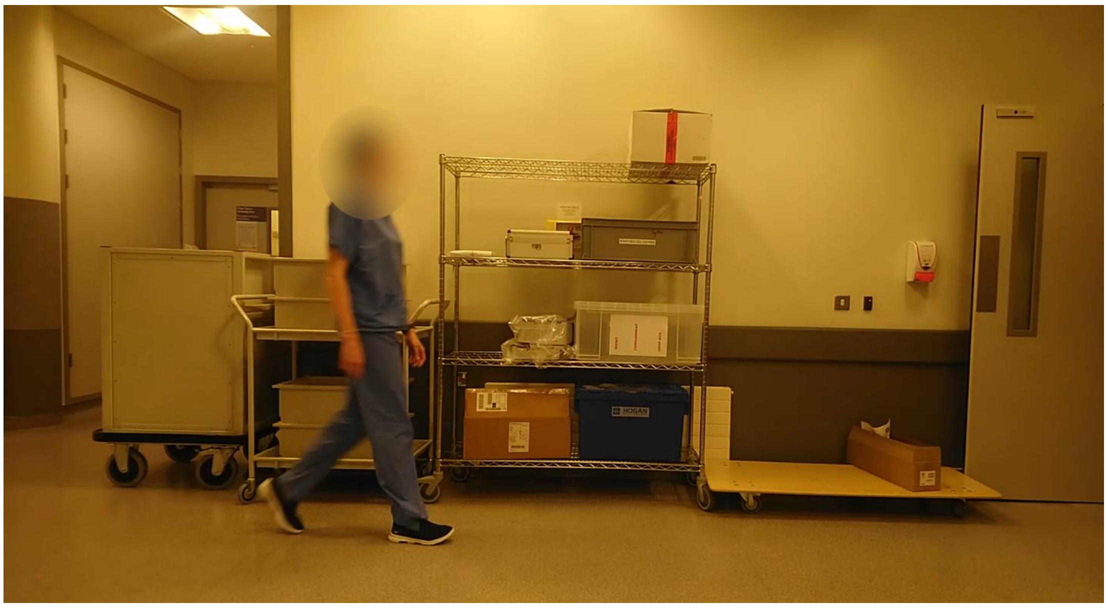


Figure 3-9 Pre-Extraction (Female, Hospital)

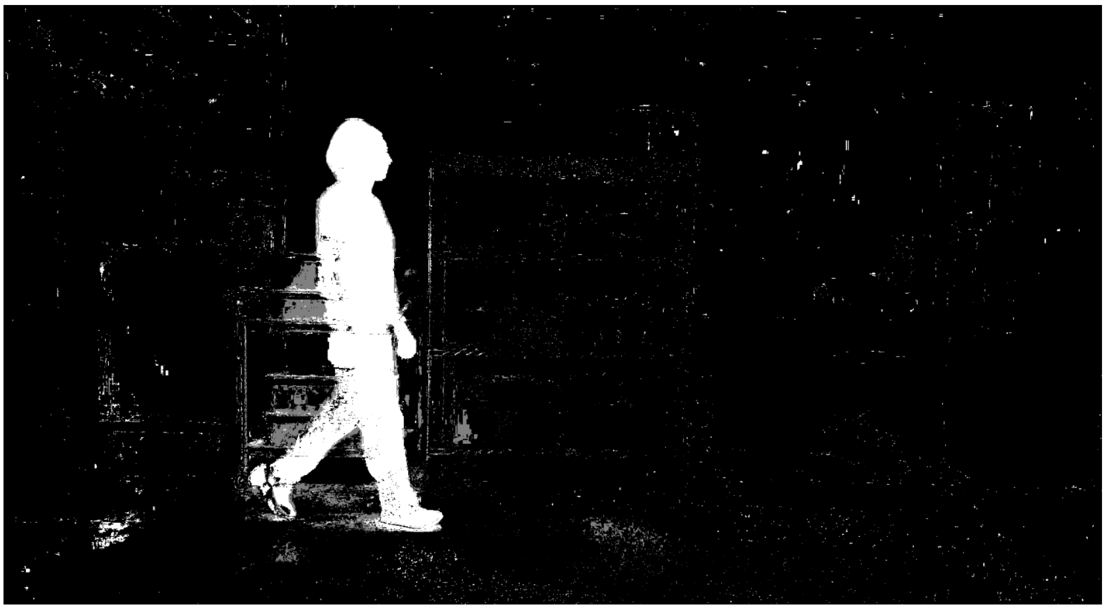


Figure 3-10 Post-Extraction (Female, Hospital)

After the silhouette-extracted video was produced, the clips were trimmed to isolate the volunteers' third walks back and forth. This clip trimming was carried out using the 'Microsoft Photos' software (version 2019.19081.22010.0), built into Windows 10. The names of each of these clips were then recorded in text files to be used later. Ten such text files were written, five pertaining to training data, and five pertaining to testing data. These ten files were passed through to the machine learning system, with the accompanying, silhouette-extracted video files.

All processing was undertaken on a home-built PC. It had an Intel i5 CPU, an Nvidia GTX 970 GPU, and was running Windows 10. The version of python used throughout this research was version 3.7.4.

3.2 Machine Learning

3.2.1 Initial Machine Learning Method (SVM)

Initially a Support Vector Machine (SVM) approach was explored for use. This involved the use of the openpose python library, which can be used to determine the approximate location of key points of bodily motions. These key points are points of motion throughout the body; namely the neck, shoulders, elbows, hands, hips, knees, and ankles.

This data would have been extracted in the form of coordinate data, to be presented as the video metadata to the machine learning system for a model to be trained. As will be discussed later, this approach was abandoned as it was found to be unsuitable.

3.2.2 Final Machine Learning Methods (CNN)

The practice of Transfer Learning was then undertaken, to retrain a pre-existing model for this use case. The machine learning elements of this were carried out using Keras (version 2.3.1), wrapping Tensorflow (version 2.0.0) or Tensorflow-GPU (version 2.0.0) where appropriate. The VGG16, and model was elected for use in this instance. It was further trained to differentiate the two activities of "walking left-handed" and "walking right-handed".

The python script implemented first reads in a text file, listing the videos to be used as training data. This text file having been previously created during the silhouette extraction phase. For this study a split of 80/20 was between training and testing data was elected. This was to allow for sufficient testing data, given the relatively small sample size. The script then scanned through the training video samples to extract still frames as sequential images.

These images are stored in a format of "[tag]_[original video name]_frame[frame number]". Where the 'tag' is either 'Left' or 'Right'. The 'original video name' is the name of the silhouette extracted video that the frame is taken from. And 'frame number' is the sequential number of the given frame from the video. This numbering is zero indexed. The frames were extracted at a frame rate of 5, resulting in every fifth frame being extracted to later be presented as a sequential image.

A comma separated value (csv) file was also created, listing each frame image as well as its associated class. With the class being equivalent to the tag mentioned above. After the csv file and the frame images were created the next python script was run.

The next script then read in the csv file in order to identify all extracted frames, as well as to identify the class of each. The VGG16 model then underwent transfer learning, producing five different models, over 200 epochs to identify the two chosen activities. This training was conducted using both a CPU and GPU methodology. The GPU methodology used version 10.2 of CUDA for tensorflow to make use of the GPU. The results from both methodologies shall be discussed in following sections.

The final python script then used the list of testing videos to sequentially open and examine each, and to identify the class of the opened video through a sequential image scanning of each video. This was similar to how the image frames were extracted for the training of each model.

3.2.3 Devising Transfer Learning Models

Five model types were selected to be trained:

1. Entire volunteer population (weighteverything.hdf5)
2. Population reduced for left- and right-handed parity (weightFiftyFifty.hdf5)
3. Population reduced to males only (weightmale.hdf5)
4. Population reduced to female only (weightfemale.hdf5)
5. Population reduced to only those reporting no injury (weightuninjured.hdf5)

Models 2 through 5 were devised when preliminary results from model 1 presented only a system which assigned a label of ‘Right’ to all testing data. Making use of these additional models proved to be much more informative for the entirety of the research, as will be discussed later.

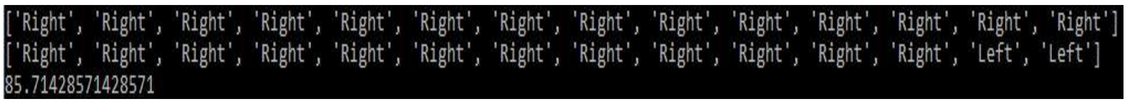
4. RESULTS, EVALUATION, AND DISCUSSION

The initially attempted machine learning method (pre-processing for use with an SVM system), produced no results. The openpose python library was to be used to determine approximate joint positions from each recorded, silhouette extracted video. However, this library was found to be unable to do so. It was surmised that this is likely due to the relatively noisy videos obtained from the silhouette extraction step.

In addition to this statement it must be noted that the same functions were attempted to be used on a small selection of the pre-extraction videos for the sake of completeness. These were also found to yield no extracted information. As such the use of this library was abandoned, and further research led to the method of performing transfer learning on pre-existing CNN models. This methodology is outlined in 3.2.3 above.

4.1 Machine Learning Utilising CPU

Utilising this new method of transfer learning found success in further training the pre-trained model. The initially further trained model showed an apparent ability to excel at correctly identifying hand laterality in 85.71% of test cases. Unfortunately, this high 'accuracy' was the result of this model making the 'lucky guess' that each sample showed a right-handed individual. Due to the high number of right-handed individuals present in the population, they were found to make up ~85% of both training and testing data. As such this observed 'accuracy' was as successful as a 0% accuracy.

