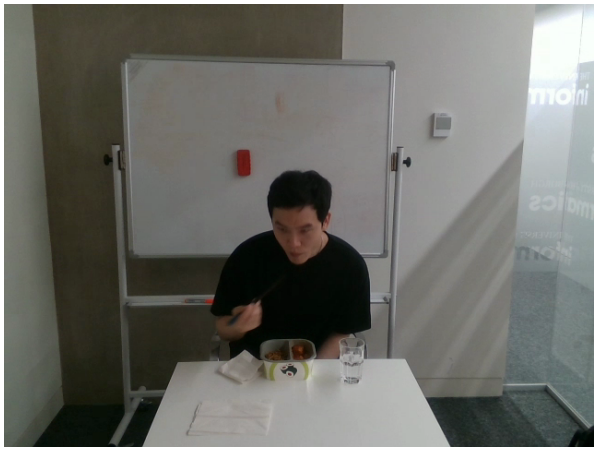
Face Annonymization and Recognising Eating Actions

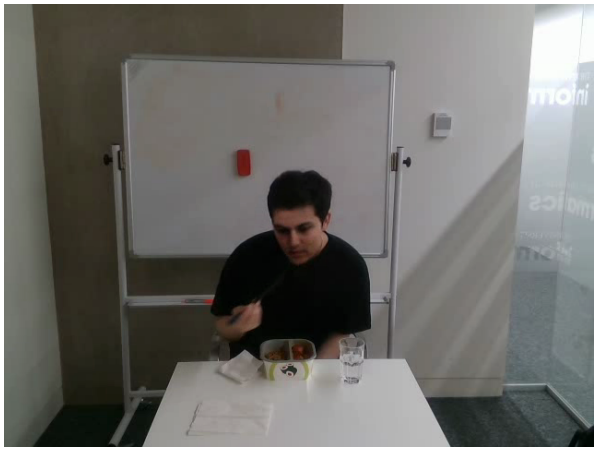
Minseong Kim

A Dissertation
Submitted to the School of Informatics
Supervised by Prof Robert B Fisher

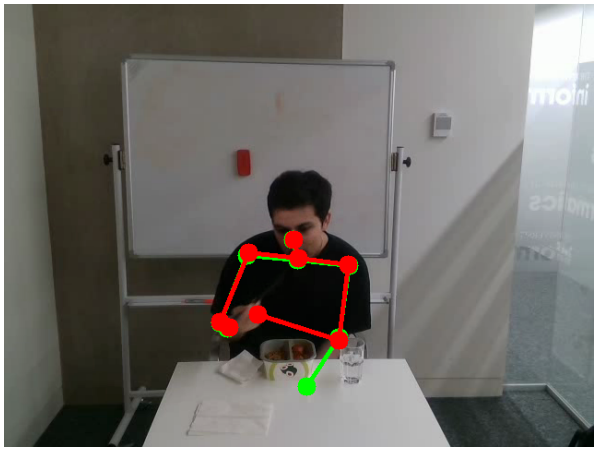
School of Informatics
University of Edinburgh
April 2023



(a) Original image



(b) Deepfaked image



(c) Pose comparison between original and deepfaked images

Figure 1

Abstract

Artificial Intelligence (A.I.) has improved dramatically in recent years, and it has become common to encounter A.I in everyday life, where it can serve to increase human well-being. This includes the healthcare industry. In this dissertation, we investigate the possibility of improving elders' well-being by adopting a computer vision-based surveillance camera system. This system will use action-based and joint-motion analysis-based algorithms, such as Temporal-Adaptive-Module (TAM), to monitor the motor activity of elderly people to determine whether they are still capable of living independently. We aimed to train the algorithms to recognise eating gestures in videos or photos so that they could be used to detect motor decline in the elderly. We created a custom dataset by filming eating motions with and without wrist weights to simulate the physical strength of an elderly person. We anonymized the custom dataset using DeepPrivacy2 to enable us to share our dataset with other researchers. To assess how effectively and accurately models generalise and classify activities, we used cross validation. We assessed the anonymized dataset by comparing it to the original dataset, ensuring its usability and reliability for other researchers. As a result, we found the anonymized dataset demonstrated reasonable accuracy and quality for action recognition tasks and has the potential to be used by other researchers in the field.

Contents

Chapter 1

Introduction

1.1 Motivation

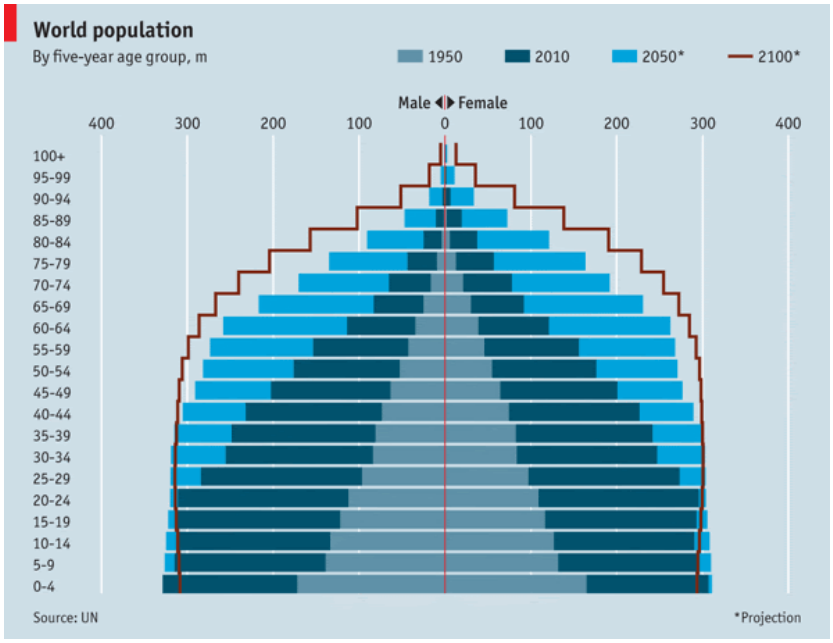


Figure 1.1: World Population Distribution[1]

The population demographics in most developed countries usually demonstrate a reversed pyramid shape, indicating a higher proportion of older individuals than younger individuals. This pattern is predicted to continue due to a combination of factors, including declining mortality rates and increased life expectancy in developing countries, as shown in Figure 1.1. Consequently, there will be more people in need of care than those who can provide care. In addition, training and supplying healthcare professionals is not a simple task. Thus, society must take proactive measures to prevent overwhelming societal resources with eldercare demands.

Machine learning has made significant advancements in recent years, affecting various industries both positively and negatively. Like previous industrial revolutions, it is undeniable that machine learning has the potential to automate many tasks, freeing hu-

mans from simple, repetitive labor. As a result, workers can focus on more creative and complex jobs that require characteristically human abilities, such as problem-solving and decision-making. Additionally, it appears that machine learning could offer solutions to the challenges in the healthcare sector.

With increasing age people face a higher risk of developing a large number of diseases. One example is osteoporosis, where the body generates too little new bone or loses too much bone, or both. Elderly individuals are also more susceptible to falls due to factors such as vision loss, balance problems, dementia, and heart disease. Coupled with osteoporosis, this can result in severe injuries. These diseases highlight why elders require more comprehensive care, but finding full-time caregivers and affording the additional cost can be challenging. Moreover, patients may resist moving to care homes due to the stress of being in a new and unfamiliar environment. However, many of these challenges can be overcome if elders can continue living in their homes with additional monitoring and timely detection of any significant changes in their pulse and posture. Thus, implementing camera-based action recognition technology to detect changes in their daily activities would significantly reduce the amount of trained personnel and cost typically required for monitoring them.

Although machine learning is a rapidly evolving field, finding appropriate datasets for training and testing models can still be challenging due to limited availability and a lack of standardization. However, the quality and quantity of the dataset can have a significant impact on the accuracy and effectiveness of the resulting model in machine learning. Having a high-quality dataset is important for several reasons: Improving model accuracy, generalization, avoiding bias, and speeding up development. In addition, a well-designed dataset can also save time and resources in the development process.

Due to this importance of using suitable datasets and the difficulties in obtaining them, we decided to curate our own datasets specifically for eating actions and subsequently shared them with the wider machine learning community as a contribution. Nevertheless, we have encountered several challenges, including concerns related to privacy. Since the datasets include images and videos of participants, including their faces while eating, determining their identities can be relatively straightforward, which may lead to significant legal consequences. One example of a machine learning dataset leak occurred in 2019, when a dataset containing facial recognition data was leaked online. The dataset, which was compiled by a company called Clearview AI, contained billions of images of individuals from social media and other online sources, along with facial recognition data that could be used to identify them. Clearview AI Inc has been fined £7,552,800 by the Information Commissioner's Office (ICO) for creating a worldwide online database for facial recognition by using images of individuals from the UK and other locations.[2].

Thus, it is necessary to safeguard the privacy of individuals captured in videos and images, which can be achieved through a process called anonymization: This is the process of removing personally identifiable information (PII) from datasets, thereby preventing the identification of individuals whose data is being used. Overall, anonymiza-

tion is an important tool for protecting privacy, preventing discrimination, improving data security, and complying with legal and ethical requirements when sharing personal data.

1.2 Research Questions

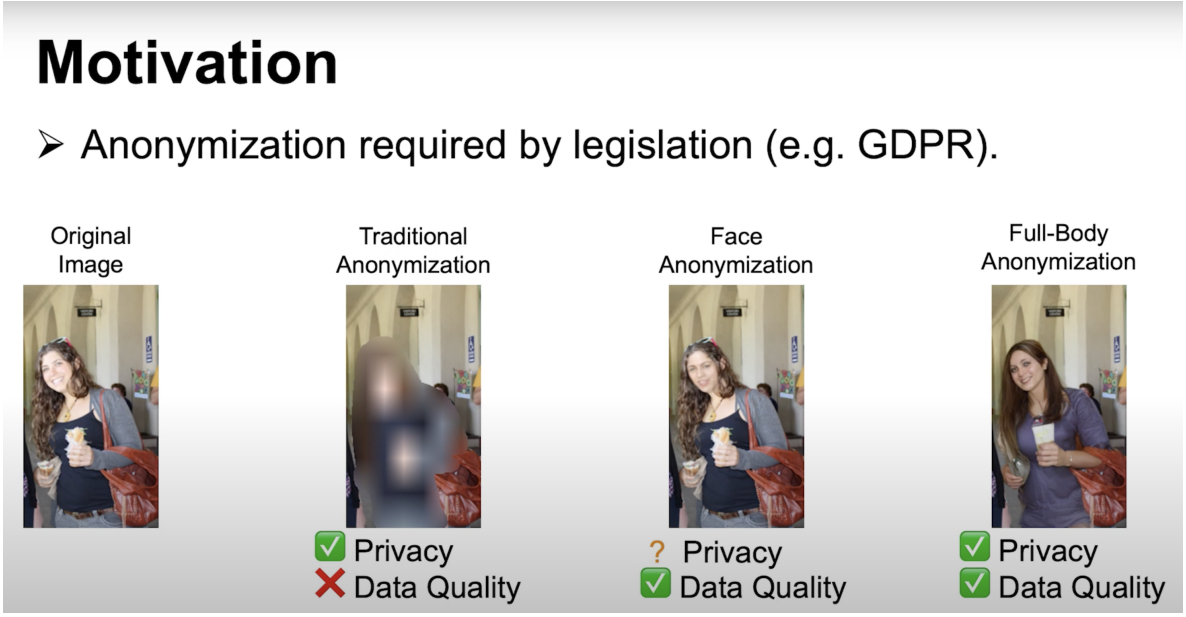


Figure 1.2: The motivation for anonymization as presented in DeepPrivacy2[3])

There are two main questions we aimed to address as part of this project:

- What methods can be used to anonymize datasets? Initially, the datasets had to be anonymized to ensure privacy while also being suitable for use in action recognition algorithms. While blurring is a common anonymization technique, it is not effective for action recognition as it results in insufficient visual detail and clarity for identifying and classifying human movements and activities. Blurring can lead to the loss of significant features and movement information, hindering computer vision algorithms' ability to accurately interpret actions. Moreover, it can introduce noise and distortion to the image, making it challenging for algorithms to differentiate between various actions. Consequently, more precise and detailed images are necessary for accurate action recognition. As a result, we explored alternative techniques that could be employed to anonymize the datasets.
- Is it possible to use anonymized datasets? Once we obtained our anonymized datasets, we aimed to verify their compatibility with various machine learning algorithms. As a result, we sought to develop a means of evaluating whether the anonymized datasets performed as effectively as the original dataset.

1.3 Overview

In essence, the primary objective of this dissertation is to create anonymized datasets and evaluate their efficacy in action recognition.

To achieve this, we explored different anonymization and action recognition methods to determine the most effective approaches for our research. The relevant literature we consulted for this task is outlined in Chapter 2.

Through our review of relevant papers, we gained insights into the current algorithms used in the field. We provided a comprehensive account of how we obtained and labeled the video data. In terms of anonymization, we explored the effectiveness of Generative Adversarial Networks (GANs) and diffusion methods, and ultimately decided to implement the style-based GAN framework of DeepPrivacy2. Further details can be found in Chapter 3.

Once we collected the data and completed the anonymization process, we needed to ensure that the anonymized data were of sufficient quality. To achieve this, we employed 2D and 3D pose estimations to compare the original and anonymized data. Additionally, we used Temporal Adaptive Module (TAM) to evaluate the performance of the anonymized data compared to the original data for action recognition. To perform these evaluations, we utilized an open-source video understanding toolbox based on PyTorch called MMAction2. For a more comprehensive description, please see Chapter 4.

Based on the statistical data obtained, we conducted an evaluation by measuring the Euclidean distance between the original and anonymized datasets for 2D and 3D pose estimations. In addition, we used top 1 and top 2 accuracy as well as macro accuracy for evaluating the performance of TAM. To gain a better understanding of the model's performance, we also created confusion matrices. The results of these evaluations can be found in Chapter 5.

Using all of the information above, we discuss our conclusion in Chapter 6. This chapter contains the scope of our work, what was and was not effective, and opportunities for future research.

Chapter 2

Literature Review

Due to privacy concerns, we had to anonymize the participants in our data before sharing it with others. While simply covering faces through edge detection and Gaussian blur may seem like a straightforward solution, it can actually cause distortion of the data and create issues for algorithms trying to detect poses or produce accurate results. As a result, we needed to find an alternative method to anonymize the data while preserving facial features such as GAN-based methods and diffusion methods. Because of constraints such as computational complexity and a lack of known methods, we chose the GAN method DeepPrivacy2. Further details at 2.1.

