Master's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Medicine and Health Sciences
Department of Neuromedicine and Movement Science

Atle Skjelbred

Video-based assessment of head centered, hand-to-hand contact, and body symmetry in infants for early prediction of cerebral palsy

Master's thesis in Physical Activity and Health - Exercise Physiology
Supervisor: Espen A. F. Ihlen
Co-supervisor: Lars Adde

May 2022

NTNU
Kunnskap for en bedre verden

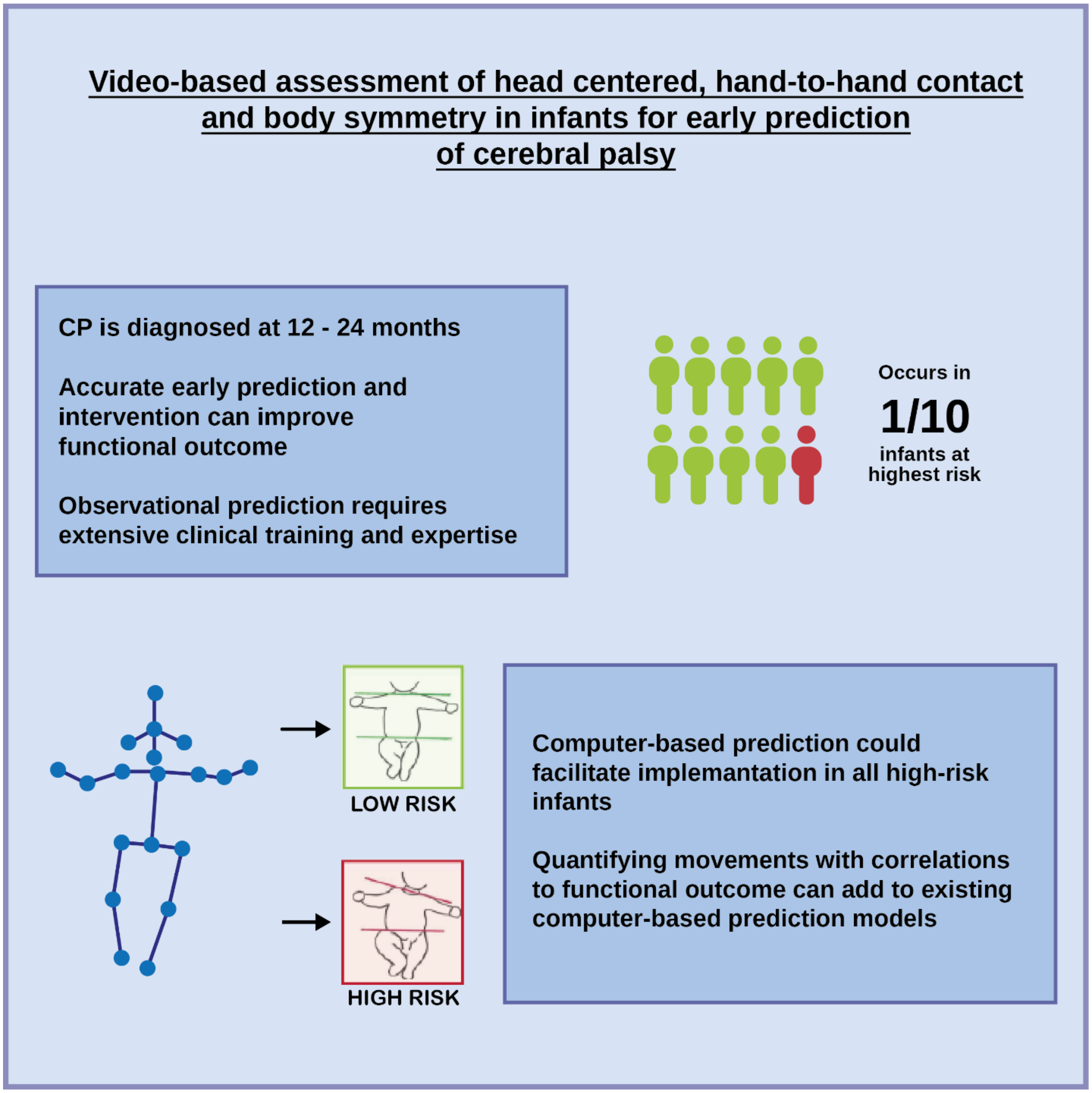
Atle Skjelbred

Video-based assessment of head centered, hand-to-hand contact, and body symmetry in infants for early prediction of cerebral palsy