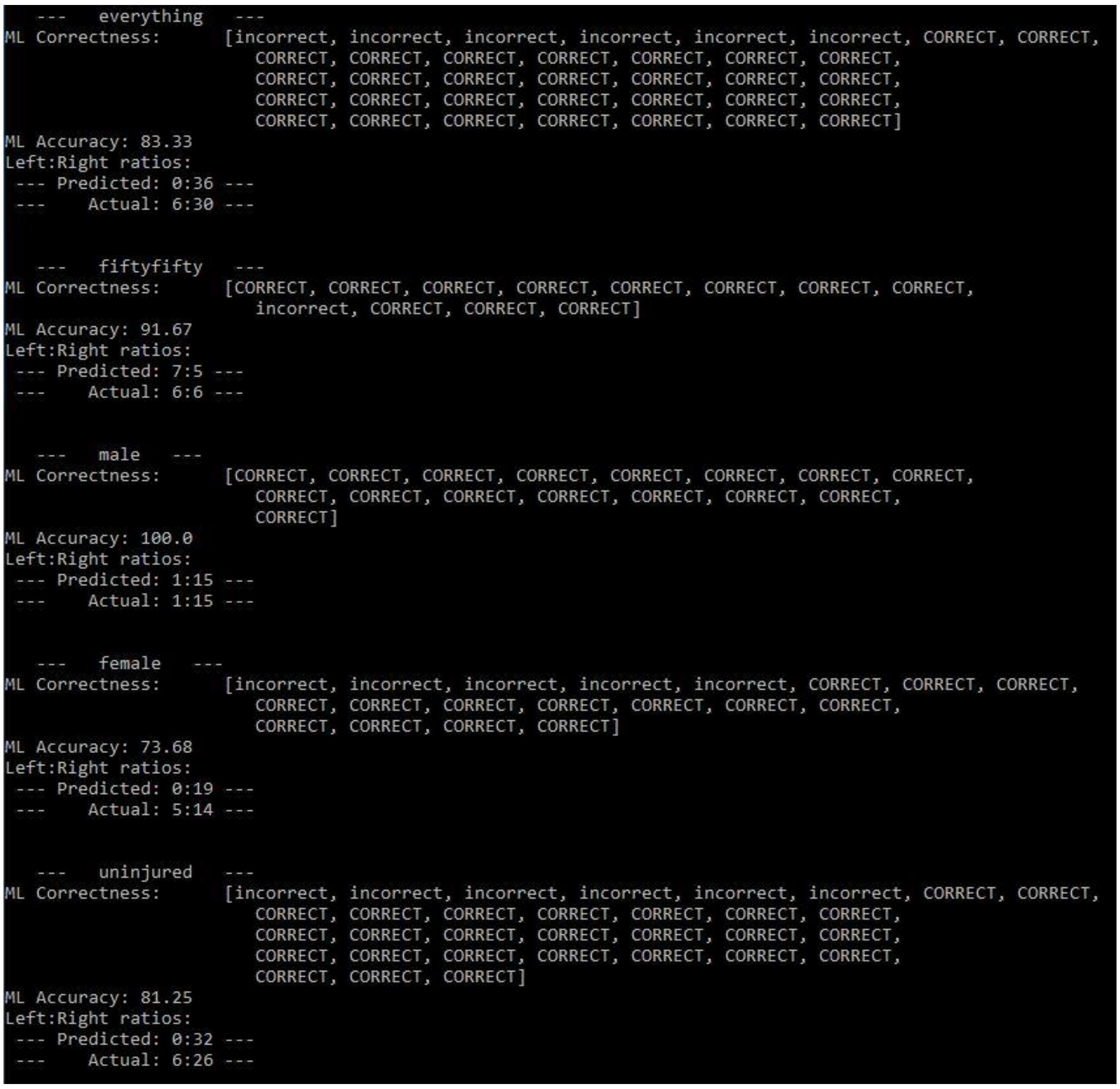


```
['Right', 'Right', 'Right', 'Right', 'Right', 'Right', 'Right', 'Right', 'Right', 'Right', 'Right', 'Right', 'Right', 'Right']
['Right', 'Right', 'Right', 'Right', 'Right', 'Right', 'Right', 'Right', 'Right', 'Right', 'Right', 'Right', 'Left', 'Left']
85.71428571428571
```

Figure 4-1 CNN Initial Test Result

The main take away from this initial result was that the population was oversaturated with right handed walkers. The resulting issue of this was that the training model was provided with very few data points relating to left handed individuals. This led to devising the approach for the next round of transfer learning.

As this study had a split of 15.56%:84.44% Left:Right, it was decided that the sample population should be artificially limited to produce a data set where parity could be observed between left and right handed individuals.



```
   ---   everything   ---
ML Correctness:      [incorrect, incorrect, incorrect, incorrect, incorrect, incorrect, CORRECT, CORRECT,
                        CORRECT, CORRECT, CORRECT, CORRECT, CORRECT, CORRECT, CORRECT,
                        CORRECT, CORRECT, CORRECT, CORRECT, CORRECT, CORRECT, CORRECT,
                        CORRECT, CORRECT, CORRECT, CORRECT, CORRECT, CORRECT, CORRECT,
                        CORRECT, CORRECT, CORRECT, CORRECT, CORRECT, CORRECT, CORRECT]
ML Accuracy: 83.33
Left:Right ratios:
 --- Predicted: 0:36 ---
 ---    Actual: 6:30 ---


   ---   fiftyfifty   ---
ML Correctness:      [CORRECT, CORRECT, CORRECT, CORRECT, CORRECT, CORRECT, CORRECT, CORRECT,
                        incorrect, CORRECT, CORRECT, CORRECT]
ML Accuracy: 91.67
Left:Right ratios:
 --- Predicted: 7:5 ---
 ---    Actual: 6:6 ---


   ---   male   ---
ML Correctness:      [CORRECT, CORRECT, CORRECT, CORRECT, CORRECT, CORRECT, CORRECT, CORRECT,
                        CORRECT, CORRECT, CORRECT, CORRECT, CORRECT, CORRECT, CORRECT,
                        CORRECT]
ML Accuracy: 100.0
Left:Right ratios:
 --- Predicted: 1:15 ---
 ---    Actual: 1:15 ---


   ---   female   ---
ML Correctness:      [incorrect, incorrect, incorrect, incorrect, incorrect, CORRECT, CORRECT, CORRECT,
                        CORRECT, CORRECT, CORRECT, CORRECT, CORRECT, CORRECT, CORRECT,
                        CORRECT, CORRECT, CORRECT, CORRECT]
ML Accuracy: 73.68
Left:Right ratios:
 --- Predicted: 0:19 ---
 ---    Actual: 5:14 ---


   ---   uninjured   ---
ML Correctness:      [incorrect, incorrect, incorrect, incorrect, incorrect, incorrect, CORRECT, CORRECT,
                        CORRECT, CORRECT, CORRECT, CORRECT, CORRECT, CORRECT, CORRECT,
                        CORRECT, CORRECT, CORRECT, CORRECT, CORRECT, CORRECT, CORRECT,
                        CORRECT, CORRECT, CORRECT, CORRECT, CORRECT, CORRECT, CORRECT,
                        CORRECT, CORRECT, CORRECT]
ML Accuracy: 81.25
Left:Right ratios:
 --- Predicted: 0:32 ---
 ---    Actual: 6:26 ---
```

Figure 4-2 Further CNN Results; Output for Five VGG16 Models (CPU)

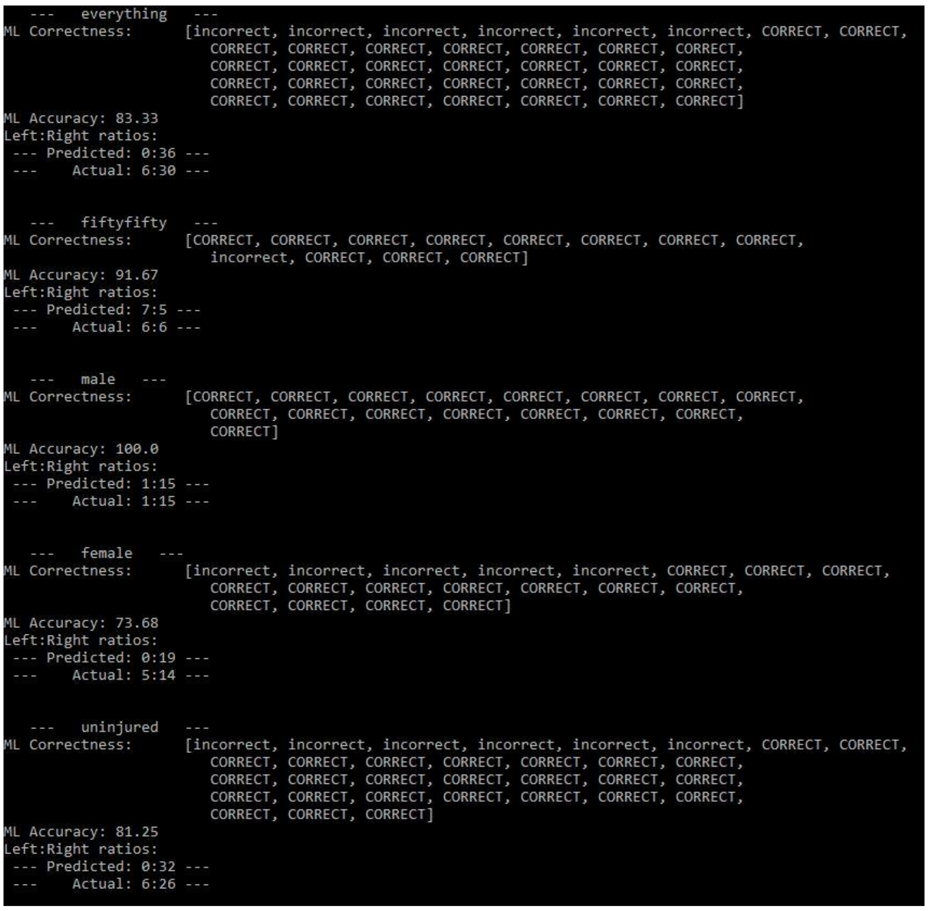
This artificial reduction proved to be much more successful (weightFiftyFifty). As seen in Figure 4-2 the correct identifications were made not through ‘lucky guesses’, but through correct classification of video samples presented. This particular model showed a success rate of 91.67%, misidentifying one test sample from twelve.

This suggests a potential for promise utilising this method. If a sample population were obtained with a more even split between left and right handedness, the result would likely be more accurate. As such, it appears that the major shortfall of this study is the size of the sample population, and lack of parity observed between the two classes to be identified.

4.2 Machine Learning Utilising GPU

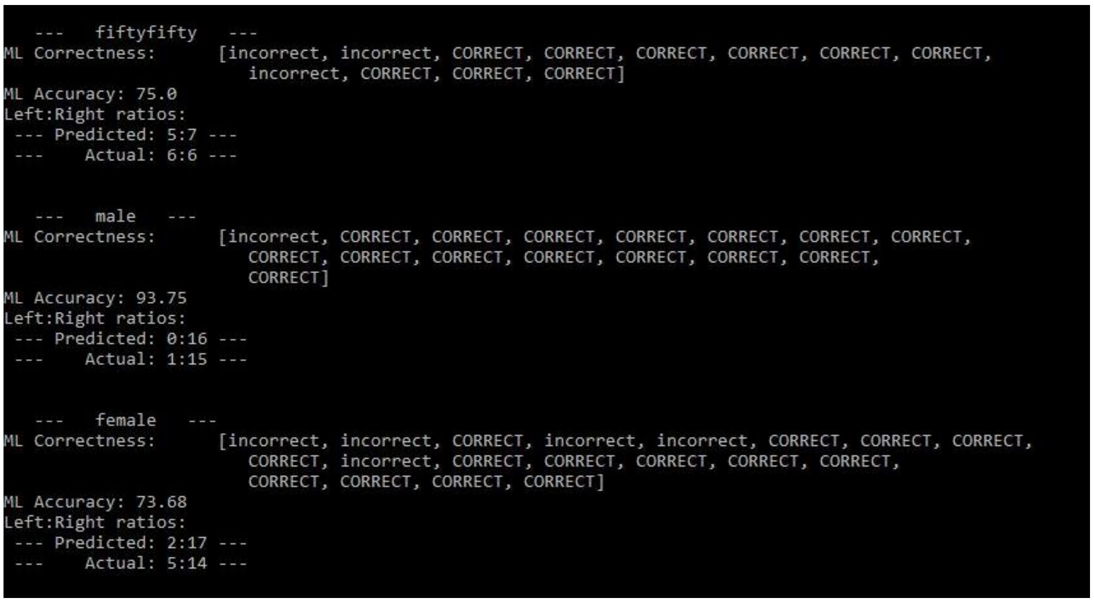
After the CPU tests were carried out, tensorflow was uninstalled from the computer, and replaced with tensorflow-gpu. This was to see the difference (if any) between the two calculation devices. The first noticeable, and unsurprising, difference was a vast increase in calculation speed when switching to the GPU method.