In their daily lives, elderly people face numerous challenges, such as neurological weakness, motor movement deterioration, and restricted movement, all of which can impair their ability to live independently. Several techniques for diagnosis and prognosis have been developed, including wearable sensors and vision techniques that detect falls. Wearable sensors, however, may not be appropriate for the elderly due to forgetfulness or resentment of intrusion.

Consequently, we concentrated on algorithms that rely solely on vision techniques. Taking into account various factors, including accuracy and training time, we selected the Temporal Adaptive Module (TAM) as our action recognition algorithm, which utilizes RGB data from videos or images. Further details can be found in 2.2.

2.1 Face Annonymization

Data processing that protects privacy is becoming increasingly critical, especially in light of rules such as the General Data Protection Regulation (GDPR). The removal of privacy-sensitive information while generating realistic faces with seamless transitions remains a difficulty when anonymizing images without sacrificing quality.

Masking, blurring, and pixelation are common image anonymization methods, however they impair image quality, rendering the data unsuitable for many purposes. Although early K-same family algorithms give greater privacy assurances and data usability, they produce extremely damaged images. Current research on deep generative

models such as GAN and diffusion models shows that learning-based anonymization may realistically anonymize data while keeping it usable for downstream applications.

Because of its capacity to generate high-quality images while retaining data distribution and privacy, GANs and diffusion models have grown in popularity for image anonymization. Its adaptability, recent research advances, and strong privacy assurances make them suited for a wide range of anonymization jobs, ensuring the data's usability for downstream applications.

2.1.1 Generative Adversarial Networks (GANs)

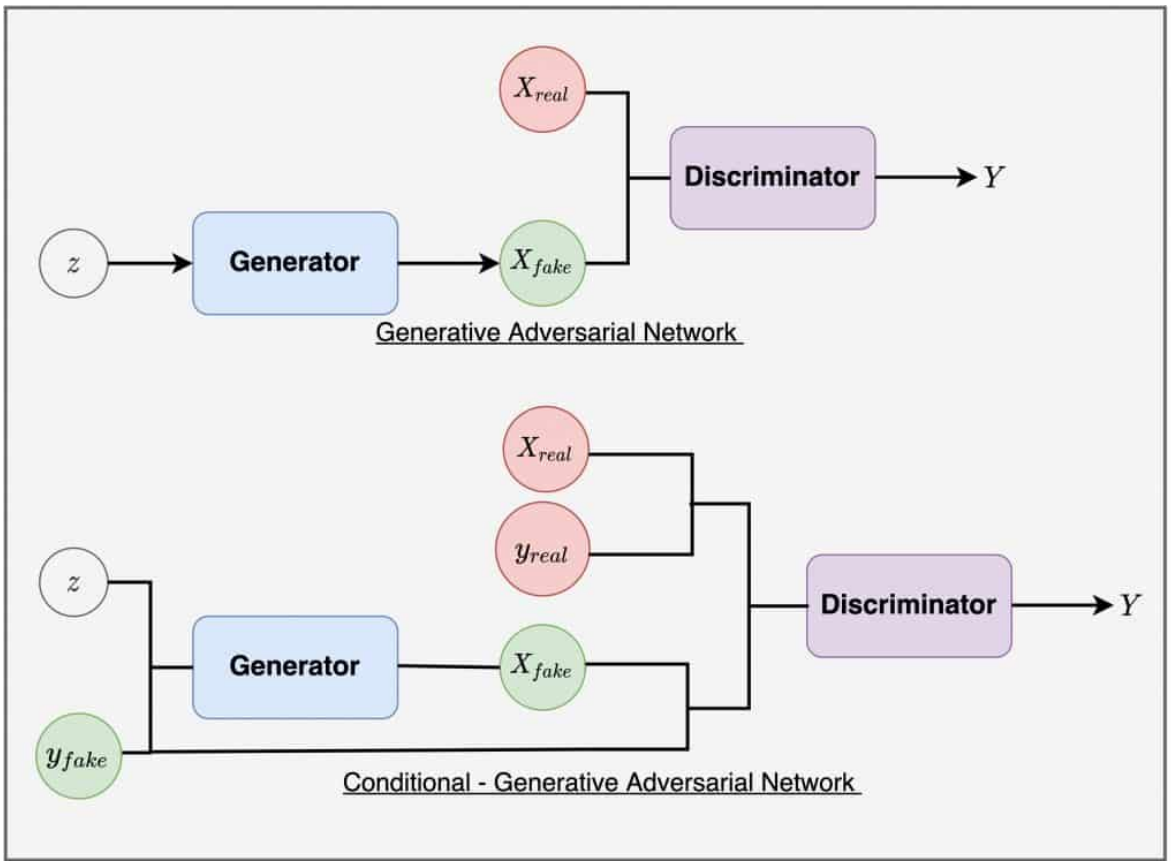


Figure 2.1: GAN and conditional GAN Structure. GANs combine two neural networks: The purpose of the generator is to create synthetic data which is then provided to the discriminator, which should decide if the data is genuine or not. In contrast to regular GANs, cGAN also provides additional information in the form of labels 'y_fake' and 'y_real'[4]

Generative Adversarial Networks (GANs) are a form of machine learning technique that generates new data by combining two neural networks. The generator network generates synthetic data, while the discriminator network checks the created data's validity by comparing it to genuine data. The two networks collaborate, with the generator striving to produce data that is indistinguishable from actual data and the

discriminator increasing its capacity to detect faked data. GANs have been used to create incredibly realistic pictures, films, and sounds that are difficult to differentiate from the genuine item. They are currently used in a variety of industries, including art, entertainment, and medicine[5].

Conditional GAN

A Conditional Generative Adversarial Network (cGAN) is a kind of GAN that learns to create data samples based on external input such as labels or attributes. In contrast, a normal GAN learns to generate realistic data samples from a random noise input, while the discriminator learns to distinguish between genuine and generated samples. A cGAN provides additional information to both the generator and the discriminator, which acts as a condition for the data generation process.

The primary goal of cGAN is to make generated data samples more controlled and focused on specific features or classes. A cGAN, for example, can be trained to generate images of specific things, such as cats or dogs, when given their labels. The generator and discriminator are conditioned on the information provided, allowing the network to provide more accurate and tailored outputs.

Conditional GANs have found widespread use in image-to-image translation, text-to-image synthesis, data augmentation, and style transfer.

U-net

U-Net is a convolutional neural network (CNN) architecture specifically built for image segmentation tasks. The architecture is known as "U-Net" because of its U-shaped structure when visualised.

U-Net is divided into two sections: The contracting (encoder) path and the expansive (decoder) path. The contracting route consists of a series of convolutional and pooling layers that capture context while reducing the spatial dimensions of the input image. The expansive path is made up of a sequence of up-convolutional and concatenation layers that recover the spatial dimensions and merge the contracting path's high-level contextual information with the local spatial information. This combination enables U-Net to do exact image segmentation.

Because of its capacity to deliver accurate and precise segmentations with a relatively small number of training photos, U-Net has been widely used in many image segmentation tasks, particularly in biomedical applications[6].

DeepPrivacy

One example of a popular method that employs both a cGAN and U-net architecture is DeepPrivacy. It uses a cGAN that creates anonymized faces from existing backdrops and sparse posture annotations. For increased picture quality and training time, it employs a U-net architecture and progressive increasing training technique.

Two basic annotations are required for the model: a bounding box to designate the privacy-sensitive area and sparse posture estimation with keypoints for the ears, eyes, nose, and shoulders. The authors provide the Flickr Diverse Faces (FDF) dataset, which contains 1.47 million faces annotated with bounding boxes and keypoints. The performance of the model is assessed by anonymizing the WIDER-Face dataset and analyzing the impact on Average Precision (AP). DeepPrivacy outperforms typical anonymization techniques such as pixelation, severe blur, and black-out, achieving 99.3% of the original AP.

When tested with a cutting-edge face identification method, the model produced high-quality images on the heterogeneous WIDER-Face dataset, attaining 99.3% of the original average precision. In terms of image quality and anonymization certainty, this is a huge improvement over earlier systems. Ablation tests on the FDF dataset show that a higher model size and the addition of sparse posture information are important for generating high-quality images.

DeepPrivacy is a conceptually simple GAN that may be easily adapted to improve further. Handling irregular poses, tough occlusions, complicated backdrops, and maintaining temporal consistency in movies, on the other hand, are topics for future development.[7].

On top of DeepPrivacy, DeepPrivacy2 uses additional methods such as a Context Surface Encoder (CSE) and StyleGAN2.

CSE

DeepPrivacy2 employs the Context Surface Encoder (CSE) to better preserve picture structural information during anonymization. The CSE is built into the full-body synthesis generator, one of three independently trained generators used by DeepPrivacy2 for anonymization tasks.

The CSE embedding is supplied into the generator along with the masked input picture. This enables the generator to use information about the scene's underlying 3D geometry, which aids in the realistic appearance of anonymised photographs, especially in the presence of occlusions and difficult poses.

To summarise, the Context Surface Encoder is a key component of DeepPrivacy2 that improves image quality by taking into account the 3D structure of the scene. [8].

StyleGAN

StyleGAN, short for Style Generative Adversarial Network, is a deep learning architecture that uses a unique mapping network and synthesis network to generate high-quality, high-resolution images. It introduces the concept of style transfer, which enables the generator to generate images with a wide range of configurable attributes. The approach has produced astonishing results in the creation of lifelike images such as human faces, animals, and objects[9].

StyleGAN2

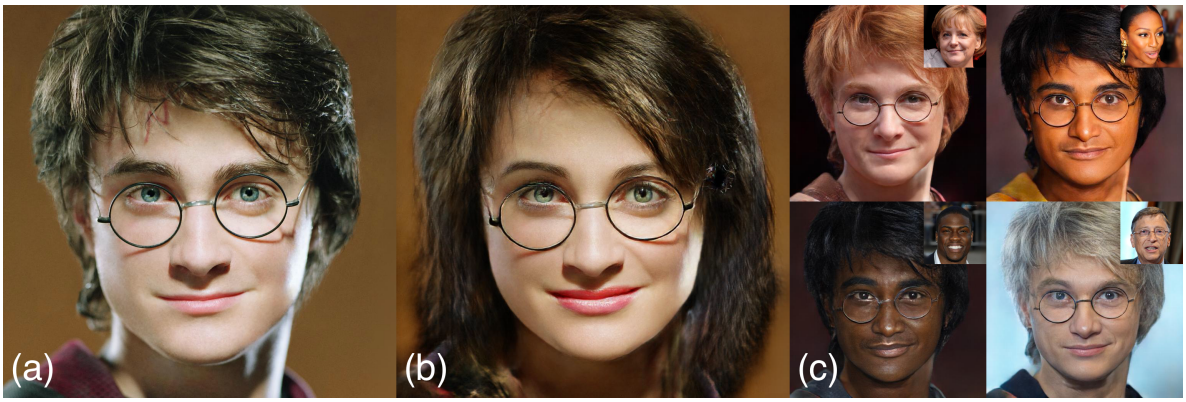


Figure 2.2: Generated image examples from (a) a Celeba-HQ source image: (b) gender switch at 1024x1024 and (c) style mixing at 512x512. Samples are generated feed-forward, and StyleGAN2 was trained on FFHQ [10]

StyleGAN2 builds on the original StyleGAN architecture. StyleGAN2's authors solved some of the flaws and shortcomings that existed in the initial edition. Some of the major issues of StyleGAN that were addressed in StyleGAN2 are as follows:

- StyleGAN had a habit of producing images with artefacts, such as repetitive patterns, blob-like formations, or other irregularities that made the created images appear less realistic. StyleGAN2 eliminates these artefacts by introducing a new normalisation approach known as Weight Demodulation, which aids in the creation of more visually attractive and realistic images.

- StyleGAN2 included a new generator architecture with a modified normalization mechanism, which considerably improved training stability. StyleGAN2's training procedure is more robust and converges faster, resulting in superior picture production.

- StyleGAN2 contains a new regularisation technique known as Mixing Regularization. The generator is encouraged to employ different regions of the latent space for different characteristics of the generated image using this strategy. This creates a more diversified group of generated images and aids in the prevention of mode collapse, a major problem in GANs when the generator produces only a restricted variety of images.

- Style inputs may now be defined more simply in StyleGAN2, providing for greater control over the appearance of the output images. Its enhanced configurability makes it easy to modify the look of generated graphics and fine-tune the model for specific applications.

In summary, StyleGAN2 solved various flaws found in the original StyleGAN, resulting in higher image quality, increased training stability, and more adjustable styles.[11].

DeepPrivacy2

DeepPrivacy2 is an image anonymization system that employs three independently trained generators for different detection categories: detection with Context Surface Encoder (CSE), detection without CSE, and detection of faces. The anonymization task is framed as an image inpainting task, in which the portions to be anonymized are deleted and the missing region is filled in by a generator. The generators are built on a style-based U-Net architecture that includes a context encoder and a style-based decoder.

At each feature map resolution, the context encoder employs a series of convolutions and downsampling layers with residual connections. The encoder does not employ any normalising layers, however instance normalisation is done to the features in the U-net skip connections. The decoder is designed in the same way as StyleGAN2, with operation order instance normalisation, convolution, and style modulation. In StyleGAN2, instance normalisation replaces the previous weight demodulation. DeepPrivacy2 employs StyleGan2 to produce images that depict particular attributes, such as individuals wearing glasses or having a mustache.

DeepPrivacy and DeepPrivacy2 are both anonymization technologies for protecting people' privacy in photos or videos. This is accomplished by replacing the faces in the input photographs or videos with synthetic faces generated by deep learning techniques. Nonetheless, there are some distinctions between the two approaches:

- DeepPrivacy2 is an upgraded version of DeepPrivacy that provides higher performance in terms of accuracy and speed. DeepPrivacy2 is faster than its predecessor due to its more efficient architecture.
- DeepPrivacy2 produces synthetic faces with higher visual quality than DeepPrivacy. This means that DeepPrivacy2's output photos and videos will generally have more realistic and higher-quality faces.
- DeepPrivacy2 is more resilient in dealing with a broader range of face orientations, occlusions, and other tough situations in input photos or videos. As a result, it is more appropriate for real-world applications where the quality of the input data may fluctuate.
- DeepPrivacy2 is trained on a larger and more diversified dataset, which aids in its generalisation capabilities. DeepPrivacy2 is hence more likely to give better outcomes on a larger range of input photos or videos.

In short, DeepPrivacy2 is an enhanced version of DeepPrivacy that improves efficiency, quality, robustness, and generalisation capabilities. It is also capable of anonymizing a whole body unlike DeepPrivacy, but we did not use the feature since we are only interested in face anonymization and it takes much more time compared to anonymize faces only[12].