Master's thesis in Physical Activity and Health - Exercise Physiology
Supervisor: Espen A. F. Ihlen
Co-supervisor: Lars Adde
May 2022

Norwegian University of Science and Technology
Faculty of Medicine and Health Sciences
Department of Neuromedicine and Movement Science

Video-based assessment of head centered, hand-to-hand contact, and body symmetry in infants for early prediction of cerebral palsy



Video-based assessment of head centered, hand-to-hand contact and body symmetry in infants for early prediction of cerebral palsy

CP is diagnosed at 12 - 24 months

Accurate early prediction and intervention can improve functional outcome

Observational prediction requires extensive clinical training and expertise

Occurs in **1/10** infants at highest risk

Computer-based prediction could facilitate implemantation in all high-risk infants

Quantifying movements with correlations to functional outcome can add to existing computer-based prediction models

Abstract (English)

Aim: The aim of this study is to develop an automatic assessment of the hand-to-hand contact, head centered, and body symmetry items of the movement optimality score. The automatic assessment is developed by producing movement features using quantifiable variables extracted from video-based motion tracking data. Additionally, analyse how accurate individual and sum of movement features differentiate and predict CP and non-CP outcomes in high-risk infants.

Method: Seven features were developed using the motion tracking data. The movement features' ability to differentiate CP and non-CP outcome groups was assessed using the area under the curve (AUC) of the receiver operating characteristic (ROC) curve, and the difference in distribution using the Mann-Whitney U test. The predictive capability in singular cases was assessed using the combined sensitivity and specificity, and the mean difference in prediction score between the CP and non-CP outcome groups.

Result: The AUC for the feature values ranged from .474 to .611. Some features displayed a difference in distribution between the CP and non-CP outcome groups. However, the difference was not statistically significant, Mann-Whitney U p = .106 to .704. The mean difference between the outcome groups for the features following conversion to prediction score was -.085 to .186 p = .031 to .474. The mean difference for the combined prediction score was .750 p = .031.

Conclusion: This study proves that an automatic assessment of this type of MOS items can be produced, and some features can differentiate between CP and non-CP outcome groups. Individual features have limited ability to predict CP outcome, however, combining features increases the predictive capability.

Abstrakt (Norwegian)

Mål: Målet med studien er å utvikle en automatisk vurdering av «hand-to-hand contact», «head centered» og «body symmetry» elementene i «movement optimality score» (MOS). Denne automatiske vurderingen oppnås gjennom produksjon av bevegelsestrekk basert på kvantifiserbare variabler hentet fra videobasert bevegelsessporingsdata. I tillegg analysere hvor nøyaktig individuelle- og summen av bevegelsestrekk skiller- og forutsier CP- og ikke-CP-utfall hos høyrisiko spedbarn.

Metode: Syv bevegelsestrekk ble utviklet med bevegelsessporingsdataen. Bevegelsestrekkenes evne til å skille CP- og ikke-CP-utfallsgruppene ble vurdert med arealet under kurven (AUC) av receiver operating characteristic (ROC) kurven, og forskjellen i distribusjon mellom gruppene testet med Mann-Whitney U-test. Den prediktive evnen i enkeltindivider ble vurdert med kombinasjonen av sensitiviteten og spesifisiteten, i tillegg til forskjellen i gjennomsnitt av prediksjonsscoren mellom CP- og ikke-CP-utfallsgruppene.