The models were all trained and tested multiple times, and the result was interesting. Three out of the five models displayed an amount of variance between each run. Shown in the figures below are three of the different runs, to display a sense of the variety of results found. The models for 'everything' and 'uninjured' have been omitted from Figure 4-4 and Figure 4-5 as they were invariant across the three separate runs.



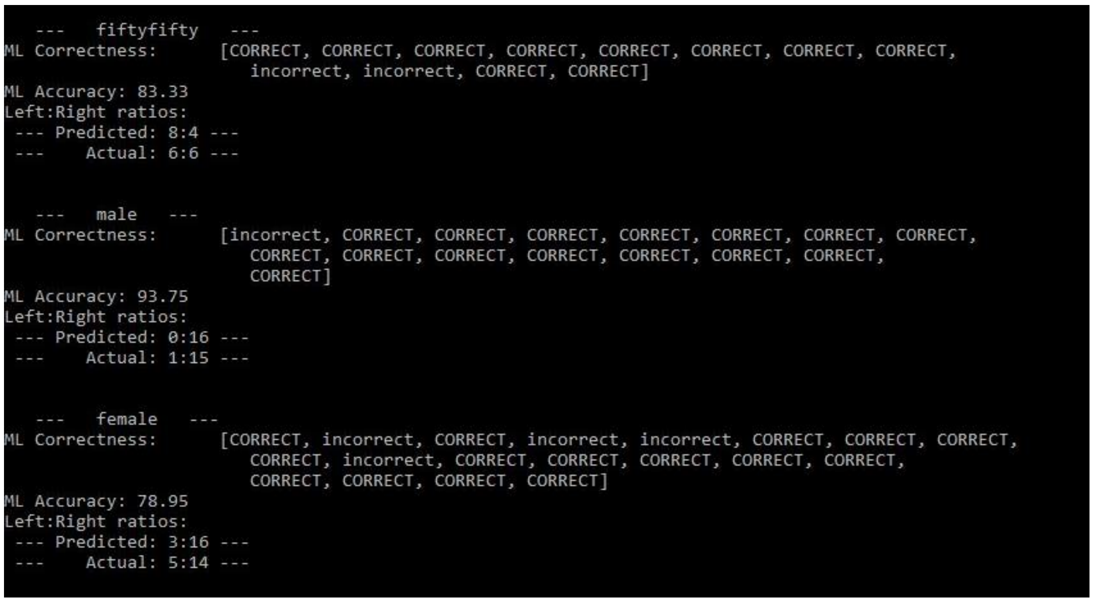
```
   ---   everything   ---
ML Correctness:      [incorrect, incorrect, incorrect, incorrect, incorrect, incorrect, CORRECT, CORRECT,
                        CORRECT, CORRECT, CORRECT, CORRECT, CORRECT, CORRECT, CORRECT,
                        CORRECT, CORRECT, CORRECT, CORRECT, CORRECT, CORRECT, CORRECT,
                        CORRECT, CORRECT, CORRECT, CORRECT, CORRECT, CORRECT, CORRECT,
                        CORRECT, CORRECT, CORRECT, CORRECT, CORRECT, CORRECT, CORRECT]
ML Accuracy: 83.33
Left:Right ratios:
 --- Predicted: 0:36 ---
 ---    Actual: 6:30 ---


   ---   fiftyfifty   ---
ML Correctness:      [CORRECT, CORRECT, CORRECT, CORRECT, CORRECT, CORRECT, CORRECT, CORRECT,
                        incorrect, CORRECT, CORRECT, CORRECT]
ML Accuracy: 91.67
Left:Right ratios:
 --- Predicted: 7:5 ---
 ---    Actual: 6:6 ---


   ---   male   ---
ML Correctness:      [CORRECT, CORRECT, CORRECT, CORRECT, CORRECT, CORRECT, CORRECT, CORRECT,
                        CORRECT, CORRECT, CORRECT, CORRECT, CORRECT, CORRECT, CORRECT,
                        CORRECT]
ML Accuracy: 100.0
Left:Right ratios:
 --- Predicted: 1:15 ---
 ---    Actual: 1:15 ---


   ---   female   ---
ML Correctness:      [incorrect, incorrect, incorrect, incorrect, incorrect, CORRECT, CORRECT, CORRECT,
                        CORRECT, CORRECT, CORRECT, CORRECT, CORRECT, CORRECT, CORRECT,
                        CORRECT, CORRECT, CORRECT, CORRECT]
ML Accuracy: 73.68
Left:Right ratios:
 --- Predicted: 0:19 ---
 ---    Actual: 5:14 ---


   ---   uninjured   ---
ML Correctness:      [incorrect, incorrect, incorrect, incorrect, incorrect, incorrect, CORRECT, CORRECT,
                        CORRECT, CORRECT, CORRECT, CORRECT, CORRECT, CORRECT, CORRECT,
                        CORRECT, CORRECT, CORRECT, CORRECT, CORRECT, CORRECT, CORRECT,
                        CORRECT, CORRECT, CORRECT, CORRECT, CORRECT, CORRECT, CORRECT,
                        CORRECT, CORRECT, CORRECT]
ML Accuracy: 81.25
Left:Right ratios:
 --- Predicted: 0:32 ---
 ---    Actual: 6:26 ---
```

Figure 4-3 Output 1 for Five VGG16 Models (GPU)



```
   ---   fiftyfifty   ---
ML Correctness:      [incorrect, incorrect, CORRECT, CORRECT, CORRECT, CORRECT, CORRECT, CORRECT,
                        incorrect, CORRECT, CORRECT, CORRECT]
ML Accuracy: 75.0
Left:Right ratios:
 --- Predicted: 5:7 ---
 ---    Actual: 6:6 ---


   ---   male   ---
ML Correctness:      [incorrect, CORRECT, CORRECT, CORRECT, CORRECT, CORRECT, CORRECT, CORRECT,
                        CORRECT, CORRECT, CORRECT, CORRECT, CORRECT, CORRECT, CORRECT,
                        CORRECT]
ML Accuracy: 93.75
Left:Right ratios:
 --- Predicted: 0:16 ---
 ---    Actual: 1:15 ---


   ---   female   ---
ML Correctness:      [incorrect, incorrect, CORRECT, incorrect, incorrect, CORRECT, CORRECT, CORRECT,
                        CORRECT, incorrect, CORRECT, CORRECT, CORRECT, CORRECT, CORRECT,
                        CORRECT, CORRECT, CORRECT, CORRECT]
ML Accuracy: 73.68
Left:Right ratios:
 --- Predicted: 2:17 ---
 ---    Actual: 5:14 ---
```

Figure 4-4 Output 2 for Three VGG16 Models (GPU)



```
   ---   fiftyfifty   ---
ML Correctness:      [CORRECT, CORRECT, CORRECT, CORRECT, CORRECT, CORRECT, CORRECT, CORRECT,
                        incorrect, incorrect, CORRECT, CORRECT]
ML Accuracy: 83.33
Left:Right ratios:
 --- Predicted: 8:4 ---
 ---    Actual: 6:6 ---


   ---   male   ---
ML Correctness:      [incorrect, CORRECT, CORRECT, CORRECT, CORRECT, CORRECT, CORRECT, CORRECT,
                        CORRECT, CORRECT, CORRECT, CORRECT, CORRECT, CORRECT, CORRECT,
                        CORRECT]
ML Accuracy: 93.75
Left:Right ratios:
 --- Predicted: 0:16 ---
 ---    Actual: 1:15 ---


   ---   female   ---
ML Correctness:      [CORRECT, incorrect, CORRECT, incorrect, incorrect, CORRECT, CORRECT, CORRECT,
                        CORRECT, incorrect, CORRECT, CORRECT, CORRECT, CORRECT, CORRECT,
                        CORRECT, CORRECT, CORRECT, CORRECT]
ML Accuracy: 78.95
Left:Right ratios:
 --- Predicted: 3:16 ---
 ---    Actual: 5:14 ---
```

Figure 4-5 Output 3 for Three VGG16 Models (GPU)

For ease of reading, the accuracies for each model have been graphed below in Figure 4-6. While the invariant nature of ‘everything’ and ‘uninjured’ would imply success was found with these models, both have been results of classifying every test sample as ‘Right’. As such, these two models can be removed from consideration. As before, they suffer from an overwhelming number of ‘Right’ over ‘Left’ samples being present in the population.