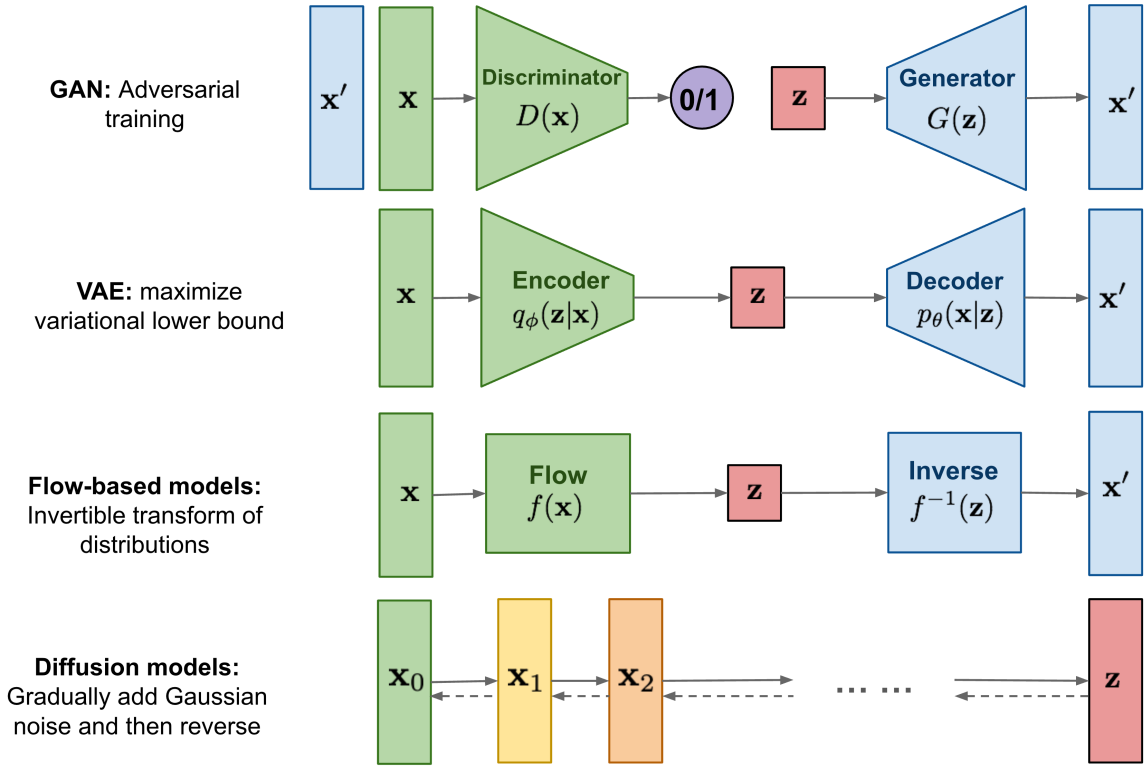


Figure 2.3: Comparison between GAN and diffusion[13]

2.1.2 Diffusion

Diffusion, on the other hand, uses the principles of the diffusion process to generate images or videos. It involves adding noise to the image or video, and then using a series of diffusion steps to spread the noise throughout the image or video. This process results in a new image or video that is similar to the original but with some key differences that can be controlled by the user.

Although diffusion techniques represent state-of-the-art algorithms and generally provide superior output quality compared to GANs, we opted for the GAN-based algorithm DeepPrivacy2 for the following reasons:

- Slower inference: Diffusion models necessitate running a denoising process, which typically consists of a series of noise-reduction steps. This is typically much slower than GAN-based generation, which typically involves a single forward pass through the generator network.
- Reliability: Since diffusion methods are cutting-edge algorithms, there are fewer APIs available and their effectiveness is not as well-established compared to GAN methods. As our goal is to create and share a dataset, ensuring its feasibility and practicality is of utmost importance, leading us to opt for GAN methods instead.

2.2 Action Recognition

Action recognition algorithms can work with a wide range of data formats, including:

- RGB data: Photos or movies captured by a camera that contain colour information for each pixel.
- Depth data: Depth data tells you how far away items are from the camera. This can be achieved using depth cameras such as the Microsoft Kinect or stereo vision techniques. Depth data can help increase the robustness of action detection systems, particularly when the backdrop and subject colours are comparable.
- Skeleton data: It is made up of 2D or 3D joint locations of a human in a video. Human posture estimating methods can be used to retrieve these positions. The spatial and temporal patterns of joint movements are analysed in skeleton-based action recognition.
- Optical Flow data: Optical flow data represents the movement of objects in a video clip between consecutive frames. This information can be utilised to record the temporal dynamics of actions, which is useful for action recognition.

In this project, our primary focus was on RGB and Skeleton data, which were acquired through 2D pose estimation using RGB data combined with depth information.

2.2.1 Pose estimation

There are two basic approaches to human pose estimation: top-down and bottom-up. The bottom-up approach detects particular keypoints (such as joints or body components) in an image without taking into account their relationships with unique people. The identified keypoints are then classified as individual human instances based on geographical linkages and other heuristic principles.

Bottom-up approaches are often more computationally efficient than top-down approaches since they do not require running the posture estimate model for each person detected in the image. However, in busy settings or when there are occlusions, the grouping step can be difficult.

MMPose

MMPose is an open-source human pose estimation toolbox based on the PyTorch deep learning framework. It offers valuable APIs, including models for pose estimation.

2.2.2 TAM

TAM is a video recogntion algorithm provided by MMAction2. It is aimed to more efficiently capture long-term temporal interdependence in videos. Conventional video recognition systems sometimes fail to capture the temporal links between frames, particularly when the relationships are complicated and non-linear. TAM overcomes this

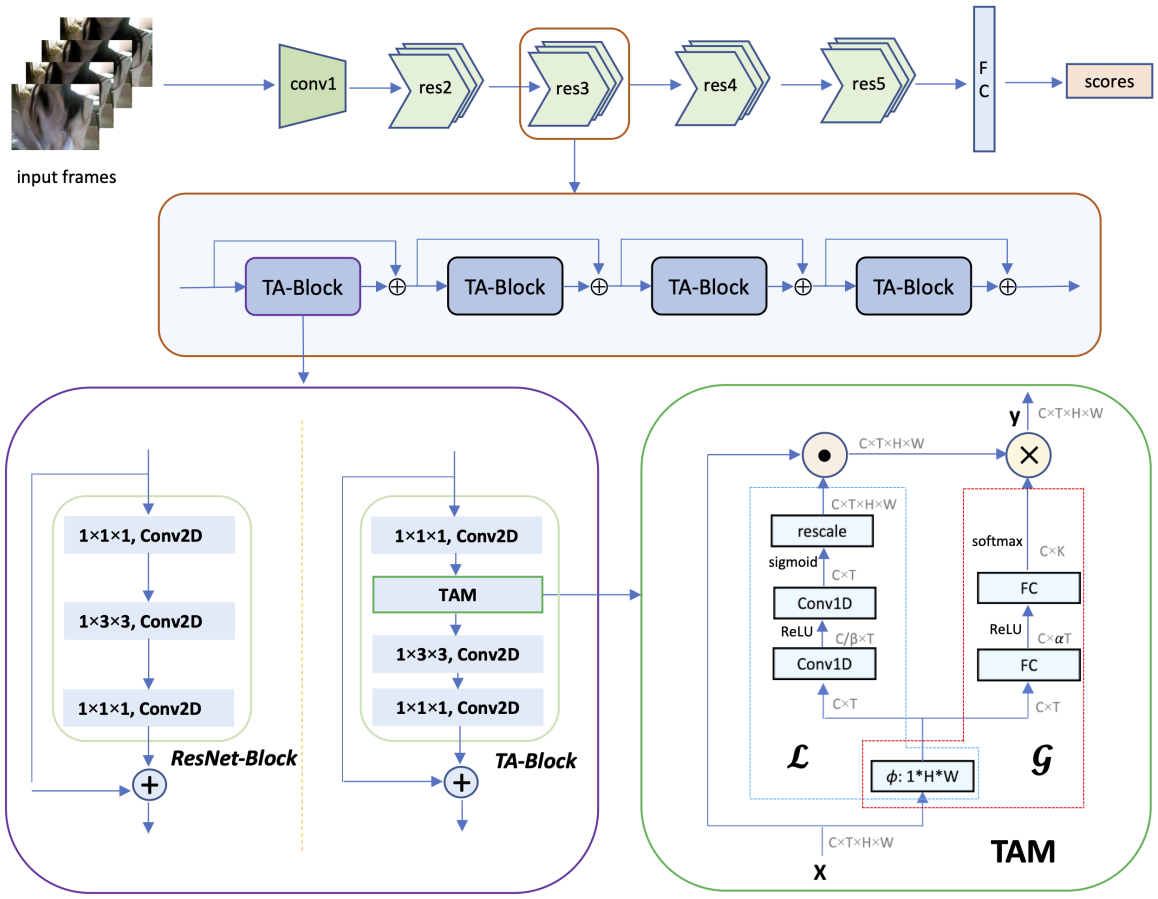


Figure 2.4: Architecture comparison between ResNet-block and TA-block[14]

issue by learning to choose relevant temporal information from many scales and resolutions adaptively.

TAM is inserted between convolutional layers in existing video recognition algorithms, such as 3D CNNs. This module is lightweight and can be readily added into existing models without raising computational complexity significantly. TAM's main advantage is that it can learn adaptively and focus on the most significant temporal information, resulting in increased video identification ability. [14]

Chapter 3

Data Acquisition

3.1 Creating and Labelling Eating Action Dataset

3.1.1 Recording

Since there is no video-based dataset dedicated to eating actions, we decided to create our own dataset. We shot at least two eating videos for each participant, one with the person wearing weight wristbands and the other without, in order to mimic the increased physical strain experienced by elders. The dataset includes 64 sequences of Professor Robert Fisher and 9 of his student. The dataset was collected by Ahmed Raza, a current Ph.D student who initially proposed this project[15]. The sequences were also recorded at various frame per second (fps) rates. The majority of the sequences were recorded at 30 frames per second, although some sequences were recorded at 15 frames per second. We recorded with an RGB-Depth camera, which can capture RGB rawframes and depth data while filming videos.

We anticipated that because most individuals eat while sitting, elderly would do the same. We mainly employed upper body poses since major lower body movements occur infrequently, if at all, while individuals are eating - and a camera cannot usually view the lower half of the body since the lower body is often obstructed by the table. The head, shoulders, elbows, and wrists can all be included in the upper body position. We need to collect continual data from elders in order to detect indicators of behavioural changes, thus we assumed in this scenario that elders would eat in the same area - at home, with a camera present to capture their eating. We shared a mostly common environment setup to simulate this environment; a person sitting at a table indoors while eating and drinking.

A senior surveillance system should be designed to accommodate a diverse range of individuals, as people have varied eating habits depending on their cultural background. For instance, East Asians are more likely to use chopsticks, while Europeans typically prefer knives and forks. We recorded our videos using various eating utensils to reflect the diverse preferences of seniors. For example, one student used chopsticks, while others used a spoon or their hands. As a result, our dataset includes videos of students eating with chopsticks, a spoon, fork and knife or their hands.

3.1.2 Labelling

We needed to describe actions as a combination of sub-actions, which make up composite actions, in order to quantify these varied scenarios. Drinking, for example, is a composite action consisting of the following sub-actions: picking up a glass, drinking, and putting the glass back. We went through various procedures to create a sub-action list in order to avoid a lack of data. If we try to create too many sub-actions, we may not have enough data to train the model. This is due to the fact that with highly detailed labels, some classes might have very few examples, leading to class imbalance. For example, separating actions based on the utensils individuals use, such as chopsticks, knives, and forks, increases the likelihood of the model misclassifying actions if the data was obtained from a population sample of a Western country.

Our provisional classification was composed of 33 sub-actions, but due to a scarcity of samples, we had to combine several sub-actions. Additionally, certain sub-actions were difficult to detect visually, and others were irrelevant to eating, so we merged or deleted them. Therefore, the number of actions was eventually reduced to 16, such as 'Pick food from utensil with both hands', 'Chewing', etc.. After sub-actions were decided, we utilised an open-source tool called VGG Image Annotator (VIA) to label our videos[16].

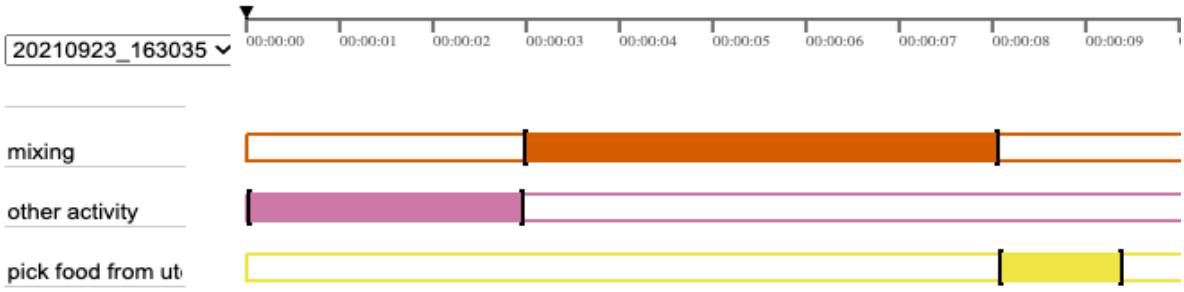


Figure 3.1: VIA timestamps

As shown in Figure 3.1, we started by entering the activities in the bottom left corner. Following that, we played the video and set timestamps to indicate that an action occurred during that time frame. For example, 'other activity' occurred between 0 and 3 seconds while 'mixing' occurred between 3 and 8 seconds.

One major limitation in this stage is misclassification, since we asked the participants to label their own videos. This might cause several problems such as inaccurate ground truth, inconsistent labeling, and incomplete labeling. People not only have varied eating habits, but also have various perspectives on classification, which means that individuals may label activities differently even when faced with the same actions. In South Korea, for example, soup is commonly served with rice. Some individuals combine rice with soup and sip it from a bowl while holding the bowl, although this is considered eating. Those who consume soup with a spoon from a soup dish, on the other hand, would consider this drinking. Additionally, in some cases, participants accidentally skip some frames due to the repetitive nature of the task, which can cause them to lose focus easily. As a result, a certain amount of misclassification of labelled

data is possible. However, because machine learning models rely on precisely labelled data to learn the correct patterns and generate accurate predictions, such misclassification can have a negative impact on their training and performance. To address this issue, techniques such as quality checks or automated technologies to identify potential missed frames could have been implemented. Nonetheless, even though some quality checks were carried out, due to a lack of resources, these methods were not applied to the entire dataset, which may have resulted in some inaccuracies in the labeled data.