Resultat: AUC for bevegelsestrekkverdiene varierte fra .474 til .611. Noen bevegelsetrekk viste en forskjell mellom CP- og ikke-CP-utfallsgruppene, men forskjellen var ikke statistisk signifikant, Mann-Whitney U p = .106 til .704. Forskjellen mellom utfallsgruppene da bevegelsestrekkene ble konvertert til en prediksjonsscore var -.085 til .186 p = .031 til .474, og forskjellen mellom utfallsgruppene for den samlede prediksjonsscoreen var .750 p = .031.

Konklusjon: Denne studien viser at en automatisk vurdering av denne typen MOS elementer er mulig, og noen bevegelsestrekk evner å skille CP- og ikke-CP-utfallsgruppene. Individuelle bevegelsestrekk har en begrenset evne til å forutsi CP-utfall, men prediksjonsevnen øker ved å kombinere bevegelsestrekk.

1 - Introduction

Cerebral palsy (CP) is a term used to describe multiple permanent non-progressive disorders affecting movement, posture, and activity. The condition is caused by damage to the brain or abnormal brain development during pregnancy, birth, or early infancy (1). The prevalence of CP as of 2013 in the general population was 2.1 per 1000 (2). There are many known high-risk factors for the development of CP, including but not limited to; low birth weight, shortened gestational age, and neonatal encephalopathy (2). The group at highest risk is infants with a gestational age of fewer than 28 weeks, with a prevalence of 112 per 1000 (2). A clinical diagnosis of CP is done based on both clinical and neurological signs, most commonly when the child is between 12 and 24 months of age (3). The gross motor function classification system (GMFCS) is commonly used in conjunction with a CP- and subtype-diagnosis as a way of describing the degree of disability an individual case exhibits. The scoring system ranges from I to V, where someone with GMFCS I might only have some limitations with advanced motor skills, while GMFCS V has no means for independent mobility (4). Accurate early prediction of CP could facilitate earlier implementation of interventions that may have a positive influence on the child's development as the plasticity of the brain is at its highest (5,6). Possibly improving the outcome of motor function, cognitive skills, communication, and more (6). In addition, due to the elevated risk of mental health problems in parents, interventions targeting parent-infant interactions and cognitive behavioural therapy may have a positive influence on parental mental health and parent-infant relationships (6).

The current most accurate method of assessing the risk of CP development before 5 months of age is observational general movement assessment (GMA) in conjunction with cerebral imaging (MRI) (3). However, MRI requires expensive equipment leaving GMA the most suitable low-cost method for all infants at risk (3). GMA consists of classifying an infant's fidgety movements as normal, abnormal, or absent based on a 3-to-5-minute video recording. This video recording needs to be conducted under certain conditions for reliable analysis. The infant must lie on its back, awake, content, and without any distractions (7). The fidgety movements are only present during a specific period, from about 6 to 9 weeks to 16 to 20 weeks corrected age (CA), before conscious movements become predominant (8). The absence of fidgety movements during this period is highly predictive of the development of CP (9,10). A summary of a small selection of observational GMA studies from the past ten years is displayed in table 1.

Table 1: A summary of previous studies using Prechtl's GMA method for observational prediction of CP in infants.

Study	Sample Size [1]	Sensitivity (%)	Specificity (%)	Accuracy (%)
Øberg (8)	87 (10)	100	70	74
Dimitrijevic (11)	79 (11)	100	85	87
Crowle (12)	202 (11)	100	92	93
Brogna (13)	574 (22)	100	97	97
Støen (14)	405 (42)	100	85	87

[1] Sample size and the number of infants with later CP diagnosis in parenthesis.

The movement optimality score (MOS) expands upon the GMA, evaluating the FMs, movement character, age-adequate movement repertoire, observed movement patterns, and observed postural patterns (15–18). These five categories total a score from 5 to 28, where 28 is optimal (15). Previous research found that although 95% of children who later developed CP did not exhibit FMs, 100% had a non-optimal MOS. They also found a strong correlation between the MOS and functional outcome with GMFCS (15). Both the GMA and MOS require extensive clinician training and experience, making the widespread application of the methods for all high-risk infants unrealistic (19). The subjective nature of the observational methods can also be a source of inaccuracies (19). Research on the inter-observer reliability for FM assessment found high agreement when GMA observers differentiate normal and non-normal FMs, but the experience of the observer was important. Additionally, once FMs are further divided into sub-categories like intermittent FMs, continuous FMs, etc, the inter-observer agreement is reduced (20).