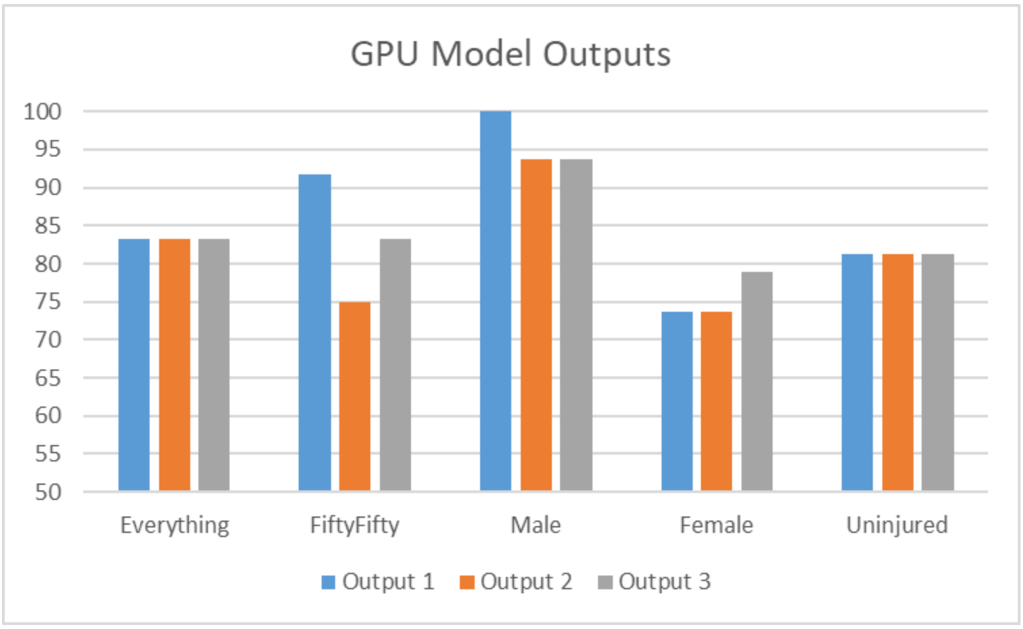


Figure 4-6 GPU Generated Model Output Graph

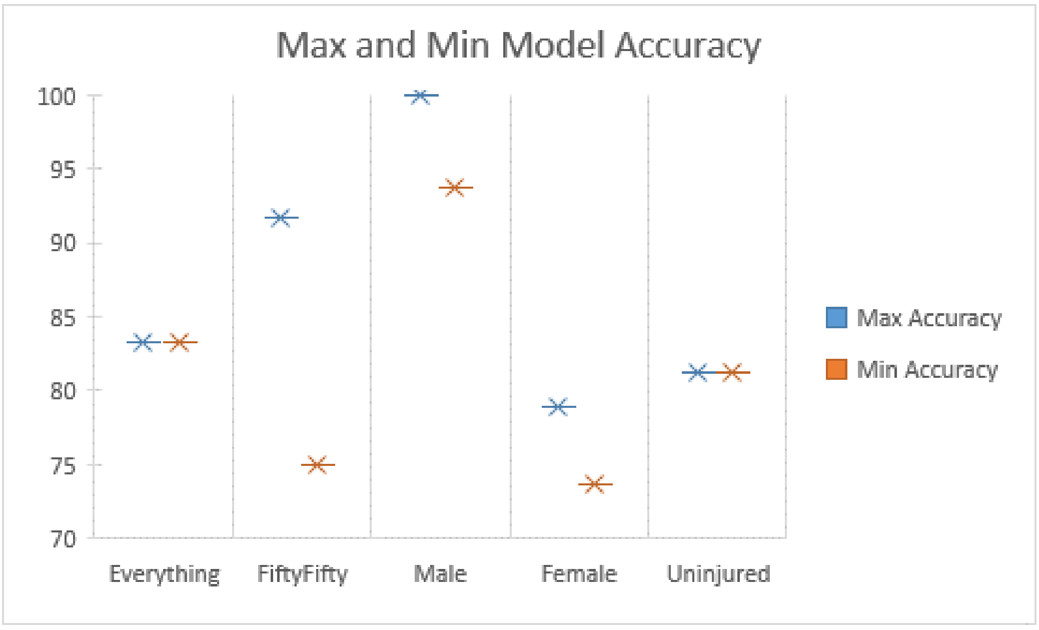


Figure 4-7 GPU Model Variance

Model	Max Accuracy	Min Accuracy	Spread	Mean Accuracy
Everything	83.33	83.33	0	83.33
FiftyFifty	91.67	75.00	16.67	83.33
Male	100.00	93.75	6.25	95.83
Female	78.95	73.68	5.27	75.44
Uninjured	81.25	81.25	0	81.25

Table 4-1 Model Max and Min Accuracy, with Spread

Figure 4-7 and Table 4-1 demonstrate the range of accuracies given by each model. Immediately 'Everything' and 'Uninjured' should be removed from consideration. Despite the 0 spread noted both models identified each sample as 'Right' and, as mentioned previously, this resulted in a false positive due to the small minority of the population that constituted left handed individuals.

Of immediate interest is the female only model. Despite its comparatively low mean accuracy, it also holds the lowest variance between runs. This suggests that it is the most consistently accurate of the models, and as such should be considered the most successful. This result is likely, in part, due to the female population providing the majority of the left-handed sample points.

21.74% of the female population surveyed were left-handed. This is in contrast to the 9.52% of the male population. The entire observed population had a 15.56% left-handed portion. It is difficult to draw a conclusion from the male model, as there was only one left-handed sample in the test data. However, it was able to make correct identifications across some runs.

Unfortunately, the 'Everything' model ran with no reliability in identification. The same can be said for the 'Uninjured' model, which presented 18.18% as left-handed. As such it should be concluded that, while some success was seen in a 9.52% population in the male model, that <=18% population being left-handed is not sufficient for accurate model training. From this data it should be concluded that

~21% is the minimum of the population being left-handed that is necessary for accurate model training.

The FiftyFifty model is of lesser interest than the female model, but remains important for multiple reasons. Firstly it was the first model to show signs of success for this research undertaking. The second reason for importance is that due to the first reason, it became a guiding factor in training the female, male, and uninjured models. This decision was made when the success was observed in utilising the FiftyFifty model.

Finally, despite the large spread of FiftyFifty, its minimum accuracy meets the accuracy threshold stated in the research question. This was observed in each training-testing run undertaken.

5. CONCLUSION

5.1 Research Overview

Over the course of the study, it was found that further research into the topic is necessary. An assumption of an SVM method being suitable for the task was made, which was later found not to be the case. Most of the above discussed results would suggest a success of H1, however much of this was the result of 'lucky guesswork' on the part of the further trained models, so this should not be taken as indicative of complete success.

The few positive results however, do present a solid base to work from in the future to show the success of the presented hypothesis H1.

Far from conclusive, it is advised that a further, more large scale study is necessary to claim this success definitively.

5.2 Problem Definition

A suitable problem definition seems to have been provided. As noted however, through the undertaking of this research it was found that a CNN model was more suited to the appointed task than an SVM model. This change from the posed problem has been mentioned through the methodology and discussion.

Given the relatively small sample size acquired, it is difficult to determine if the problem has been given a final answer. However, it is likely that that answer is that a highly accurate determination of a person's hand laterality can be made with the aid of a suitably informed machine learning system.

5.3 Design/Experimentation, Evaluation & Results

The experimental design provided a good method of data collection for this study. The shortcomings were found in the breadth of this collection. Ultimately the results discussed above show that the population surveyed was overwhelmingly biased towards right handed individuals. Despite this statement, the few positive results do show that the experimental design was good, if somewhat unaccommodating of such a minority sample size being presented to it.

5.4 Contributions and Impact

The impact of this study is a display of the foundations it has built. Through the few positive results obtained it has been indicated that there is much promise in producing a machine learning system that can correctly identify handedness through video alone.

The results of this study more so present the difficulties in this area of research. Unlike more sophisticated studies, it did not use additional measurement tools to provide data such as force of impact from footfalls, etc. This adds an additional layer of complexity to extrapolate more information from a reduced source of input data.

Moving forward it has been shown through this study that "walking left-handed" and "walking right-handed" are subtly, sufficiently different to be considered different activities for the purpose of activity identification. In and of itself, this is significant enough to inform future undertakers of this topic that differentiating the two is possible.

5.5 Future Work & Recommendations

If this study were to be undertaken again, additional base models should be added to find the most accurate resulting models for laterality determination. In an ideal scenario, a future study would be broad enough to produce the data to train an entire model without the need to perform transfer learning on a base model.

Regardless of training an entire model, a much larger sample population should be observed, in order to generate a more accurate resulting model. The above discussed results indicate that this is entirely feasible, provided enough data is collected for training and testing purposes.

If a larger dataset is still required, but a larger population cannot be procured, there are some methods to artificially increase the dataset by a factor of six. This increase is found as follows:

- The dataset itself (factor of one achieved)
- The dataset rotated by 90° (factor of two achieved)
- The dataset rotated by 180° (factor of three achieved)
- The above three points repeated with the dataset mirrored, and its label (left/right) inverted; flipped on the vertical axis (factor of six achieved)

The addition of a recurrent neural network (RNN) to the CNN utilised could also see an increase in accuracy. An RNN would allow for the consideration of the temporal element, which is unique to video over image processing. Should further processing be performed on the data gathered during this undertaking, the addition of an RNN would be the first suggested change in how this is conducted.

One of the major shortcomings of this study lies within the lack of a large-scale dataset to draw from. Should any further work be carried out on this topic, that is the first point of improvement. As such the following is a proposal for a follow up research study which should provide a more fairly spread dataset, as well as one of higher quality. The result of which should be an improvement on the results found here.

Should a future dataset be large enough, it could be worthwhile to train a model to differentiate between four activities; "walking right-handed male", "walking right-handed female", "walking left-handed male", "walking left-handed female". As seen previously, training models with just male, or just female participants proved to produce more accurate results. As such, it suggests that separating based on gender for each activity could create a more informed model.

5.5.1 Future Protocol Proposal

Data collection should be undertaken in a large venue. One which either has a lot of foot traffic, which would provide a more "random" population sample, or which could accommodate the necessarily large, required sample size. Based on the results from this study, of particular note the weightFiftyFifty result, the volunteer population should be at least equal to the number gathered here (45 count) which presented 180 video data points to work from, equating to 720 images formed into varying-length sequential sets. However, ideally this new set would be several times larger than the dataset gathered here to allow for even greater accuracy.

This new population should exist with a split such that 40% is the minimum for either handedness expressed by the population, and as a result 60% would be the maximum allowance for the opposite handedness. Should ambidextrous individuals make up a significant portion of the population surveyed a third activity of "walking ambidextrous" should be considered. However, they should not be included in the analysis unless they make up at least 20% of the population. Even at such a considerable percentage, this section of the population is likely to be mis-identified; noting that from the female population result above, this percentage of the population was necessary for accurate identification in a two-class population.
A similar setup to that used for this study should be implemented. Ideally the small course walked by participants should be at least 4 meters long, with clearer (though non-distracting) markings used to denote the course. There were several walks that required re-recording as the volunteers failed to walk in successive straight lines, and drifted towards the camera.

For the camera setup; the recording area would ideally be very well lit, with sufficient lighting to render volunteers' shadows a negligible change. This would result in clearer, cleaner silhouettes extracted from the raw footage.

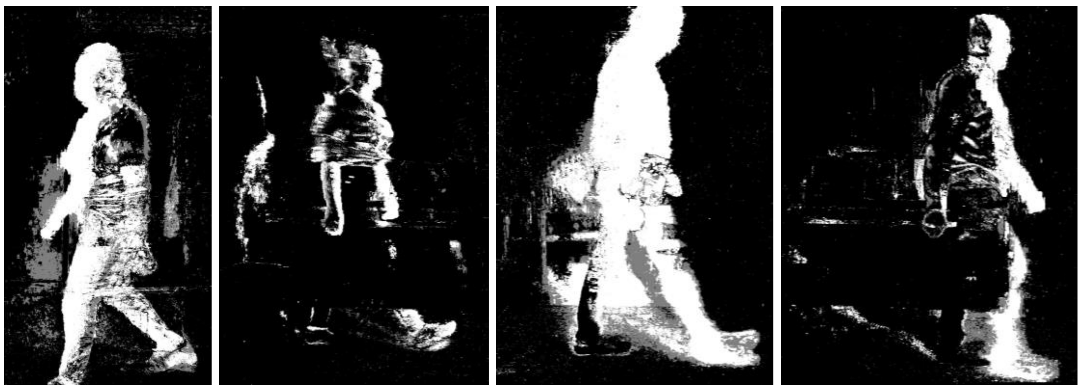


Figure 5-1 Samples from office

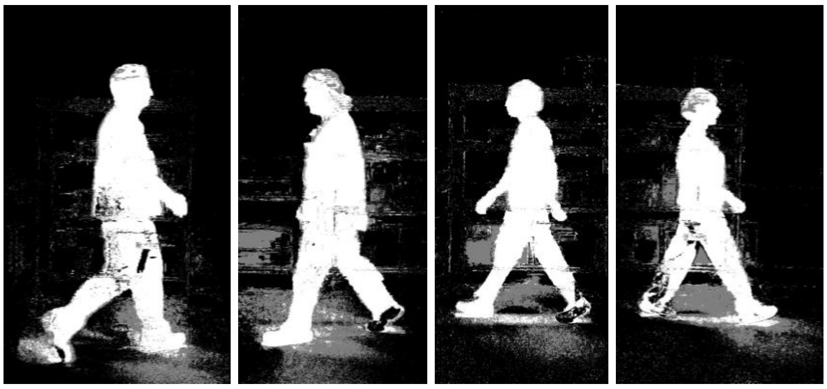


Figure 5-2 Samples from hospital

Above are selective, but indicative, samples from both the office and hospital used for data collection. As can be seen, the inferior office lighting led to ghostly images being extracted due to changing shadows. This loses some of the finer points of the videos captured, and is why good lighting is necessary for more precise results.