Action ID	**Action Name**	**Count**
0	Other	763
1	Pick food from utensil with both hands	39
2	Pick food from utensil with one hand	269
3	Move hand towards mouth	1205
4	Eat it	1106
5	Move hand away from mouth	1201
6	Chewing	114
7	Pick up a cup/glass	689
8	Drink	70
9	Put the cup/glass back	225
10	No action	311
11	Pick food from utensil with tool in one hand	104
12	Put one tool back	134
13	Food in hand at table	104
14	Pick up tools with both hands	396
15	Pick food from utensil with tools in both hands	60
	Total	6790

Table 3.1: Action ID, Action Name, and Count of the respective action in the dataset.

The tables 3.1 and 3.1.2 show the summary statistics on the different actions as well as participants in the videos and Figure 3.2 shows the distribution of actions. The actions are tied to IDs, so we can use the IDs for training and testing models.

3.2 Anonymizing the videos

After recording and labelling the videos, we proceeded to anonymize them using DeepPrivacy2. Anonymizing the videos was relatively straightforward, as we only needed to use the Application Programming Interface (API) provided by DeepPrivacy2,which we needed to set the appropriate directories and execute the command in the terminal. Once this was done, the anonymization process was initiated. However, obtaining anonymized videos took a considerable amount of time due to the following factors:

- **Environment setting:** It was challenging to run the API since it is quite new, with only a few users. Consequently, it doesn't provide adequate support for certain environments, such as Windows or MacOS.

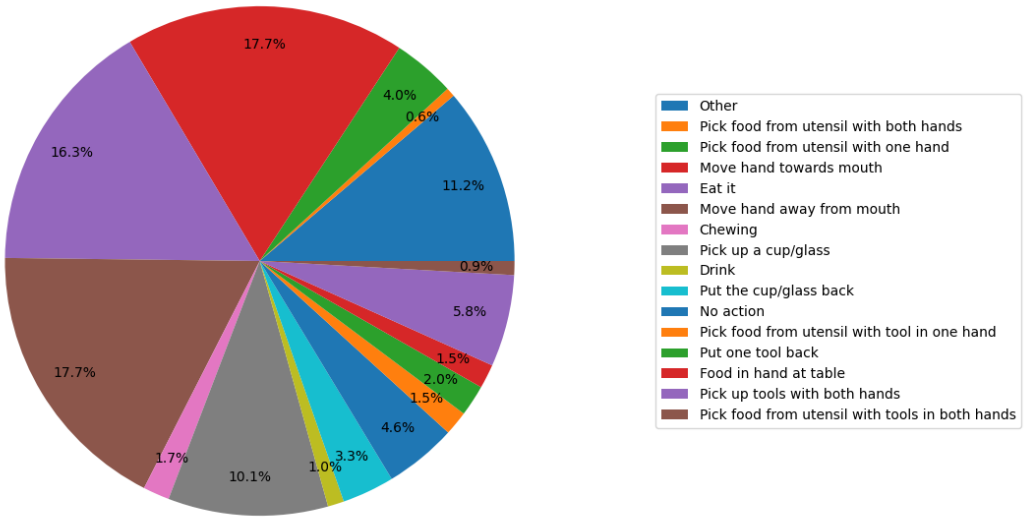


Figure 3.2: Distribution of actions

- **Lack of GPUs:** The process heavily requires GPUs, but we were not able to access high-quality GPUs. For example, researchers often use several GPUs together to make and benchmark models while we had only 1 outdated GPU, so this process proved to be time consuming.
- **Lack of Disk space:** Videos and depth data occupy a significant amount of disk space, with zip files taking up around 500GB and over 1TB when uncompressed. We needed to purchase an external drive to store the data, and accessing the data on the external drive slowed down the process further.

It took a week to anonymize 64 videos. We initially planned to use diffusion methods to convert videos, but we changed our approach after encountering these challenges, as diffusion methods require more GPUs and were not feasible in our circumstances.

3.3 Summary

In conclusion, in order to create a dataset for eating actions, we recorded 64 sequences of individuals eating, using a variety of eating utensils to accommodate different cultural backgrounds. They focused on upper body poses and recorded videos with and without weight wristbands to mimic the physical strength of elders. The dataset was collected and labelled using VGG Image Annotator (VIA), resulting in 16 different actions.

However, there were potential issues with misclassification as students labeled their own videos, and some inaccuracies in the labeled data may have resulted from a lack of quality checks. After recording and labeling the videos, they were anonymized using DeepPrivacy2, but the process was slow due to challenges related to environment setting, lack of GPUs, and disk space.

Videos	Age	Gender	Tools	Ethnicity
8	age > 35	M	fork, knife, spoon, no tool	Caucasian
32	$26 <$ age < 35	M	no tools, fork, knife, spoon	South Asian
4	$26 <$ age < 35	M	chopsticks, no tool, fork	East Asian
4	$26 <$ age < 35	M	no tool, fork	East Asian
4	age ≤ 25	M	chopsticks, fork	East Asian
4	$26 <$ age < 35	M	no tools, spoon	Caucasian
2	$26 <$ age < 35	F	Spoon	South Asian
2	age ≤ 25	F	Spoon	South Asian
2	age ≤ 25	M	chopsticks	East Asian
2	age ≤ 25	M	no tool	South Asian

Despite utilizing weighted wristbands to simulate the movements of elderly individuals, our participant pool still lacked a significant number of older individuals. Additionally, the sample was heavily skewed towards individuals of East and South Asian backgrounds. Female participants were also underrepresented here.

Despite utilizing weighted wristbands to simulate the movements of elderly individuals, our participant pool still lacked a significant number of older individuals. Additionally, the sample was heavily skewed towards individuals of East and South Asian backgrounds. Female participants were also underrepresented here.

Table 3.2: Summary table of participant Information[17].

Despite utilizing weighted wristbands to simulate the movements of elderly individuals, our participant pool still lacked a significant number of older individuals. Additionally, the sample was heavily skewed towards individuals of East and South Asian backgrounds. Female participants were also underrepresented here.

Chapter 4

Methodology

After collecting, labeling, and anonymizing the data, we had to ensure that the anonymized data met certain quality standards, so that they could be shared and utilized by other researchers. Since we already had 2D and 3D pose estimations for the original dataset, we gathered the 2D and 3D pose estimations for the anonymized data and compared them with the original using Euclidean Distance between the original and the anonymized pose coordinates.

The subsequent step involved preparing the dataset for model training using TAM. This procedure depends on the specific framework employed, as different frameworks use unique approaches for reading labels and files. In this instance, we used an open-source toolbox based on PyTorch called MMAction2. For training, TAM is already integrated into MMAction2, eliminating the need for separate algorithm implementation. To ensure the reliability of the results, we carried out 5-fold cross-validation for training and testing the model. Training was conducted five times for each split. For testing, we utilized the saved weights from the training to predict classes, and we gathered various statistics, such as top 1, top 2, and Macro accuracy, for assessment.

4.1 Pose Comparison

We employed an API from MMPose that utilizes RGB data and a bottom-up model pretrained on the COCO dataset to obtain 2D pose estimation. Following the acquisition of 2D pose estimations, we used depth data to achieve 3D pose estimation.

Before proceeding to the next stage, we had to preprocess the acquired data to address the presence of outlier frames in the statistics. For instance, some projects featured participants who were absent from the video at the beginning and the end because they had to turn the camera on and off themselves. We labeled these frames as 'others', and it was necessary to remove these actions when gathering statistics.

We then compared each joint illustrated in Figure 4.1 between the original and anonymized 2D and 3D poses. These poses were visualized using dots and lines with varying colors, as demonstrated in Figure 4.2, when the distance between the original and anonymized poses was at its maximum, to assess the acceptability of the

anonymized data. Additionally, we collected statistics from these poses, as displayed in Table 4.1 and Table 4.2. We plotted 3D poses without images because it was hard to show 3D poses in 2D images. Figure 4.3 shows how we visualized 3D poses. We also gathered the frequency of 2D and 3D distances for each joint. Figure 4.4 shows the case of joint head for 2D and 3D distances.

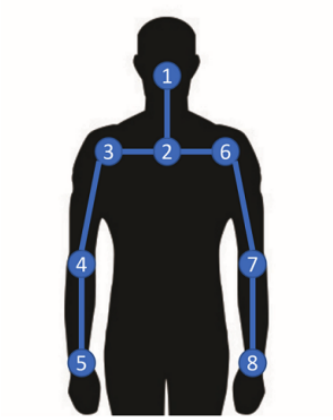
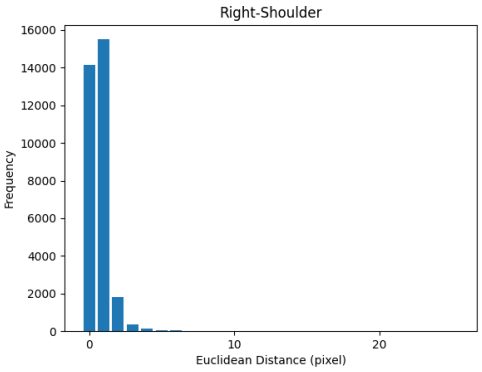
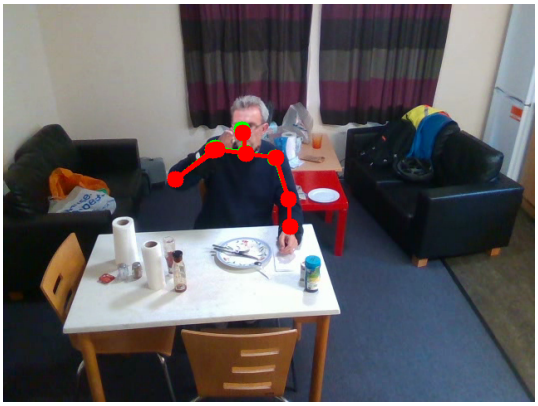


Figure 4.1: The 8 upper body-joints. 1) head, 2) Mid-Shoulder, 3) Right-Shoulder, 4) Right-Elbow, 5) Right-Wrist, 6) Left-shoulder, 7) Left-Elbow, 8) Left-Wrist[17]



(a) Project 20210923_155848



(b) Project 20210811_132135

Figure 4.2: Two distinct projects display images with 2D poses, where the most significant difference in the Right-Shoulder position occurred. Green lines represent the original 2D poses, while red lines depict the anonymized 2D poses.

4.2 Action Recognition

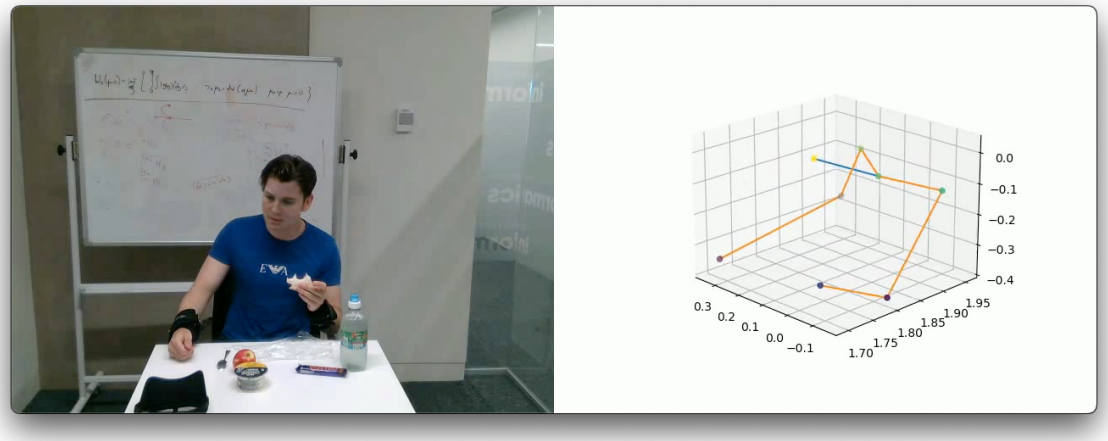
After we found out 2D and 3D poses of the anonymized are feasible, we wanted to investigate further using action recognition. To get fair results, we used cross validation shown in Table 4.3. It aids in reducing overfitting and obtaining a more accurate approximation of the model's performance on previously unknown data. Cross-validation divides the dataset into a number of subsets or "folds." Here we used 5 folds. After

Keypoint	**Median**	**Mean**	**Max**	**Min**	**STD**
head	2.001	2.337	21.466	0.075	1.643
Mid-Shoulder	0.318	0.407	10.81	0.008	0.532
Right-Shoulder	0.615	0.698	11.459	0.021	0.643
Right-Elbow	0.744	1.097	13.093	0.014	1.222
Right-Wrist	1.975	3.116	30.707	0.022	4.124
Left-shoulder	0.417	0.489	10.439	0.017	0.495
Left-Elbow	0.73	0.897	5.193	0.038	0.701
Left-Wrist	2.03	2.247	13.302	0.094	1.316

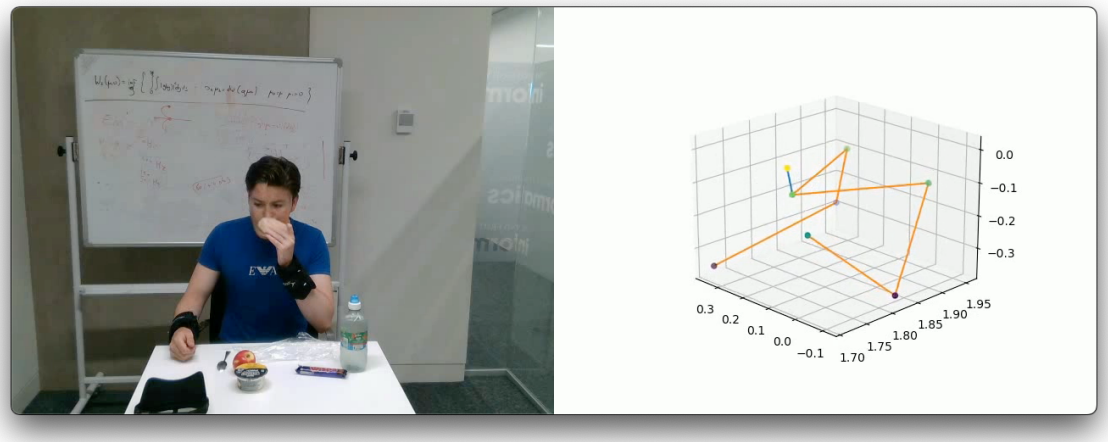
Table 4.1: 2D Euclidean distance statistics between the original and the anonymized dataset for the sameple project "20210529_153708". Unit is pixel. STD stands for Standard Deviation

Keypoint	**Median**	**Mean**	**Max**	**Min**	**STD**
head	0.01	0.014	0.236	0	0.021
Mid-Shoulder	0.003	0.004	0.076	0	0.006
Right-Shoulder	0.004	0.004	0.087	0	0.006
Right-Elbow	0.006	0.008	0.351	0	0.018
Right-Wrist	0.011	0.028	0.928	0	0.054
Left-shoulder	0.004	0.004	0.067	0	0.005
Left-Elbow	0.005	0.006	0.216	0	0.01
Left-Wrist	0.011	0.015	0.63	0	0.022

Table 4.2: 3D Euclidean distance statistics between the original and the anonymized dataset for the sameple project "20210529 153708". Unit is meter. STD stands for Standard Deviation



(a) Project 20220812_124345 3D pose



(b) Project 20220812_124345 3D pose with the left wrist being in front of the mid shoulder

Figure 4.3: Anonymized video for project 20220812_1243453D and its 3D pose. The mid-shoulder is obscured due to the left wrist obstructing the infrared light required to capture depth information in the mid-shoulder region. The video can be found here

training on all but one of the folds, the model is tested on the remaining fold. This method is performed several times, with each fold serving as the test set only once. After this, the results are averaged to offer an overall performance metric. This method ensures that the model's performance is not unduly dependent on the precise training and test data used, resulting in more robust and dependable outcomes.