Due to the challenges associated with observational GMA and MRI as early prediction methods for CP, the development of low-cost automated assessment of CP has received considerable attention in recent years (21,22). Studies have explored various methods for gathering movement data with 3D motion tracking (23), wearable movement sensors, and more (22), but the most feasible method thus far is video capture (21). Video capture has the advantage of not influencing the infants' movements, being non-intrusive, not reliant on specialised equipment, and easily applicable considering the current state of video- and smartphone cameras. Several studies using video capture have been conducted with various means for analysis. A summary of these studies is displayed in table 2. Most previous quantitative studies attempt to classify the FMs in the infant's movements similarly to observational GMA. This can be challenging due to the complexity and variability observed in infant's spontaneous movements, as well as the variability of movement over time throughout a recording (24). Including GMA experts to classify and annotate normal and abnormal epochs in GMA videos for machine learning algorithms would reintroduce the inter-observer reliability as a potential confounder (20).

Table 2: Summary of previous studies using video-based automated movement analysis for prediction of CP in infants. An expanded version of table 1 in Ihlen et al. 2020.

Study	Sample Size [1]	Sensitivity (%)	Specificity (%)	Accuracy (%)
Adde (25)	30 (13)	85	88	87
Rahmati (26)	78 (14)	50	95	87
Rahmati (27)	78 (14)	86	92	91
Stahl (28)	82 (15)	85	95	94
Orlandi (29)	127 (16)	44	99	92
Ihlen (30)	377 (41)	93	84	85

[1] Sample size and the number of infants with later CP diagnosis in parenthesis.

The development of video-based motion tracking systems allows for the quantification of specific movement characteristics within the overall movements of the infant (31,32). Additionally, vast amounts of video recordings with corresponding CP outcomes are becoming available through research and clinical implementation of observational GMA (14,29,33,34). These developments could permit a different approach to automated early prediction of CP in high-risk infants that were not previously available. Facilitating the development of CP prediction models investigating specific individual movement- and postural characteristics within the infants' spontaneous movements. This can allow for the quantification of items within observed movement patterns and observed postural patterns categories of the MOS, and possibly the discovery and development of additional features previously undetectable with observational GMA. The observed movement- and postural patterns items in the MOS can be divided into two main categories, 1: Items with compound movements consisting of complex changes of speed and direction in specific limbs or body parts requiring several subsequent frames of the video recording to detect, like the kicking and swipes items. 2: Items with a binary state which can be extracted and evaluated on any singular frame of the video recording like the head centered and body symmetry items (15). The complexity of items in the first category likely requires annotation and machine learning to quantify. Due to this, only items of the second category were considered in this study. Additionally, items based on the mouth-, tongue-, and eye movements were not included as the motion tracking data utilised do not provide movement information on these body parts.

The aim of this study is to develop an automatic assessment of the hand-to-hand contact, head centered, and body symmetry items of the movement optimality score. The automatic assessment is developed by producing movement features using quantifiable variables extracted from video-based motion tracking data. Additionally, analyse how accurate individual and sum of movement features differentiate and predict CP and non-CP outcomes in high-risk infants.