Once the future data collection is completed, it would be ideal if there were enough samples present to train a machine learning model from the ground up. This would likely result in a much more accurate, if also much more restrictive, resulting model. Typically this would require several thousand data points for training and testing purposes. Based on this study producing a maximum of 720 images, for the weighteverything model, from 45 participants, 200 participants would likely be the minimum needed to produce a suitably accurate model. This also assumes a generally even mix of left and right handed individuals.

Failing enough data to produce an entirely self-trained model, or finding that the resulting model is inaccurate, the remainder of this future study could follow the remainder of the undertaken research. Using transfer learning to further educate pre-existing models. With the goal of achieving an accuracy equal to, or greater than, that found here, across a much larger sample size.

A successful future study would see a result equivalent to that produced by the weighteverything model, though without that result arising from labelling all test data as “Right” with a very low “Left” population.

5.6 Summary

While this study did not result in success in every action undertaken, it has shown that the desired outcome to the research objective is achievable. It is possible to determine a person’s handedness through video analysis techniques, however much more data is required to build a consistently accurate model which can achieve this outcome across all cases. The key missing points of data observed were males who are left handed. As such, these were among the most inaccurately classified group.

BIBLIOGRAPHY

Abdulhay, E., Arunkumar, N., Narasimhan, K., Vellaiappan, E., & Venkatraman, V. (2018). Gait and tremor investigation using machine learning techniques for the diagnosis of Parkinsons disease. *Future Generation Computer Systems*, 366-373.

Abellanas, A., Frizera, A., Ceres, R., & Raya, R. (2009). Assessment of the laterality effects through forearm reaction forces in walker assisted gait. *Procedia Chemistry*, 1227-1230.

Agarwal, A., & Triggs, B. (2004). Learning to track 3D human motion from silhouettes. *Twenty-first international conference on machine learning.* ICML.

Aqib Haqmi Abas, M., Ismail, N., Ihsan Mohd Yassin, A., & Nasir Taib, M. (2018). VGG16 for Plant Image Classification with Transfer Learning and Data Augmentation. *International Journal of Engineering & Technology*, 90-94.

Avery, J., Vogel, D., Lank, E., Masson, D., & Hanae, R. (2019). Holding Patterns: Detecting Handedness With A Moving Smartphone At Pickup. *IHM'19.* Grenoble, France: ACM.

Awad, M., & Motai, Y. (2008). Dynamic classification for video stream using support vector machine. *Applied Soft Computing*, 1314-1325.

Bardikoff, N., & McGonigle-Chalmers, M. (2014). Testing nonverbal IQ in children with Autism Spectrum Disorders. *Research in Autism Spectrum Disorders*, 1200-1207.

Barnich, O., Jodogne, S., & Van Droogenbroeck, M. (2006). Robust Analysis of Silhouettes by Morphological Size Distributions. *Lecture Notes in Computer Science* (pp. 734-745). Belgium: Springer-Verlag Berlin Heidelberg.

Begg, R., & Kamruzzaman, J. (2005). A machine learning approach for automated recognition of movement patterns using basic, kinetic and kinematic gait data. *Journal of Biomechanics*, 401-408.

Begg, R., Palaniswami, M., & Owen, B. (2005). Support Vector Machines for Automated Gait Classification. *IEEE Transactions on Biomedical Engineering*, 828-838.

Brereton, R., & Lloyd, G. (2010). Support Vector Machines for classification and regression. *The Analyst*, 230-267.

Chen, Y., & Xue, Y. (2015). A Deep Learning Approach to Human Activity Recognition Based on Single Accelerometer. *2015 IEEE International Conference on Systems, Man, and Cybernetics.* Kowloon, China: IEEE.

Corballis, M., & Morgan, M. (1978). On the biological basis of human laterality: Evidence for a maturational left-right gradient. *Behavioral and Brain Sciences*, 261-269.

Dadashi, F., Araabi, B., & Soltanian-Zadeh, H. (2009). Gait Recognition Using Wavelet Packet Silhouette Representation and Transductive Support Vector Machines. *2nd International Congress on Image and Signal Processing.*

Dai, W., Yang, Q., Xue, G.-R., & Yu, Y. (2007). Boosting for Transfer Learning. *ICML '07: Proceedings of the 24th international conference on Machine learning* (pp. 193-200). ICML.

Davis, R., Ounpuu, S., Tyburski, D., & Gage, J. (1991). A gait analysis data collection and reduction technique. *Human Movement Science*, 575-587.

Franz, E., Rowse, A., & Ballantine, B. (2002). Does Handedness Determine Which Hand Leads in a Bimanual Task? *Journal of Motor Behavior,* 402-412.

Gopalakrishnan, K., K. Khaitan, S. C., & Agrawal, A. (2017, December). Deep Convolutional Neural Networks with transfer learning for computer vision-based data-driven pavement distress detection. *Construction and Building Materials*, 322-330.

Hu, C., Yu, Q., Li, Y., & Ma, S. (2002). Extraction of parametric human model for posture recognition using genetic algorithm. *Fourth IEEE International Conference on Automatic Face and Gesture Recognition.* IEEE.

Ignatov, A. (2018, Janaury). Real-time human activity recognition from accelerometer data using Convolutional Neural Networks. *Applied Soft Computing, 62*, 915-922.

Ishikawa, K., Edo, M., Yokomizo, M., Terada, N., Okamoto, Y., & Togawa, K. (1994). Analysis of Gait in Patients with Peripheral Vestibular Disorders. *ORL*, 325-330.

Jayadeva, Khemchandani, R., & Chandra, S. (2007). Twin Support Vector Machines for Pattern Classification. *IEEE Transactions on Pattern Analysis and Machine Inteligence*, 905-910.

Krizhevsky, A., Sutskever, I., & E. Hinton, G. (2012). ImageNet Classification with Deep Convolutional Neural Networks. *Advances in neural information processing systems*.

Liu, B., Zhang, X., Gao, Z., & Chen, L. (2018). Weld Defect Images Classification with VGG16-Based Neural Network. *Digital TV and Wireless Multimedia Communication*, 215-223.

Löchtefeld, M., Schardt, P., & Krüger, A. (2015). Detecting Users Handedness for Ergonomic Adaptation of Mobile User Interfaces. *The 14th International Conference on Mobile and Ubiquitous Multimedia* (pp. 245-249). Linz, Austria: ACM.

Malinverni, L., Mora-Guiard, J., Padillo, V., Valero, L., Hervás, A., & Pares, N. (2017). An inclusive design approach for developing video games for children with Autism Spectrum Disorder. *Computers in Human Behavior*, 535-549.

Maubert Crotte, A., H. Hepting, D., & Roshchina, A. (2019). Left-Handed Control Configuration for Side-Scrolling Games. *CHI 2019*. Glasgow, Scotland: ACM.

Maupas, E., Paysant, Datie, A., Martinet, N., & André, J. (2002). Functional asymmetries of the lower limbs. A comparison between clinical assessment of laterality, isokinetic evaluation and electrogoniometric monitoring of knees during walking. *Gait & Posture*, 304-312.

Nasreen, A., Vinutha, H., & Shobha, G. (2017). Analysis of Video Content Through Object Search Using SVM Classifier. *Lecture Notes in Networks and Systems*, 325-333.

O. Mazurek, M., R. Engelhardt, C., & E. Clark, K. (2015). Video games from the perspective of adults with autism spectrum disorder. *Computer in Human Behavior*, 122-130.

Olsson, H., & Ingvar, C. (1991). Left handedness is uncommon in breast cancer patients. *European Journal of Cancer and Clinical Oncology*, 1694-1695.

Pan, S. J., & Yang, Q. (2010). A Survey on Transfer Learning. *IEEE Transactions on Knowledge and Data Engineering*, 1345-1359.

Patten, E., K. Ausderau, K., R. Watson, L., & T. Baranek, G. (2013). Sensory Response Patterns in Nonverbal Children with ASD. *Autism Research and Treatment*.

Perelle, I., & Ehrman, L. (1994). An international study of human handedness: The data. *Behavior Genetics*, 217-227.

Porac, C., Coren, S., Steiger, J., & Duncan, P. (1980). Human laterality: A multidimensional approach. *Canadian Journal of Psychology/Revue canadienne de psychologie*, 91-96.

Rezende, E., Ruppert, G., Carvalho, T., Theophilo, A., Ramos, F., & de Geus, P. (2018). Malicious Software Classification Using VGG16 Deep Neural Network's Bottleneck Features. *Information Technology - New Generations* (pp. 51-59). Springer.

Sadeghi, H., Allard, P., Prince, F., & Labelle, H. (2000). Symmetry and limb dominance in able-bodied gait: a review. *Gait & Posture*, 34-45.

Sadlier, D., & O'Connor, N. (2005). Event detection in field sports video using audio-visual features and a support vector machine. *IEEE Transactions on Circuits and Systems for Video Technology*, 1225-1233.

Sidenbladh, H. (2004). Detecting human motion with support vector machines. *Proceedings of the 17th International Conference on Pattern Recognition.* ICPR.

Simonyan, K., & Zisserman, A. (2015). Very Deep Convolutional Networks for Large-Scale Image Recognition. *ICLR 2015.* San Diego: ICLR.

Singh, D., Merdivan, E., Hanke, S., Kropf, J., Geist, M., & Holzinger, A. (2017). Convolutional and Recurrent Neural Networks for Activity Recognition in Smart Environment. *Towards Integrative Machine Learning and Knowledge Extraction*, 194-205.

Su, H., & Huang, F.-G. (2005). Human gait recognition based on motion analysis. *International Conference on Machine Learning and Cybernetics.*

Suresh, V., Mohan, C., Swamy, R., & Yegnanarayana, B. (2004). Content-Based Video Classification Using Support Vector Machines. *Department of Computer Science and Engineering*, (pp. 726-731). Chennai.

Taylor, P., Dalton, R., & Fleminger, J. (1980). Handedness in Schizophrenia. *British Journal of Psychiatry*, 375-383.

Wang, J., Chen, Y., Hao, S., Peng, X., & Hu, L. (2019, March). Deep learning for sensor-based activity recognition: A survey. *Pattern Recognition Letters, 119,* 3-11.

Wang, L., & Suter, D. (2006). Analyzing Human Movements from Silhouettes Using Manifold Learning. *International Conference on Video and Signal Based Surveillance.* IEEE.

Yamada, M., Ebihara, K., & Ohya, J. (2002). A new robust real-time method for extracting human silhouettes from color images. *Third IEEE International Conference on Automatic Face and Gesture Recognition.* IEEE.