Rawframe data A, B, C, D, E		
Split	Train	Test
1	A, B, C, D	E
2	B, C, D, E	A
3	C, D, E, A	B
4	D, E, A, B	C
5	E, A, B, C	D

Table 4.3: Cross-Validation sets example

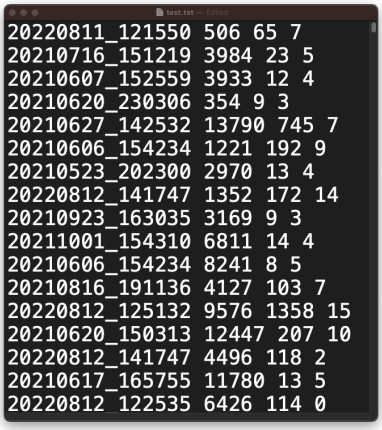


Figure 4.5: Label example. Each column signifies the data path, initial frame, frame duration, and action ID, respectively, from left to right.

Subsequently, we partitioned the projects into five segments, ensuring that the actions were evenly distributed across each segment. We then created five sets of labels for each split, containing both training and testing labels. Figure 4.5 illustrates the process of generating these labels.

For hyperparameter tunings, we used step learning rate which is a deep learning model training technique in which the learning rate is modified at specified intervals (steps) during the training process. This strategy enables the model to learn quickly at a higher learning rate at first, then gradually slow down as it approaches convergence. This assists the model in determining a more precise answer while avoiding overshooting the optimal position. Reduced learning rate at key steps focuses the model on refining the weights and biases, resulting in better generalisation and performance. Here, we decreased learning late by log scale at epoch 50, 75, and 90. For example, if the starting learning rate is 1e-2, then learning rate is 1e-3 at epoch 50, 1e-4 at epoch 75, and 1e-5 at epoch 90. We also looked at hyperparameter optimisation methods like hyperparameter sweeping and random search. Hyperparameter sweeping evaluates all potential hyperparameter value combinations within a given range, ensuring extensive exploration at the expense of higher computation. Random search, on the other hand, selects and tests hyperparameter values at random within a specified range, providing a more computationally efficient approach that can nevertheless be effective in uncovering optimal configurations. However, due to limited resources, we were unable to implement

these two methods in our project.

After this stage, the process of training and testing our models became straightforward. By using MMAction2 APIs, all we needed to do was create configuration files that specified the model structures and hyperparameters, and then add the dataset to the correct path.

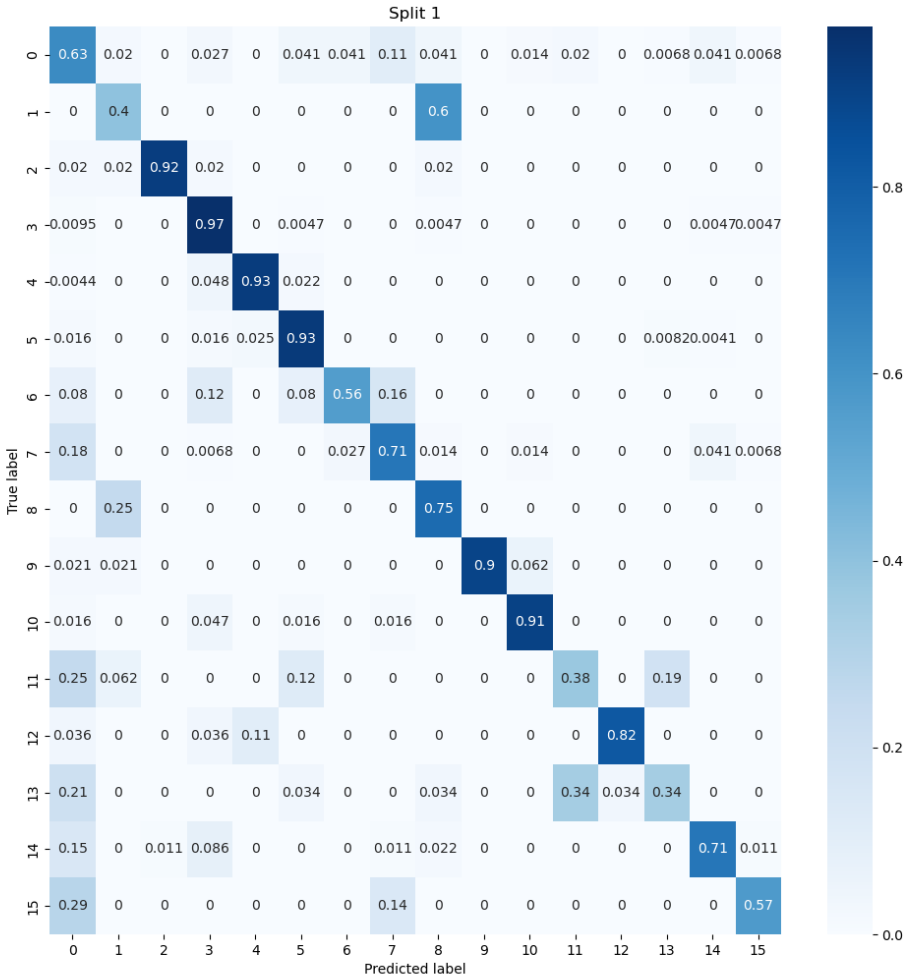
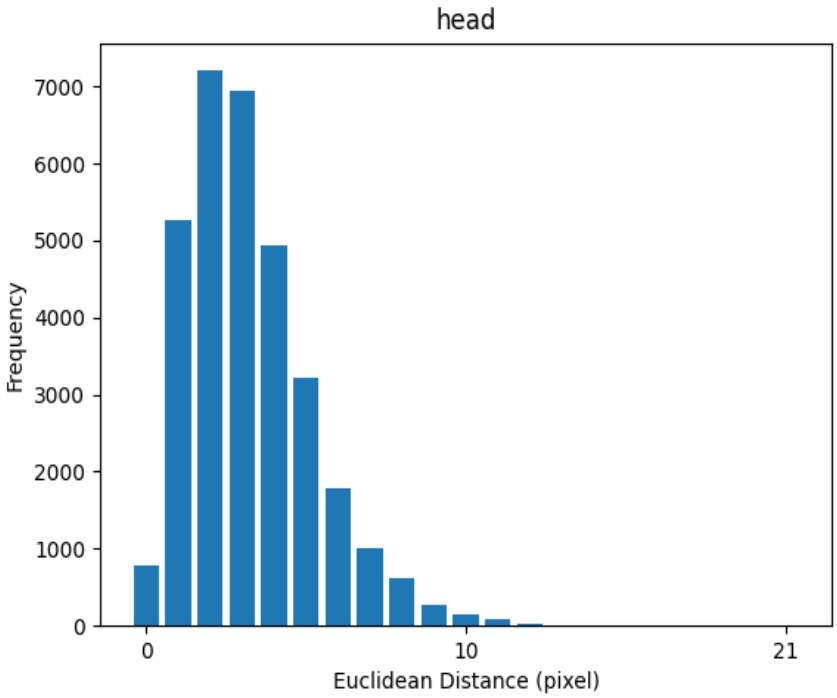


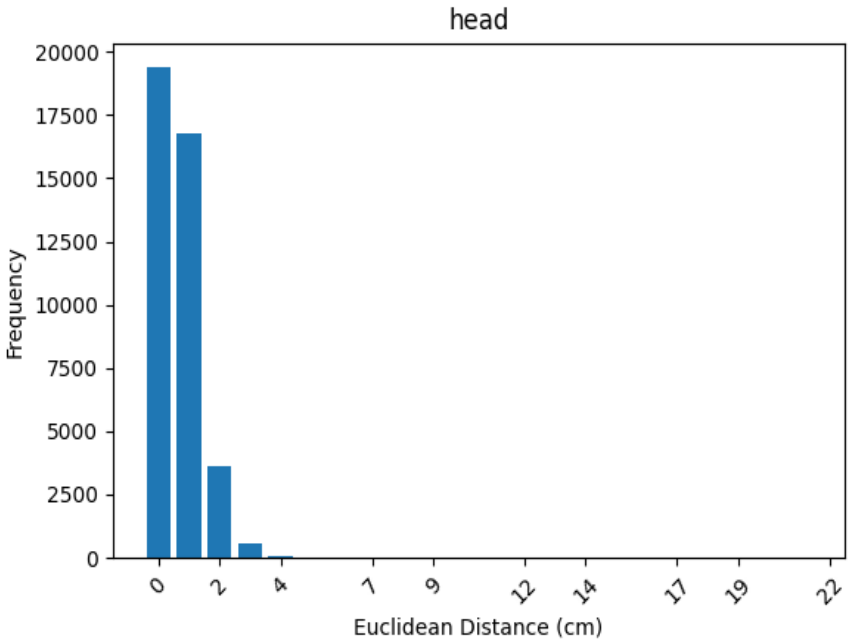
Figure 4.6: Confusion matrix for split 1. Numbers in x and y axis are action IDs which can be found in Table 3.1

During each training and testing phase, log files were generated, which included probability scores for various action classes. We utilized these logs to determine top 1, top 2, and Macro accuracy. We chose these accuracy because certain actions are dominant, so top 5 accuracy is easily over 95 percent, which is not useful to measure the performance. For macro accuracy, it is derived by averaging the accuracy of each

class. It comes in handy when the dataset is skewed and some classes have extremely few examples. In this situation, regardless of the number of cases, macro accuracy assigns equal weight to each class, so we chose macro accuracy. Additionally, we constructed confusion matrices based on these scores, as depicted in Figure 4.6. Note that we used normalized confusion matrices.



(a) Frequency of 2D distances for the head joint across 10 sample projects. Unit is pixel



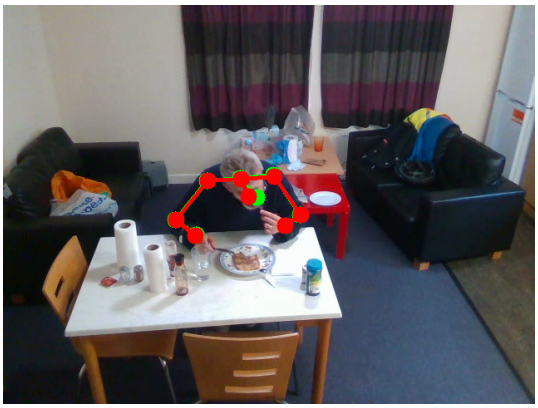
(b) Frequency of 3D distances for the head joint across 10 sample projects. Unit is cm

Figure 4.4: Frequency of 2D and 3D distances. Note that they use different units

Chapter 5

Evaluation

5.1 Pose estimation



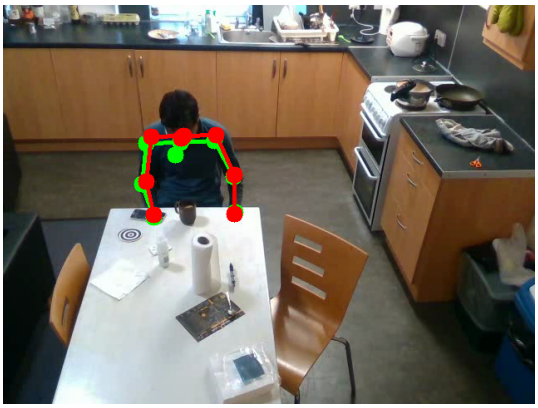


Figure 5.1: Images displaying the maximum Euclidean distance difference in head pose between the original and anonymized dataset for each project. While other poses in the images, particularly the left one, are identical between the original and anonymized datasets, the head poses show considerable differences.

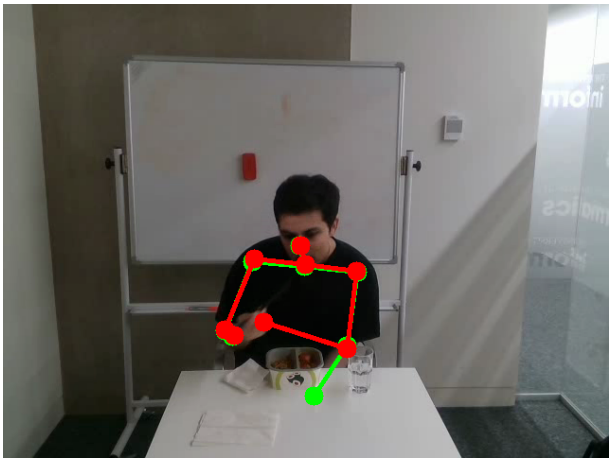


Figure 5.2: Images displaying the maximum Euclidean distance difference in left wrist pose between the original and anonymized dataset. The left wrist is located under the table, causing the anonymized pose estimations (represented by red dots and lines) to mistakenly identify the right hand as the left wrist.

Figure 5.3: A peculiar glitch occurred during the anonymization of our dataset. This particular frame did not impact the pose estimation, but we cannot be certain whether other glitches might have affected our model's performance.

We primarily focused on median values to ensure that the majority of the anonymized dataset maintains its quality, enabling other researchers to reproduce data from our dataset. In summary, the dataset is considered reliable as most median values are less than 10 pixels in 2D and 10 centimeters in 3D. However, we observed some incorrect poses, particularly for the head and left wrist. We believe that this is because people tend to move their heads frequently while eating, making estimation more challenging for these body parts. Additionally, since most participants are right-handed, their left hands are often positioned below the table, leading to failed estimations for the left wrist or significantly inaccurate markings that result in large outliers. Figure 5.1 and Figure 5.2.

5.2 Action Classification

Algorithm	Modality	Split	Top-1	Top-2	Macro Acc.	Mean Acc.
TAM	RGB	1	82.72%	93.28	71.4%	70.95%
		2	85.39%	93.82%	75.06%	73.49%
		3	84.59%	93.17%	70.3%	72.85%
		4	84.14%	93.27%	74.19%	69.86%
		5	84.67%	93.70%	71.18%	72.18%

Table 5.1: Algorithm performance for the anonymized dataset

Algorithm	Modality	Split	Top-1	Macro Accuracy
TAM	RGB	Best	86.7%	80.6%

Table 5.2: Algorithm performance for the originaldataset

The accuracy for the anonymized dataset was found to be reasonable compared to the original dataset. Top 1 accuracy did not differ much between the two datasets but macro accuracy showed 5% differences which is noticiable. We assume that the model excels at recognising majority classes, contributing to overall Top-1 accuracy, but struggles with minority classes or more difficult-to-classify situations. Because Macro accuracy computes the average of per-class accuracy while treating all classes equally, poor performance in minority or difficult classes can result in a lower Macro accuracy despite a high Top-1 accuracy. As demonstrated in Figure 5.6, the model exhibits high accuracy in classes with a large number of action counts, while its performance is lower in other classes, as shown in Table 3.1.

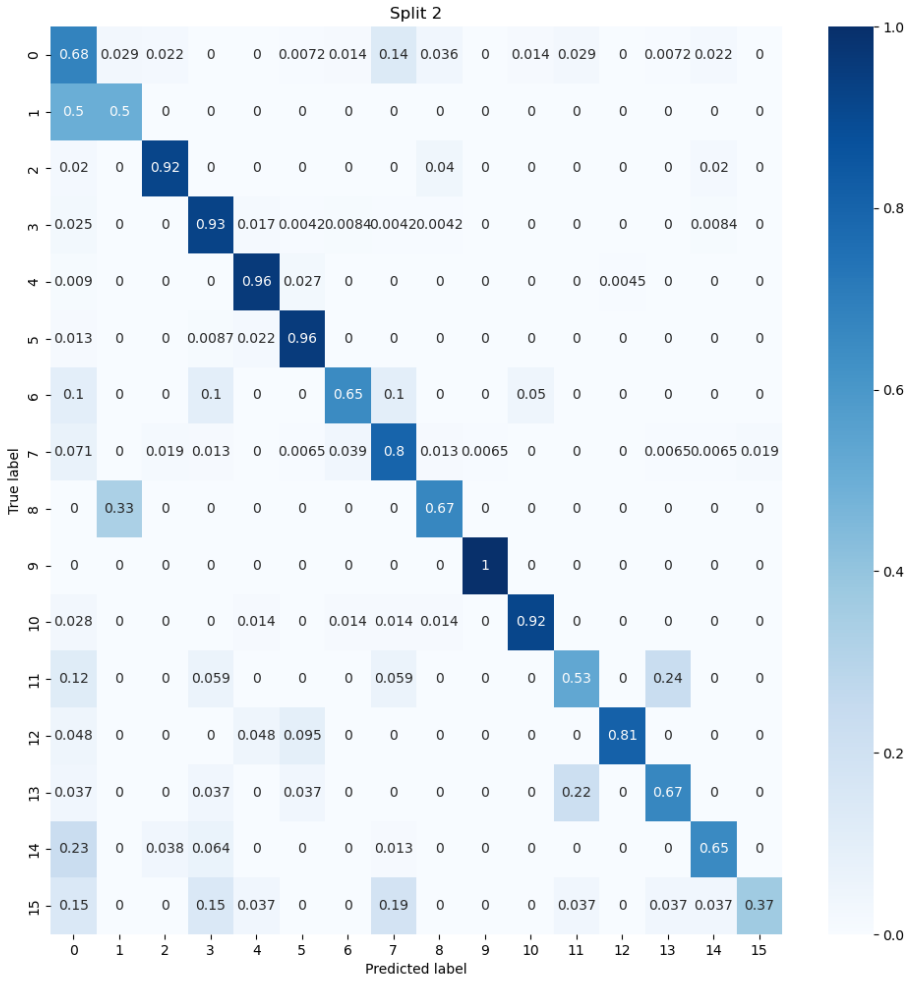


Figure 5.4: Confusion matrix for the best top 1 and macro accuracy

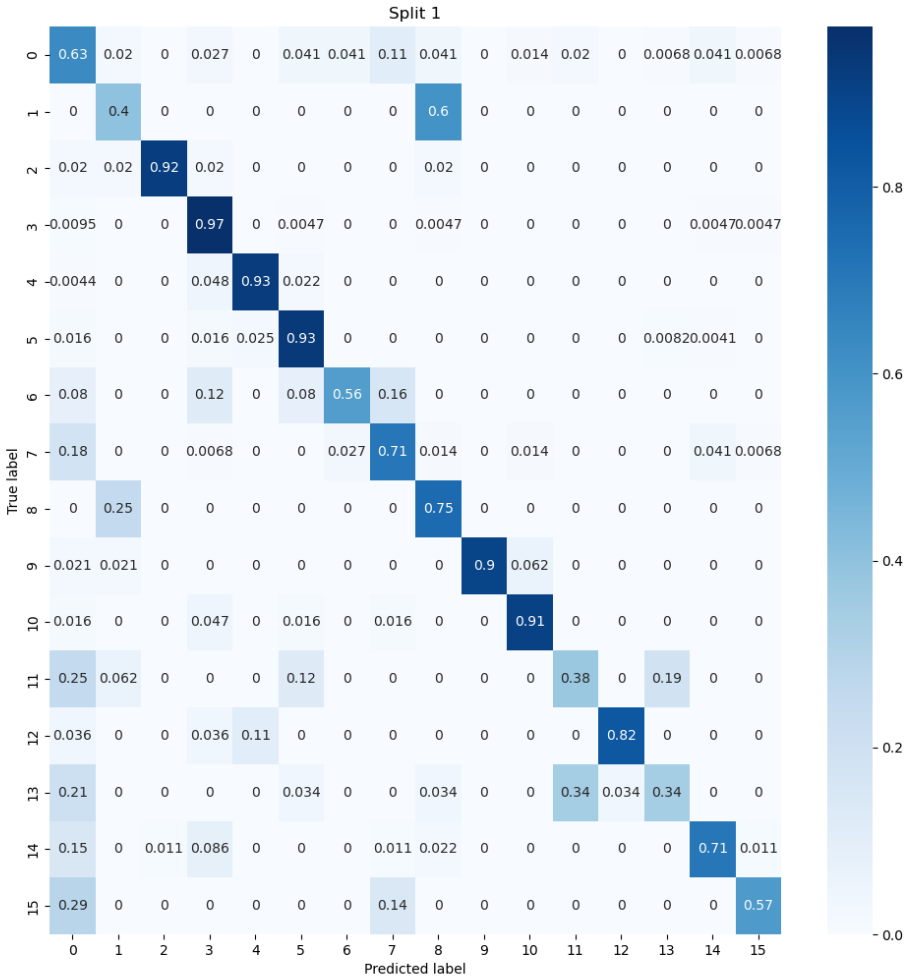


Figure 5.5: Confusion matrix for the worst top 1 accuracy

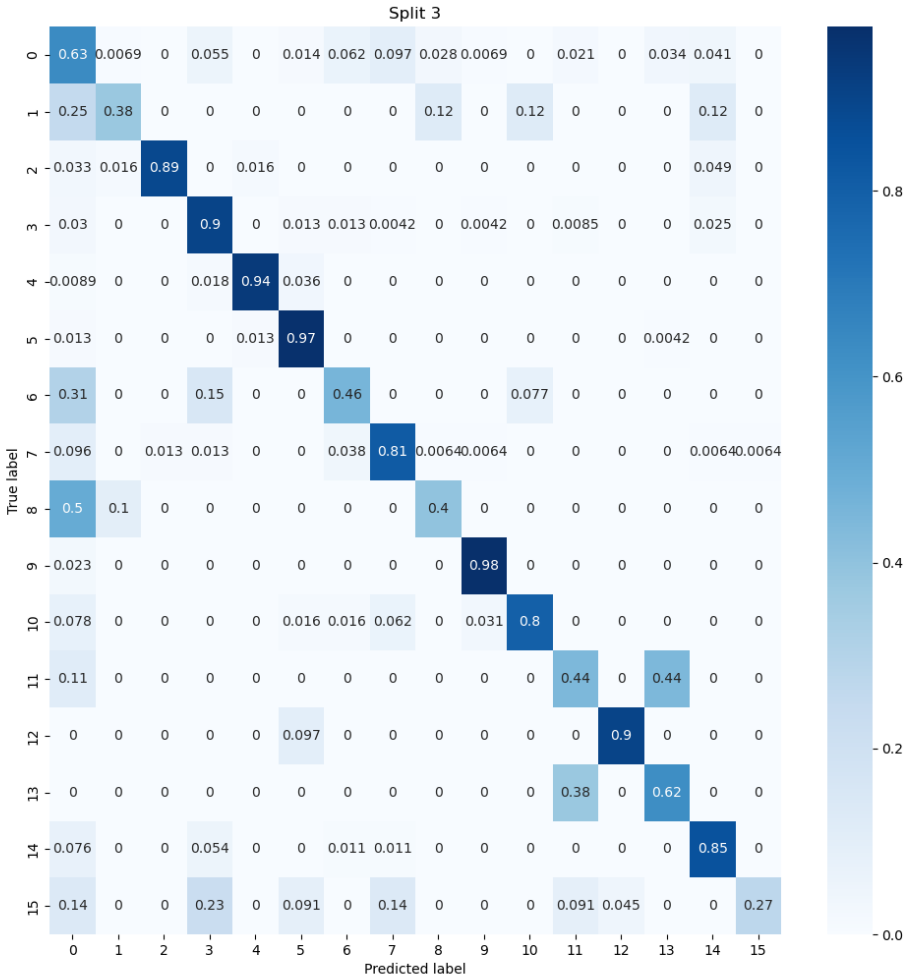


Figure 5.6: Confusion matrix for the worst macro accuracy

Figure 5.7: Training top 1 and top 2 accuracy for the worst macro accuracy

Chapter 6

Summary

In recent years, population aging has emerged as a significant challenge, leading to a shortage of medical personnel. In this project, we propose a marker-less, camera-based motion estimation system for monitoring elderly individuals' daily activities to identify any severe health issues. The adoption of this system would improve the quality of life for the elderly by eliminating the need for routine medical examinations. We recorded videos of eating gestures and anonymized them for sharing with other researchers. We then validated the anonymized dataset by comparing it to the original dataset using 2D and 3D pose estimation, as well as top 1, top 2, and macro accuracy in action recognition classification using the TAM algorithm. The quality of labeling and the instability of APIs may have an impact on our tests, necessitating further exploration in future projects.

Chapter 7

Acknowledgements

I would like to express my gratitude to Robert B. Fisher for his support and guidance throughout this project.

I also thank Muhammad Ahmed Raza for sharing his extensive technical expertise, contributing relevant scripts and helping to create the dataset, as well as Longfei Chen for providing valuable social support during stressful times.

I would also like to offer special thanks to all the people who participated in creating the dataset for this project.