Yang, M., Zheng, H., Wang, H., McClean, S., Hall, J., & Harris, N. (2012). A machine learning approach to assessing gait patterns for Compex Regional Pain Syndrome. *Medical Engineering & Physics*, 740-746.

Yoo, J., Hwang, D., & Nixon, M. (2005). Gender Classification in Human Gait Using Support Vector Machine. *Lecture Notes in Computer Science* (pp. 138-145). Berlin: ACIVS.

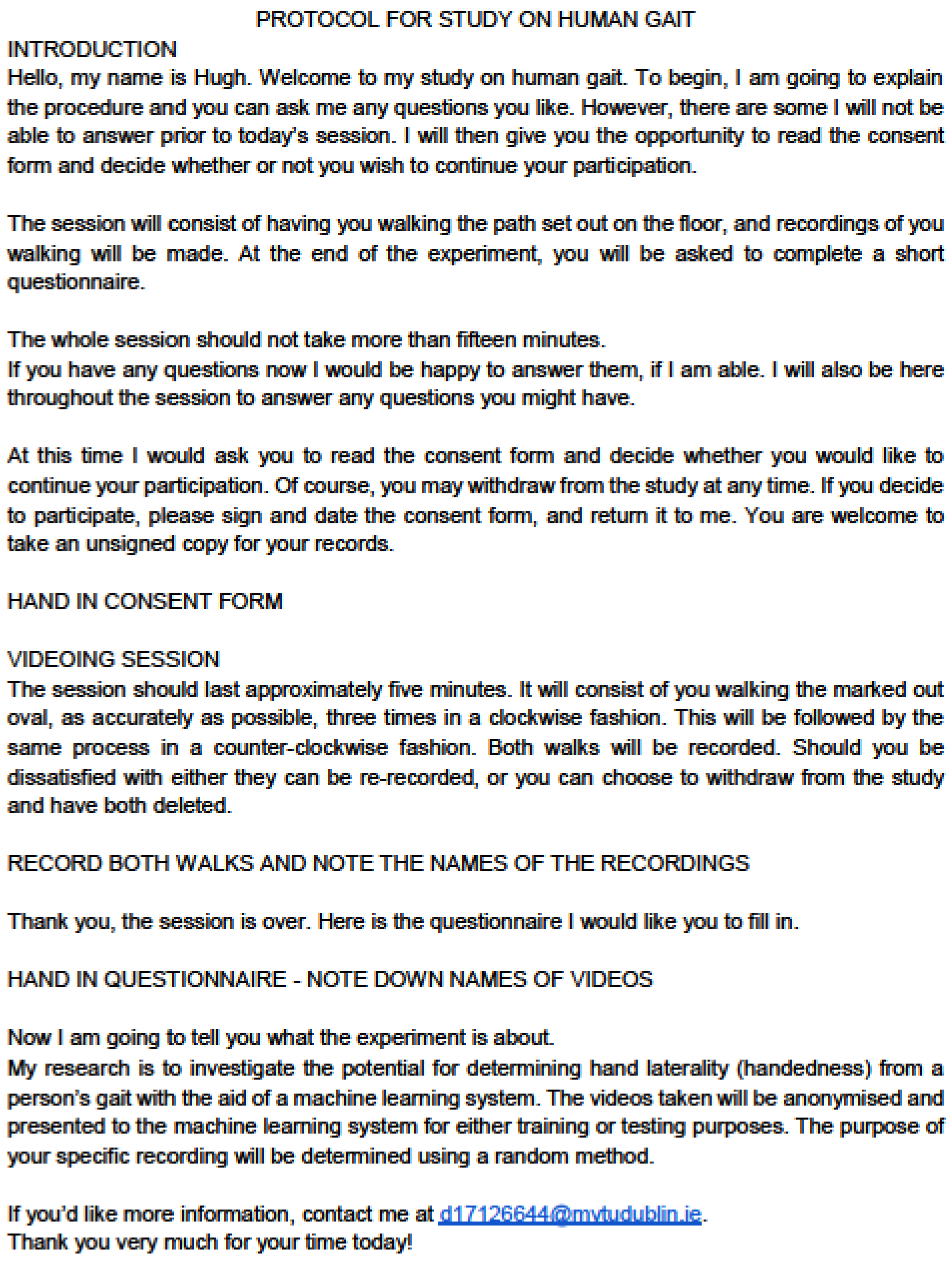
Yu, W., Yang, K., Bai, Y., Xiao, T., Yao, H., & Rui, Y. (2016). Visualizing and Comparing AlexNet and VGG using Deconvolutional Layers. *Proceedings of the 33 rd International Conference on Machine.* New York: JMLR.

Zhu, G., Xu, C., Huang, Q., & Gao, W. (2006). Automatic Multi-Player Detection and Tracking in Broadcast Sports Video using Support Vector Machine and Particle Filter. *IEEE International Conference on Multimedia and Expo.* IEEE.

Zverev, Y. (2006). Spatial parameters of walking gait and footedness. *Annals of Human Biology*, 161-176.

APPENDIX A – DOCUMENTS

Protocol Document



PROTOCOL FOR STUDY ON HUMAN GAIT

INTRODUCTION

Hello, my name is Hugh. Welcome to my study on human gait. To begin, I am going to explain the procedure and you can ask me any questions you like. However, there are some I will not be able to answer prior to today's session. I will then give you the opportunity to read the consent form and decide whether or not you wish to continue your participation.

The session will consist of having you walking the path set out on the floor, and recordings of you walking will be made. At the end of the experiment, you will be asked to complete a short questionnaire.

The whole session should not take more than fifteen minutes.
If you have any questions now I would be happy to answer them, if I am able. I will also be here throughout the session to answer any questions you might have.

At this time I would ask you to read the consent form and decide whether you would like to continue your participation. Of course, you may withdraw from the study at any time. If you decide to participate, please sign and date the consent form, and return it to me. You are welcome to take an unsigned copy for your records.

HAND IN CONSENT FORM

VIDEOING SESSION

The session should last approximately five minutes. It will consist of you walking the marked out oval, as accurately as possible, three times in a clockwise fashion. This will be followed by the same process in a counter-clockwise fashion. Both walks will be recorded. Should you be dissatisfied with either they can be re-recorded, or you can choose to withdraw from the study and have both deleted.

RECORD BOTH WALKS AND NOTE THE NAMES OF THE RECORDINGS

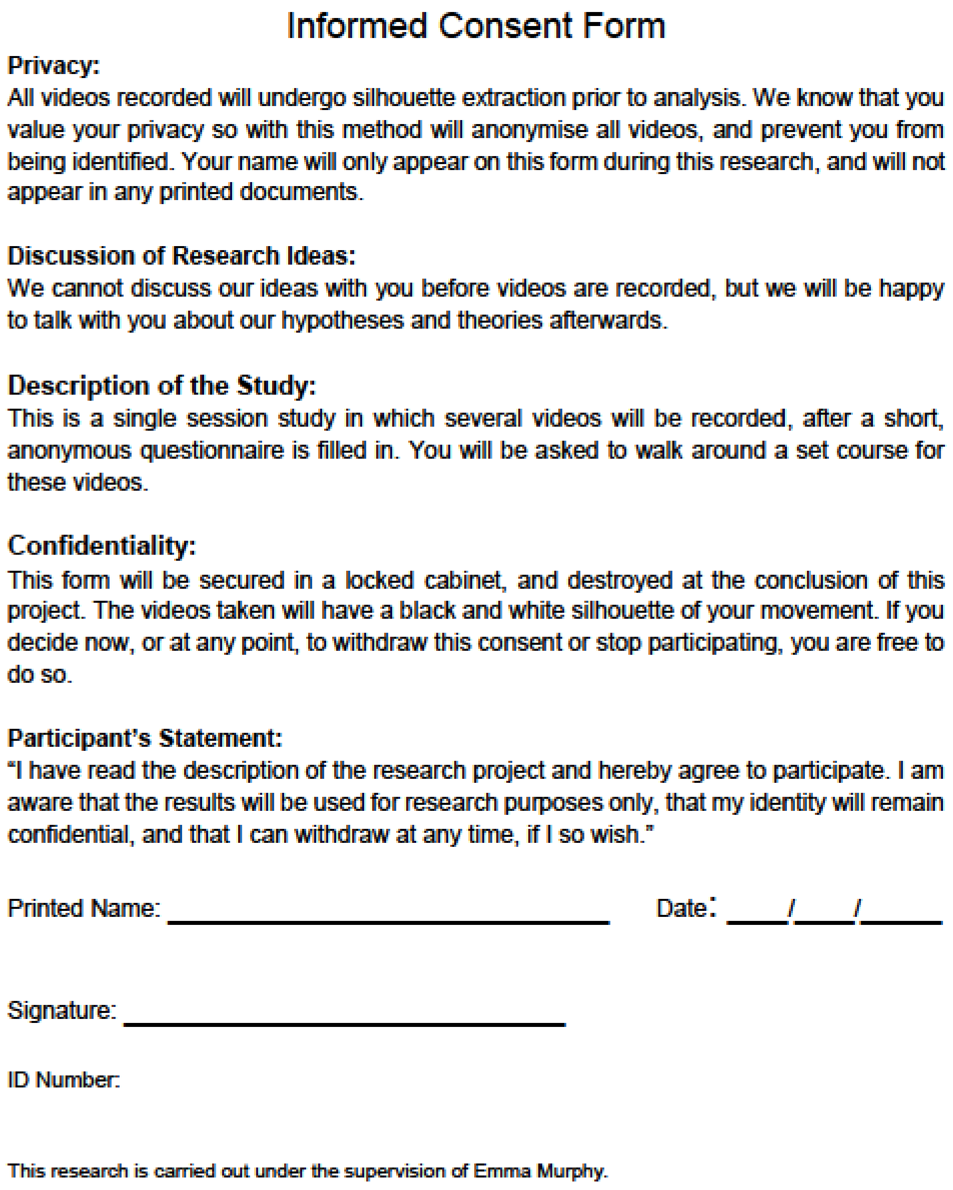
Thank you, the session is over. Here is the questionnaire I would like you to fill in.

HAND IN QUESTIONNAIRE - NOTE DOWN NAMES OF VIDEOS

Now I am going to tell you what the experiment is about.
My research is to investigate the potential for determining hand laterality (handedness) from a person's gait with the aid of a machine learning system. The videos taken will be anonymised and presented to the machine learning system for either training or testing purposes. The purpose of your specific recording will be determined using a random method.

If you'd like more information, contact me at d17126644@mytudublin.ie.
Thank you very much for your time today!

Informed Consent Form



Informed Consent Form

Privacy:
All videos recorded will undergo silhouette extraction prior to analysis. We know that you value your privacy so with this method will anonymise all videos, and prevent you from being identified. Your name will only appear on this form during this research, and will not appear in any printed documents.

Discussion of Research Ideas:
We cannot discuss our ideas with you before videos are recorded, but we will be happy to talk with you about our hypotheses and theories afterwards.

Description of the Study:
This is a single session study in which several videos will be recorded, after a short, anonymous questionnaire is filled in. You will be asked to walk around a set course for these videos.