and belena for problem solving

Chapter 8

Appendix

8.1 2D

8.1.1 Statistics

Link to the 10 sample 2D statistics

8.1.2 Frequency

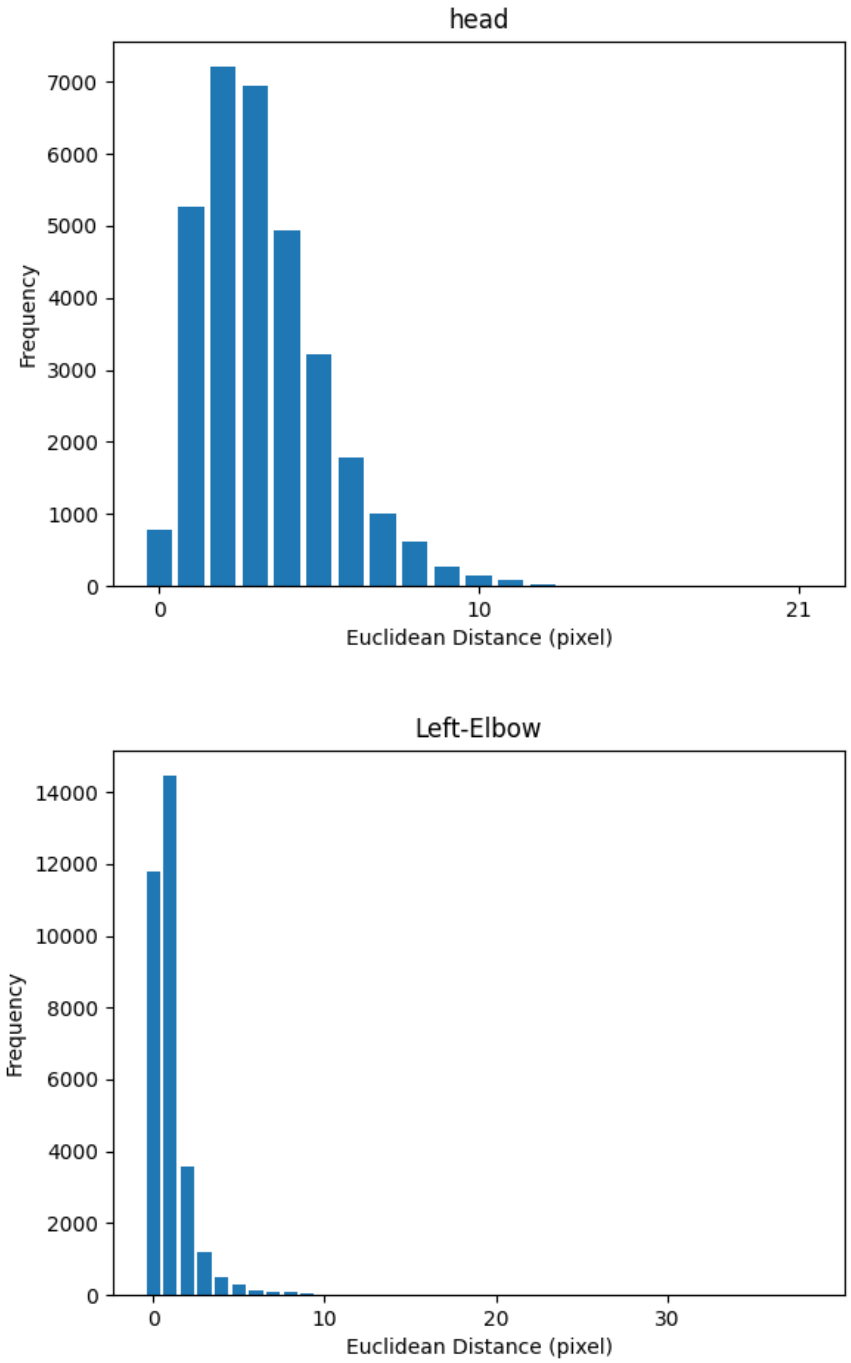


Figure 8.1: 2D frequency

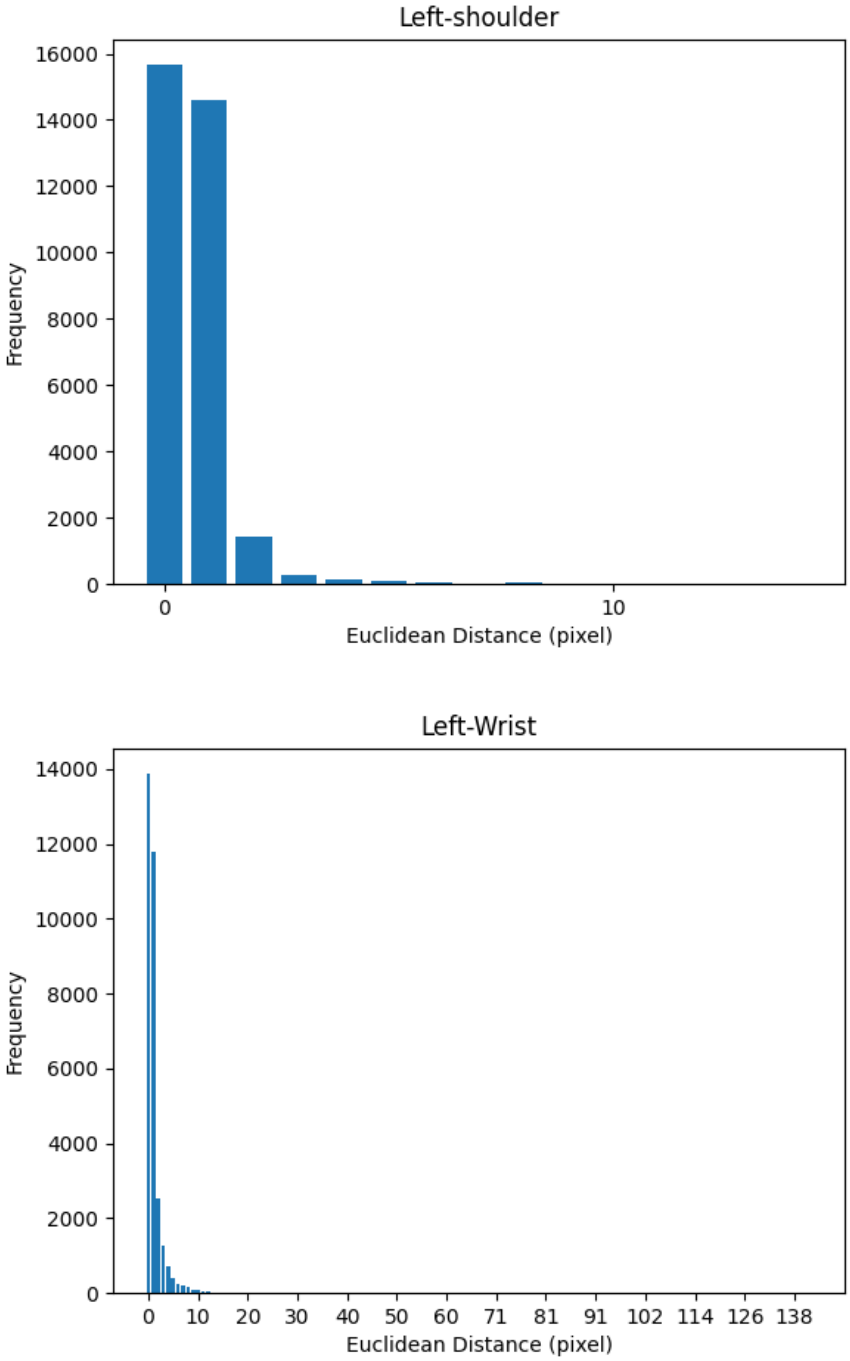


Figure 8.2: 2D frequency

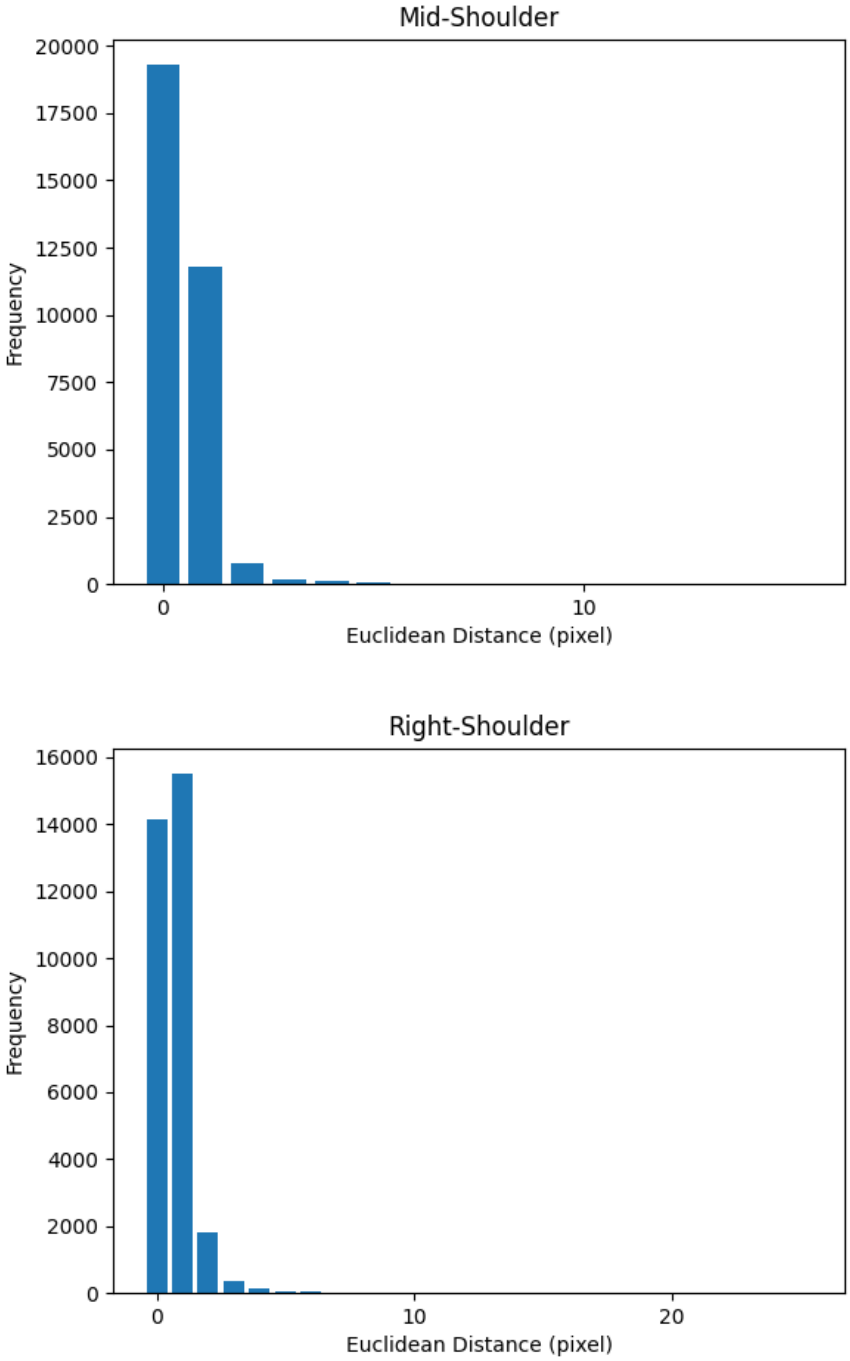


Figure 8.3: 2D frequency

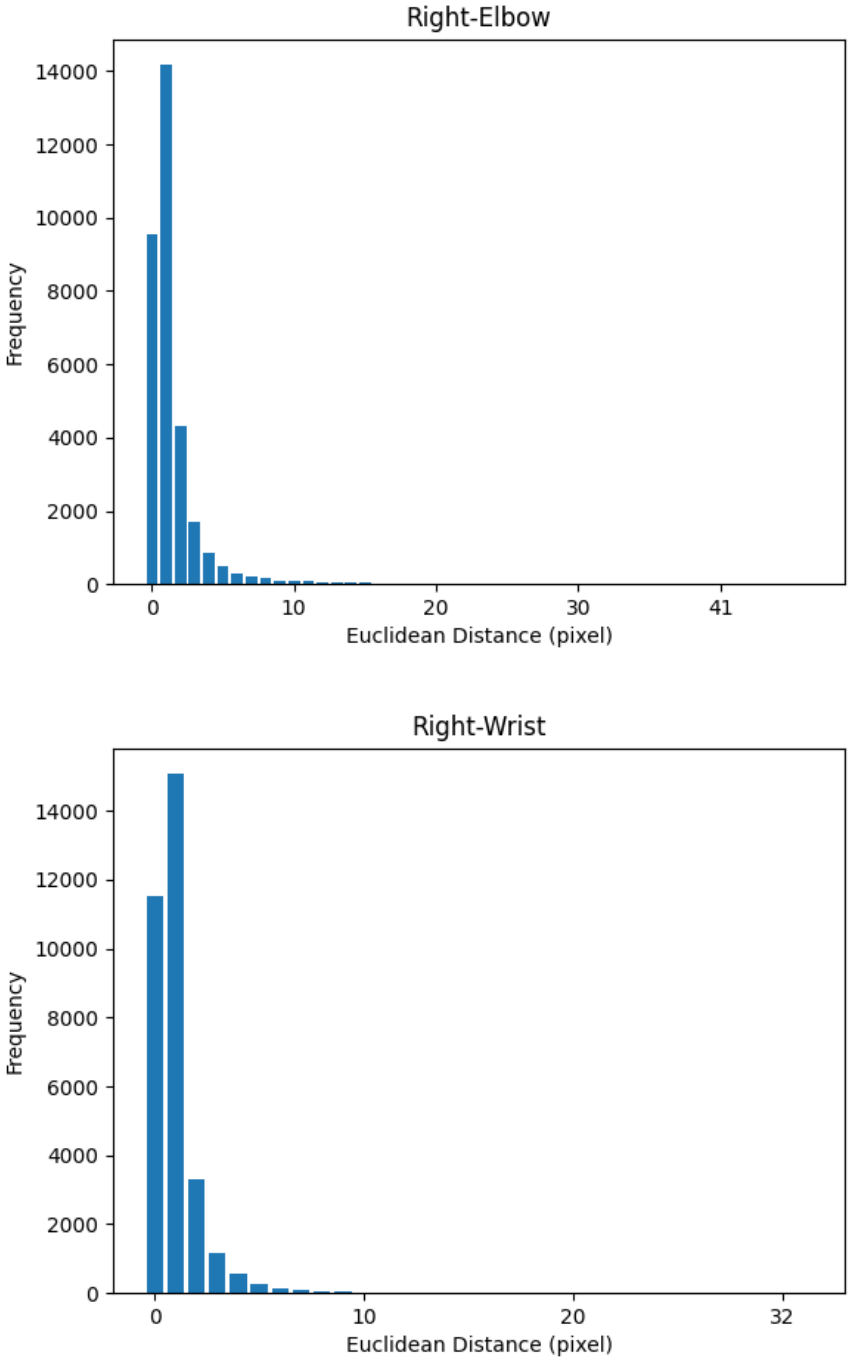


Figure 8.4: 2D frequency

8.2 3D

8.2.1 Pose videos

Link to the 10 sample 3D pose videos

8.2.2 Statistics

Link to the 10 sample 3D statistics

8.2.3 Frequency

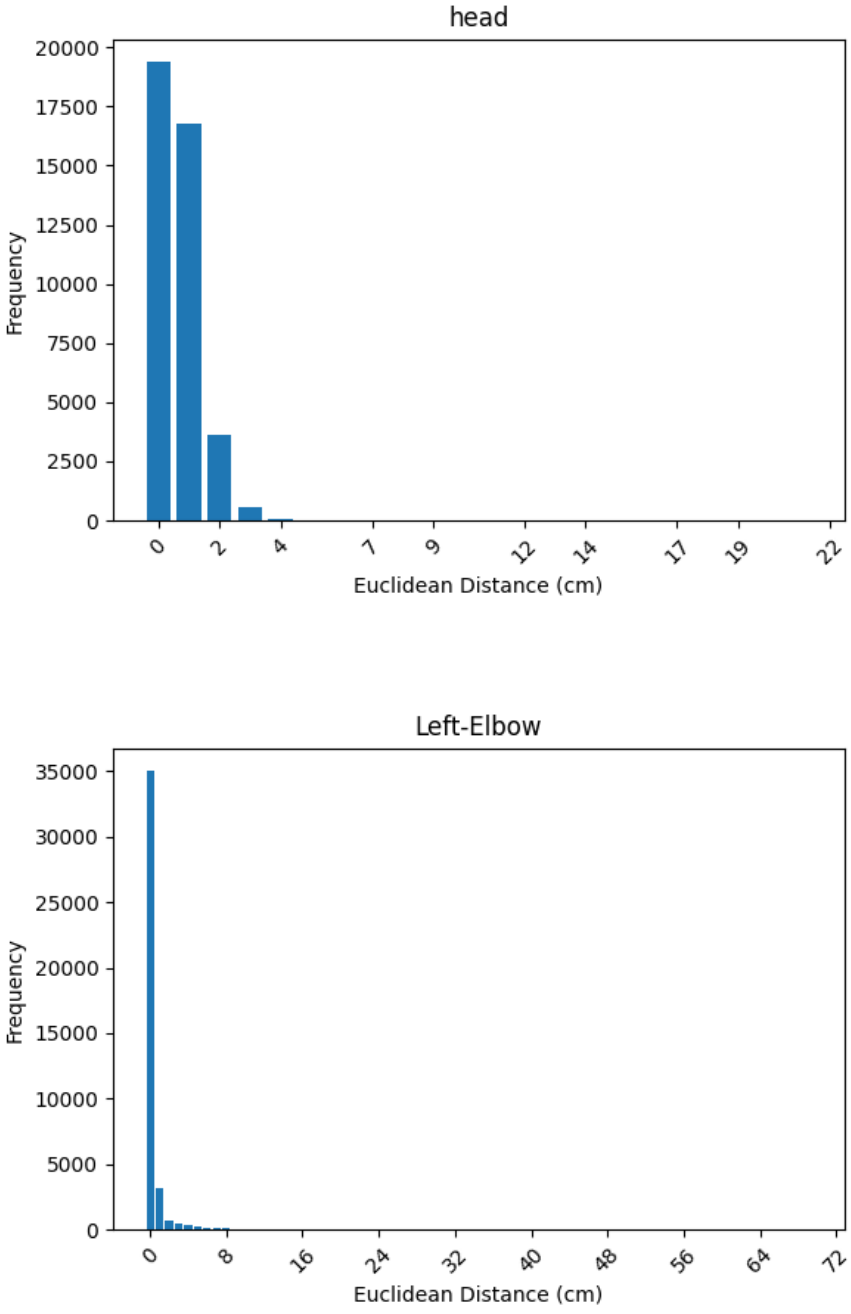


Figure 8.5: 3D frequency

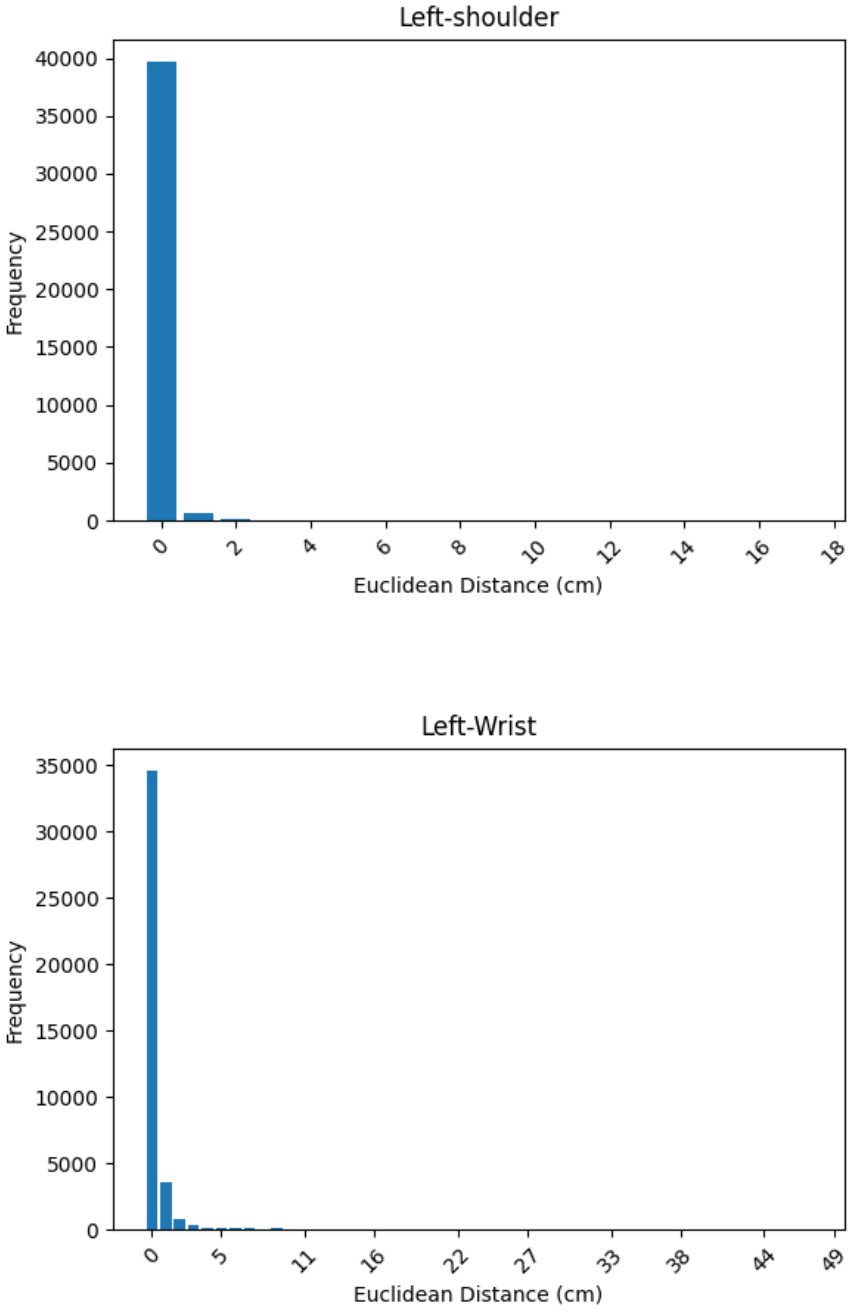


Figure 8.6: 3D frequency

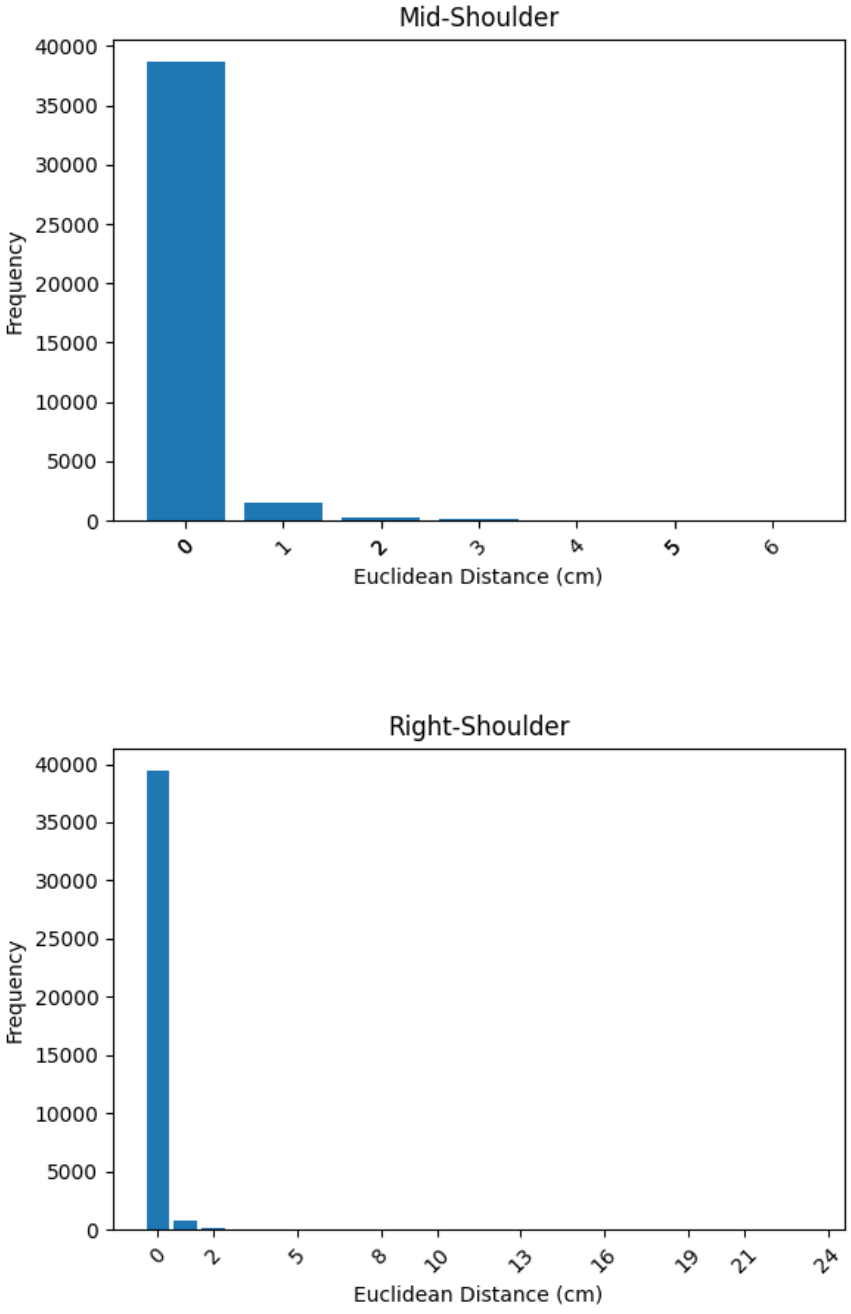


Figure 8.7: 3D frequency

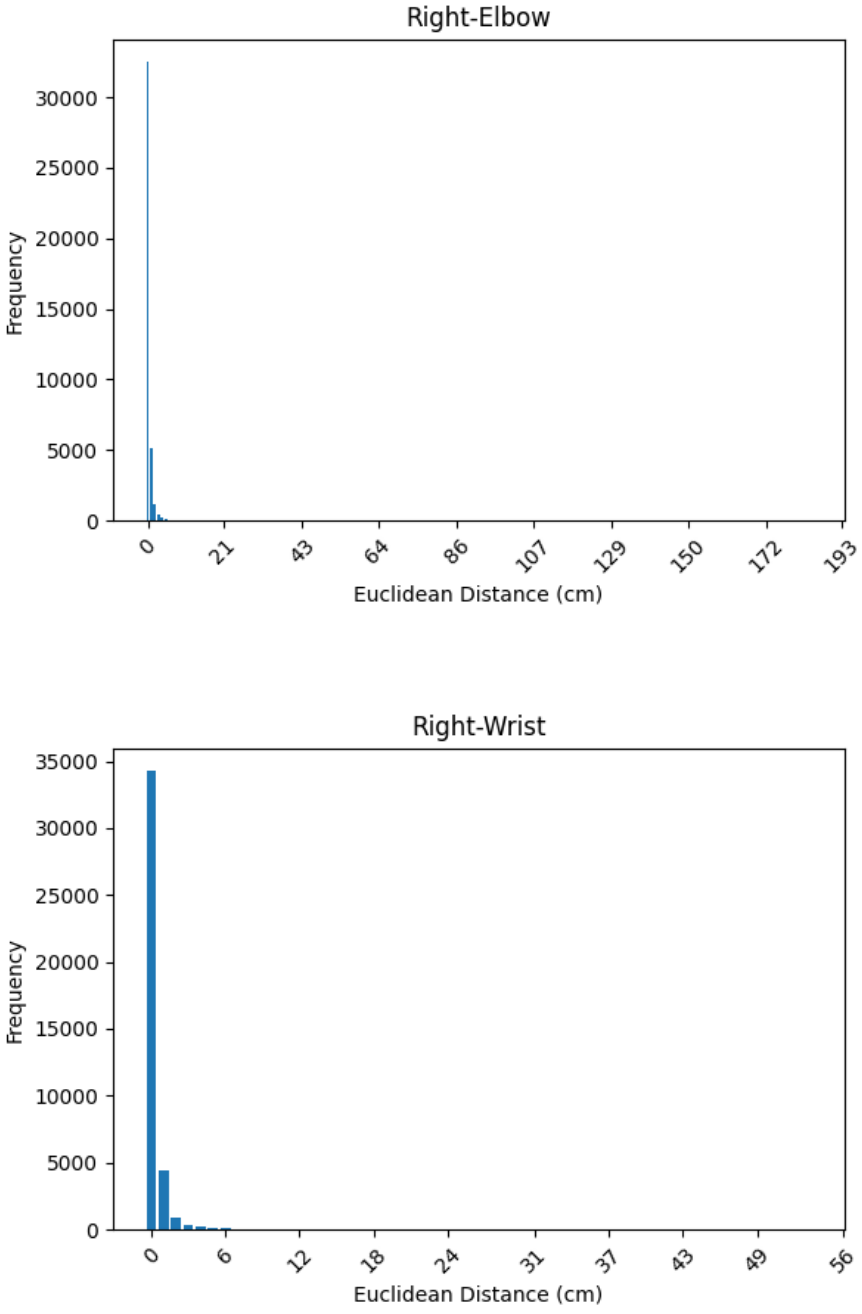


Figure 8.8: 3D frequency

Bibliography

[1] Global Population, "Demographic structure," Retrieved February 26, 2023, from https://sites.google.com/site/3oesogeography/3-the-global-population/4-demographic-structure, n.d.

[2] Information Commissioner's Office, "Ico fines facial recognition database company clearview ai inc." May 2022, accessed: February 24, 2023. [Online]. Available: https://ico.org.uk/about-the-ico/media-centre/news-and-blogs/2022/05/ico-fines-facial-recognition-database-company-clearview-ai-inc/

[3] H. Hukkelås, "Deepprivacy2 - towards realistic full-body anonymization (wacv2023 presentation)," YouTube video, January 2023, [Accessed on: February 24, 2023]. [Online]. Available: https://www.youtube.com/watch?v=wwKRkkzxKuM

[4] LearnOpenCV, "Conditional gan (cgan) in pytorch and tensorflow," https://learnopencv.com/conditional-gan-cgan-in-pytorch-and-tensorflow/, 2021, accessed: yyyy-mm-dd.

[5] M. M. B. X. D. W.-F. S. O. A. C. Y. B. Ian J. Goodfellow, Jean Pouget-Abadie, "Generative adversarial nets," vol. 78, no. 7, pp. 9101–9128, 2014.

[6] M. Mirza and S. Osindero, "Conditional generative adversarial nets," *arXiv preprint arXiv:1505.04597*, 2015.

[7] J. S. Hukkelas, R. Mester, and F. Lindseth, "Deepprivacy: A generative adversarial network for face anonymization," *arXiv preprint arXiv:1909.04538*, 2019.

[8] X. Wu, C. Wang, X. Xu, F. Huang, and Y. Wang, "Context surface encoder for temporal sentence grounding in videos," in *Proceedings of the 28th ACM International Conference on Multimedia*. ACM, 2020, pp. 1651–1659.

[9] T. Karras, S. Laine, and T. Aila, "A style-based generator architecture for generative adversarial networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[10] Y. Viazovetskyi, V. Ivashkin, and E. Kashin, "Stylegan2 distillation for feed-forward image manipulation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 138–139.

[11] T. Karras, S. Laine, and T. Aila, "Analyzing and improving the image quality of stylegan," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[12] J. S. Hukkelas, M. A. Riegler, P. Halvorsen, H. K. Stensland, K. Pogorelov, and C. Griwodz, "Deepprivacy2: Towards realistic full-body anonymization," in *2023 IEEE Winter Conference on Applications of Computer Vision (WACV)*, 2023.

[13] L. Weng, "What are diffusion models?" *lilianweng.github.io*, Jul 2021. [Online]. Available: https://lilianweng.github.io/posts/2021-07-11-diffusion-models/

[14] Y. Liu, D. Chen, B. Zhang, and W. Wang, "Tam: Temporal adaptive module for video recognition," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 2021, pp. 7311–7320.

[15] M. A. Raza, "Visual assessment of long-term changes of activity levels in elderly people," *Institute of Perception, Action and Behaviour, School of Informatics, University of Edinburgh*, 2021.

[16] A. G. Abhishek Dutta and A. Zisserman, "Vgg image annotator (via)." [Online]. Available: https://www.robots.ox.ac.uk/~vgg/software/via/

[17] M. A. Raza, L. Chen, N. Li, and R. B. Fisher, "Eatsense: Human centric, action recognition and localization dataset for understanding eating behaviors and quality of motion assessment," *The University of Edinburgh, School of Informatics*, 2023.