Confidentiality:
This form will be secured in a locked cabinet, and destroyed at the conclusion of this project. The videos taken will have a black and white silhouette of your movement. If you decide now, or at any point, to withdraw this consent or stop participating, you are free to do so.

Participant's Statement:
"I have read the description of the research project and hereby agree to participate. I am aware that the results will be used for research purposes only, that my identity will remain confidential, and that I can withdraw at any time, if I so wish."

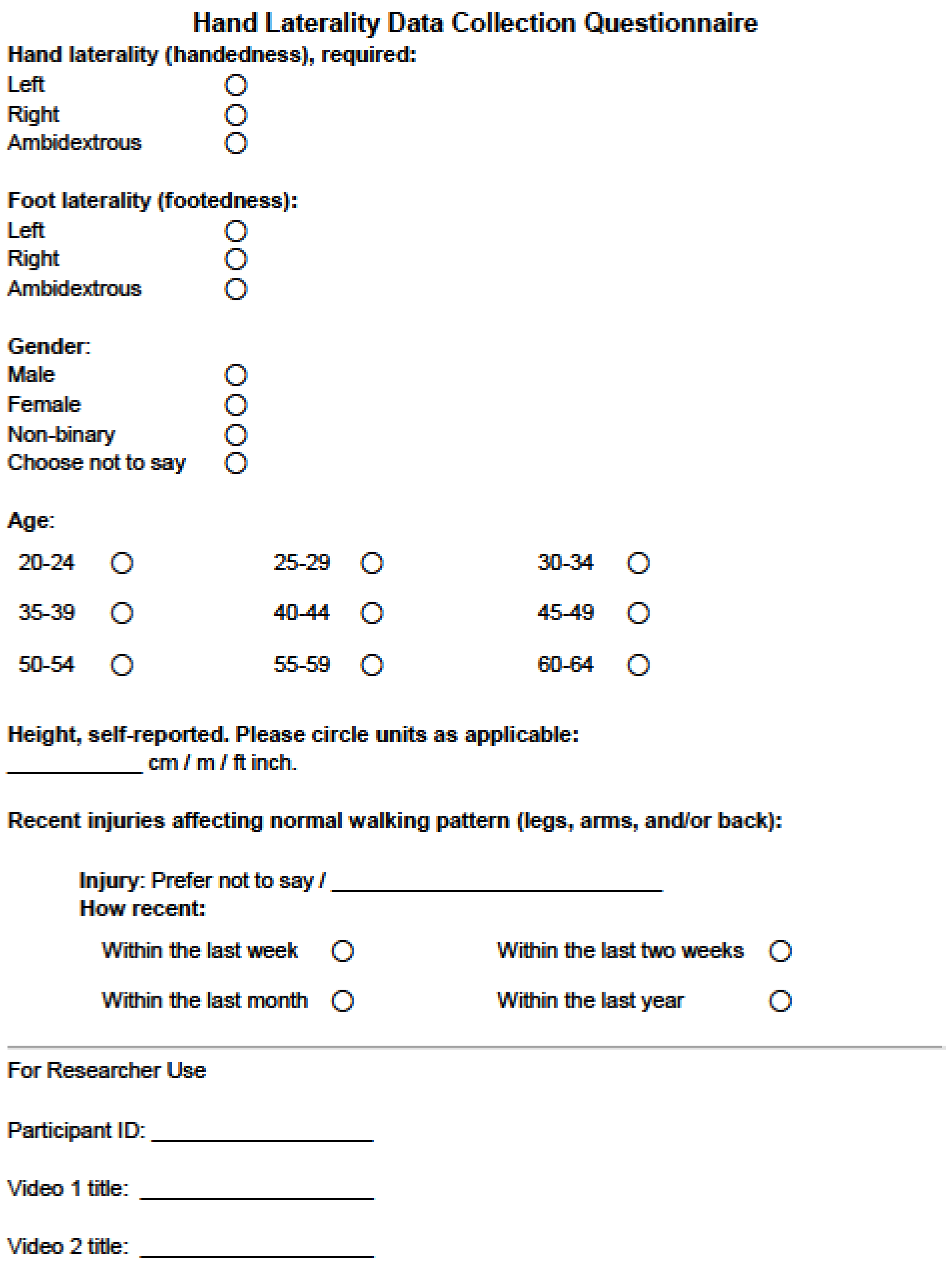
Printed Name: ________________________ Date: ____/____/_____

Signature: ________________________

ID Number:

This research is carried out under the supervision of Emma Murphy.

Questionnaire



Hand Laterality Data Collection Questionnaire

Hand laterality (handedness), required:

Left	○
Right	○
Ambidextrous	○

Foot laterality (footedness):

Left	○
Right	○
Ambidextrous	○

Gender:

Male	○
Female	○
Non-binary	○
Choose not to say	○

Age:

20-24	○	25-29	○	30-34	○
35-39	○	40-44	○	45-49	○
50-54	○	55-59	○	60-64	○

Height, self-reported. Please circle units as applicable:
___________ cm / m / ft inch.

Recent injuries affecting normal walking pattern (legs, arms, and/or back):

Injury: Prefer not to say / ___________________________
How recent:

Within the last week	○	Within the last two weeks	○
Within the last month	○	Within the last year	○

For Researcher Use

Participant ID: __________________

Video 1 title: __________________

Video 2 title: __________________

APPENDIX B - CODE

Silhouette Extraction Code

```python
from __future__ import print_function
import cv2 as cv
import argparse
import os
import pathlib
import csv
from tqdm import tqdm

hand_dict = {}
gender_dict = {}
injury_dict = {}
with open('video_details.csv') as csvfile:
    detail_grabber = csv.reader(csvfile)
    headers = next(detail_grabber)
    for row in detail_grabber:
        hand_dict[(row[0]+'.mp4')] = row[1]
        gender_dict[(row[0]+'.mp4')] = row[2]
        injury_dict[(row[0]+'.mp4')] = row[3]
fiftyfiftysplit = True
train_list=[]
test_list=[]
train_list_hand=[]
test_list_hand=[]
train_list_male=[]
test_list_male=[]
train_list_female=[]
test_list_female=[]
train_list_injury=[]
test_list_injury=[]
count = 0
```

```
left_count = 0
right_count = 0
for root,dirs,filename in os.walk('RawFootage/'):
    for file in filename:
        filepath = os.path.join(root,file)
        filepath = filepath.replace('\\', '/')
        if 'Left' in filepath and fiftyfiftysplit:
            left_count += 1
        elif 'Right' in filepath and fiftyfiftysplit:
            right_count += 1
        if filepath.endswith(".mp4"):
            parser = argparse.ArgumentParser(description= 'This program shows how to use background subtraction methods provided by OpenCV. You can process both videos and images.')
            raw_path = os.path.join(root, file)
            print("Raw Path: "+raw_path)
            tag_and_name = str(root[len('RawFootage/')::])+ '/SIL'+file[3:-4]+'.avi'
            if count % 5 == 0:
                test_list += [tag_and_name]
                if right_count <= left_count:
                    test_list_hand += [tag_and_name]
                if gender_dict[file] == 'M':
                    test_list_male += [tag_and_name]
                elif gender_dict[file] == 'F':
                    test_list_female += [tag_and_name]
                if injury_dict[file] == 'no':
                    test_list_injury += [tag_and_name]

            else:
                train_list += [tag_and_name+' 1']
                if right_count <= left_count:
                    train_list_hand += [tag_and_name+' 1']
                if gender_dict[file] == 'M':
                    train_list_male += [tag_and_name]
                elif gender_dict[file] == 'F':
                    train_list_female += [tag_and_name]
                if injury_dict[file] == 'no':
                    train_list_injury += [tag_and_name]

            count += 1
            output_name = 'machine-learning-training/videos/SIL'+file[3:-4]+'.avi'
            backup_output_name = "Extracted/"+tag_and_name
            print("Extracted Path: "+output_name)
            print("Extracted Path Backup: "+backup_output_name)
            parser.add_argument('--input', type=str, help='Path to a video or a sequence of image.', default=raw_path)
            parser.add_argument('--algo', type=str, help='Background subtraction method (KNN, MOG2).', default='KNN')
            args = parser.parse_args()
            if args.algo == 'KNN':
                backSub = cv.createBackgroundSubtractorMOG2()
            else:
                backSub = cv.createBackgroundSubtractorGMG()
            capture = cv.VideoCapture(cv.samples.findFileOrKeep(args.input))

            ret, frame = capture.read()
            fcount = int(capture.get(cv.CAP_PROP_FRAME_COUNT))
            fshape = frame.shape
            fheight = fshape[0]
            fwidth = fshape[1]
            print('Video height by width: '+str(fheight)+' '+str(fwidth))
            fourcc = cv.VideoWriter_fourcc(*'MJPG')
            fps = 20.0
            out = cv.VideoWriter(output_name,fourcc, fps,(fwidth,fheight), False)
            backup_out = cv.VideoWriter( backup_output_name,fourcc, fps,(fwidth,fheight), False)

            if not capture.isOpened:
                print('Unable to open: ' + args.input)
                exit(0)
            for i in tqdm(range(fcount-1), position=0, leave=True):
                ret, frame = capture.read()
                if frame is None:
                    print('Found None')
                        break
                    fgMask = backSub.apply(frame)
                    out.write(fgMask)
                    backup_out.write(fgMask)
                    keyboard = cv.waitKey(30)
                    if keyboard == 'q' or keyboard == 27:
                        break
                out.release()
                cv.destroyAllWindows()
                print('Done '+ file+ '\n')
            elif not filepath.endswith(".mp4"):
                print(filepath+" not a raw video")

    train_file = open("machine-learning-training/trainTest/trainlist1.txt", "w+")
    for file in train_list:
        train_file.write(file+'\n')
    train_file.close()

    train_file = open("machine-learning-training/trainTest/
  trainlist2.txt", "w+")
    for file in train_list_hand:
        train_file.write(file+'\n')
    train_file.close()

    train_file = open("machine-learning-training/trainTest/
  trainlist3.txt", "w+")
    for file in train_list_male:
        train_file.write(file+'\n')
    train_file.close()

    train_file = open("machine-learning-training/trainTest/
  trainlist4.txt", "w+")
    for file in train_list_female:
        train_file.write(file+'\n')
    train_file.close()

    train_file = open("machine-learning-training/trainTest/
  trainlist5.txt", "w+")
    for file in train_list_injury:
        train_file.write(file+'\n')
    train_file.close()


    test_file  =  open("machine-learning-training/trainTest/
  testlist1.txt", "w+")
    for file in test_list:
        test_file.write(file+'\n')
    test_file.close()
    test_file = open("machine-learning-training/trainTest/testlist2.txt", "w+")
    for file in test_list_hand:
        test_file.write(file+'\n')
    test_file.close()

    test_file = open("machine-learning-training/trainTest/testlist3.txt", "w+")
    for file in test_list_male:
        test_file.write(file+'\n')
    test_file.close()

    test_file = open("machine-learning-training/trainTest/testlist4.txt", "w+")
    for file in test_list_female:
        test_file.write(file+'\n')
    test_file.close()

    test_file = open("machine-learning-training/trainTest/testlist5.txt", "w+")
    for file in test_list_injury:
        test_file.write(file+'\n')
```

5.7 test_file.close()

Frame Extraction Code

```
import cv2      # for capturing videos
import math   # for mathematical operations
import matplotlib.pyplot as plt    # for plotting the images
import pandas as pd
import os
from keras.preprocessing import image   # for preprocessing the images
import numpy as np  # for mathematical operations
from keras.utils import np_utils
from skimage.transform import resize   # for resizing images
from sklearn.model_selection import train_test_split
from glob2 import glob
from tqdm import tqdm

def frame_extract(train_number):
     if train_number == 1:
          model_name = 'everything'
     elif train_number == 2:
          model_name = 'fiftyfifty'
     elif train_number == 3:
          model_name = 'male'
     elif train_number == 4:
          model_name = 'female'
     elif train_number == 5:
          model_name = 'uninjured'

     print('Extracting for '+model_name)
     # open the .txt file which have names of training videos
     f = open("trainTest/trainlist"+str(train_number)+".txt", "r")
     temp = f.read()
     videos = temp.split('\n')
```

```

    # creating a dataframe having video names
    train = pd.DataFrame()
    train['video_name'] = videos
    train = train[:-1]
    train.head()

    # creating tags for training videos
    train_video_tag = []
    for i in range(train.shape[0]):
        train_video_tag.append(train['video_name'][i].split('/')[0])
    train['tag'] = train_video_tag

    # storing the frames from training videos
    for i in tqdm(range(train.shape[0])):
        count = 0
        videoFile = train['video_name'][i]
        cap = cv2.VideoCapture('videos/'+videoFile.split(' ')[0].split('/')[1])   # capturing the video from the given path
        frameRate = cap.get(5) #frame rate
        x=1
        while(cap.isOpened()):
            frameId = cap.get(1) #current frame number
            ret, frame = cap.read()
            if (ret != True):
                break
            if (frameId % math.floor(frameRate) == 0):
                # storing the frames in a new folder named train_'+str(train_number)'
                filename ='train_'+str(train_number)+'/'+ videoFile.split('/')[0] + '_' + videoFile.split('/')[1].split(' ')[0] +"_frame%d.jpg" % count;count+=1
                cv2.imwrite(filename, frame)
        cap.release()

    # getting the names of all the images
    images = glob("train_"+str(train_number)+"/*.jpg")
    train_image = []
    train_class = []
    for i in tqdm(range(len(images))):
        try:
            train_image.append(images[i].split('\\')[1])
            train_class.append(images[i].split('\\')[1].split('_')[0])
        except IndexError:
            train_image.append(images[i].split('/')[1])
            train_class.append(images[i].split('/')[1].split('_')[0])

    # storing the images and their class in a dataframe
    train_data = pd.DataFrame()
    train_data['image'] = train_image
    train_data['class'] = train_class

    # converting the dataframe into csv file
    train_data.to_csv('UCF/train_new_'+model_name+'.csv',header=True, index=False)

def main():
    for i in range(1,6):
        frame_extract(i)

main()
```

Model Transfer Learning Training Code

```
import keras
from keras.models import Sequential
from keras.applications.vgg16 import VGG16
from keras.layers import Dense, InputLayer, Dropout, Flatten
from keras.layers import Conv2D, MaxPooling2D, GlobalMaxPooling2D
from keras.preprocessing import image
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from tqdm import tqdm
from sklearn.model_selection import train_test_split
from keras.callbacks import ModelCheckpoint

def train_model(model_name):
    print('Training model '+model_name)
    train = pd.read_csv('UCF/train_new_'+model_name+'.csv')
    train.head()

    # creating an empty list
    train_image = []

    # for loop to read and store frames
    for i in tqdm(range(train.shape[0])):
        # loading the image and keeping the target size as (224,224,3)
        img = image.load_img('train_1/'+train['image'][i], target_size=(224,224,3))
        # converting it to array
        img = image.img_to_array(img)
        # normalizing the pixel value
        img = img/255
        # appending the image to the train_image list
```

```python
        train_image.append(img)

# converting the list to numpy array
X = np.array(train_image)

# shape of the array
X.shape

# separating the target
y = train['class']

# creating the training and validation set
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=42, test_size=0.2, stratify = y)

# creating dummies of target variable for train and validation set
y_train = pd.get_dummies(y_train)
y_test = pd.get_dummies(y_test)

# creating the base model of pre-trained VGG16 model
base_model = VGG16(weights='imagenet', include_top =False)

# extracting features for training frames
X_train = base_model.predict(X_train)

# extracting features for validation frames
X_test = base_model.predict(X_test)

# reshaping the training as well as validation frames in single dimension
X_train = X_train.reshape(X_train.shape[0], X_train.shape[1]*X_train.shape[2]*X_train.shape[3])#(59075, 7*7*512)
```

```
    X_test = X_test.reshape(X_test.shape[0], X_test.shape[1]*X_test.shape[2]*X_test.shape[3])#(14769, 7*7*512)

    # normalizing the pixel values
    max = X_train.max()
    X_train = X_train/max
    X_test = X_test/max

    #defining the model architecture
    model = Sequential()
    model.add(Dense(1024, activation='relu', input_shape=(X_train.shape[1],)))
    model.add(Dropout(0.5))
    model.add(Dense(512, activation='relu'))
    model.add(Dropout(0.5))
    model.add(Dense(256, activation='relu'))
    model.add(Dropout(0.5))
    model.add(Dense(128, activation='relu'))
    model.add(Dropout(0.5))
    model.add(Dense(2, activation='softmax'))

    # defining a function to save the weights of best model
    mcp_save = ModelCheckpoint('weight'+model_name+'.hdf5', save_best_only=True, monitor='val_loss', mode='min')

    # compiling the model
    model.compile(loss='categorical_crossentropy',optimizer='Adam',metrics=['accuracy'])

    # training the model
    model.fit(X_train, y_train, epochs=200, validation_data=(X_test, y_test), callbacks=[mcp_save], batch_size=128, verbose=1)

```

```
def main():
    models = ['everything','fiftyfifty','male','female','uninjured']
    for model in models:
        train_model(model)

main()
```

Model Transfer Learning Testing Code

```
from keras.models import Sequential
from keras.layers import Dense, Dropout, Flatten
from keras.layers import Conv2D, MaxPooling2D
from keras.preprocessing import image
import numpy as np
import pandas as pd
from tqdm import tqdm
from keras.applications.vgg16 import VGG16
import cv2
import math
import os
from glob import glob
from scipy import stats as s

def test_model(model_name, train_number):
    print('\n\n --- Testing model: '+model_name+' ---')
    base_model = VGG16(weights='imagenet', include_top=
False)

    #defining the model architecture
    model = Sequential()
    model.add(Dense(1024, activation='relu', input_shape=
(25088,)))
    model.add(Dropout(0.5))
    model.add(Dense(512, activation='relu'))
    model.add(Dropout(0.5))
    model.add(Dense(256, activation='relu'))
    model.add(Dropout(0.5))
    model.add(Dense(128, activation='relu'))
    model.add(Dropout(0.5))
    model.add(Dense(2, activation='softmax'))

    # loading the trained weights
    model.load_weights("weight"+model_name+".hdf5")

    # compiling the model
    model.compile(loss='categorical_crossentropy',optimizer
='Adam',metrics=['accuracy'])

    # getting the test list
    f = open("trainTest/testlist"+str(train_number)+".txt",
"r")
    temp = f.read()
    videos = temp.split('\n')

    # creating the dataframe
    test = pd.DataFrame()
    test['video_name'] = videos
    test = test[:-1]
    test_videos = test['video_name']
    test.head()

    # creating the tags
    train = pd.read_csv('UCF/train_new_'+model_name+'.csv')
    y = train['class']
    y = pd.get_dummies(y)

    # creating two lists to store predicted and actual tags
    predict = []
    actual = []

    # for loop to extract frames from each test video
    for i in tqdm(range(test_videos.shape[0])):
        count = 0
        videoFile = test_videos[i]
        cap = cv2.VideoCapture('videos/'+videoFile.split(' ')[0].split('/')[1])   # capturing the video from the given path
        frameRate = cap.get(5) #frame rate
        x=1
        # removing all other files from the temp folder
        files = glob('temp/*')
        for f in files:
            os.remove(f)
        while(cap.isOpened()):
            frameId = cap.get(1) #current frame number
            ret, frame = cap.read()
            if (ret != True):
                break
            if (frameId % math.floor(frameRate) == 0):
                # storing the frames of this particular video in temp folder
                filename ='temp/' + "_frame%d.jpg" % count;count+=1
                cv2.imwrite(filename, frame)
        cap.release()

        # reading all the frames from temp folder
        images = glob("temp/*.jpg")

        prediction_images = []
        for i in range(len(images)):
            img = image.load_img(images[i], target_size=(224,224,3))
            img = image.img_to_array(img)
            img = img/255
            prediction_images.append(img)

```

```
    # converting all the frames for a test video into numpy array
    prediction_images = np.array(prediction_images)
    # extracting features using pre-trained model
    prediction_images = base_model.predict(prediction_images)
    # converting features in one dimensional array
    prediction_images = prediction_images.reshape(prediction_images.shape[0], 7*7*512)
    # predicting tags for each array
    prediction = model.predict_classes(prediction_images)
    # appending the mode of predictions in predict list to assign the tag to the video
    predict.append(y.columns.values[s.mode(prediction)[0][0]])
        # appending the actual tag of the video
        actual.append(videoFile.split('/')[0])

    # checking the accuracy of the predicted tags
    from sklearn.metrics import accuracy_score
    accuracy = accuracy_score(predict, actual)*100
    print('--- '+model_name+' ---')
    print('Prediction: ', end='')
    print(predict)
    print('Actual: ', end='')
    print(actual)
    print('Accuracy: '+str(accuracy))
    return predict, actual, accuracy


def main():
    models = ['everything','fiftyfifty','male','female','uninjured']
```

```
        predict_dict = {}
        actual_dict={}
        accuracy_dict={}
        for i in range(len(models)):
            predict_dict[models[i]], actual_dict[models[i]], accuracy_dict[models[i]] = test_model(models[i], i+1)
        for model in models:

            print('   ---   '+model+'   ---')
            print('ML Correctness:      [', end='')
            for i in range(len(predict_dict[model])):
                if predict_dict[model][i] == actual_dict[model][i]:
                    print('CORRECT',end='')
                else:
                    print('incorrect',end='')
                if i == len(predict_dict[model])-1:
                    print(']')
                    break
                else:
                    print(', ',end='')
                if i % 7 == 0 and i != 0:
                    print('\n ',end='')
            print('ML Accuracy: '+str(round(accuracy_dict[model],2))+'\n\n')
            print('Left:Right ratios: ')
            print('  ---  Predicted:  '+str(predict_dict[model].count('Left'))+':'+str(predict_dict[model].count('Right'))+' ---')
            print('  ---       Actual:  '+str(actual_dict[model].count('Left'))+':'+str(actual_dict[model].count('Right'))+' ---')
```

```

    main()
```