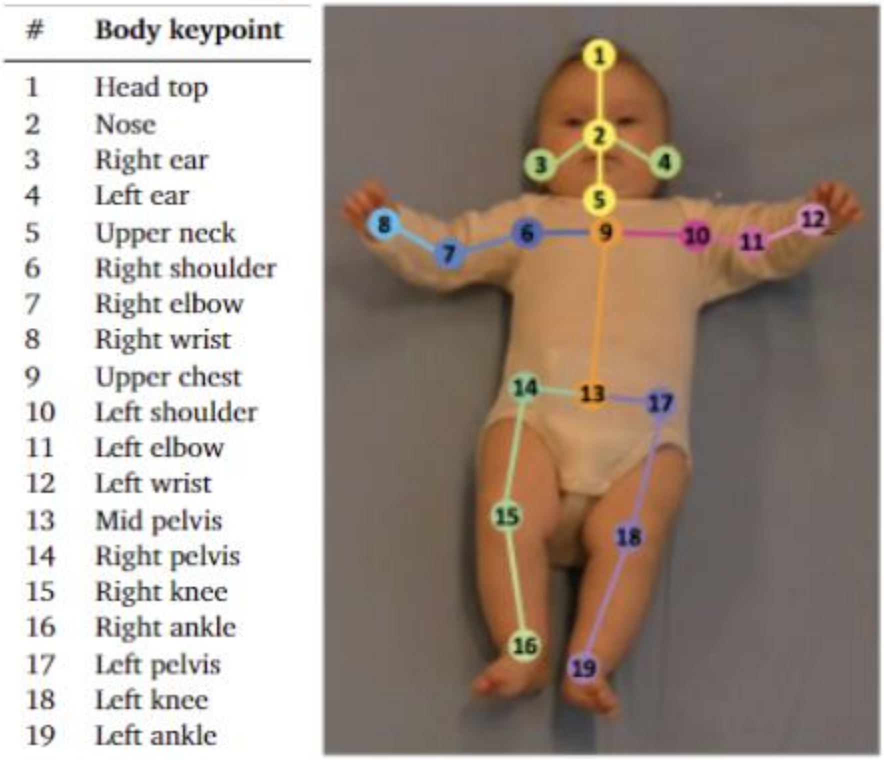
2.0 - Method

2.1 - Study participants

This study adopts a pre-existing dataset consisting of 557 high-risk infants with known CP or non-CP outcomes. The participants were prospectively enrolled in various studies between September 2001 and October 2018 (14,25,35,36). The participants originate from four countries, the United States (248 infants), Norway (190 infants), India (82 infants), and Belgium (37 infants), each with a specific composition of high-risk factors. The infants originating from the United States and Norway consisted of heterogeneous high-risk factors including short gestational age, low birth weight, neurological abnormalities etc. All infants from India had neonatal encephalopathy and all infants from Belgium had perinatal stroke. All participants were video recorded at 9-18 weeks CA during the FMs period, according to Prechtl's GMA standards, and considered for diagnosis between 12- and 89-months CA by a paediatrician. Parental consent for all infants was obtained before inclusion in the respective studies (32).

2.2 - Motion tracking data

This study utilises the motion tracking data for the high-risk infants outlined in the previous paragraph, the motion tracking data was developed by a separate study (31,32). The motion tracking data consists of the x and y coordinates of 19 body keypoints and is normalised to 30 frames per second. The name, number, and location of all body keypoints and a visual example of the location are outlined in figure 1.



#	Body keypoint
1	Head top
2	Nose
3	Right ear
4	Left ear
5	Upper neck
6	Right shoulder
7	Right elbow
8	Right wrist
9	Upper chest
10	Left shoulder
11	Left elbow
12	Left wrist
13	Mid pelvis
14	Right pelvis
15	Right knee
16	Right ankle
17	Left pelvis
18	Left knee
19	Left ankle

Figure 1: Name, number, and location of all body keypoints in the video-based motion tracking data. The list and visual adapted from Groos, Adde, et al. 2022 (31).

2.3 - MOS item quantification

Each MOS item selected for this study was assessed to find suitable variables capable of extracting the movement characteristics outlined in a previous definition of the item (15). The features selected were based on the definition used by Einspieler 2019 if the evaluation criteria of the MOS item were sufficiently explained and the body keypoints in the motion tracking data provided the necessary movement information. The original definition, variables selected, and features selected for head centered-, body symmetry- and hand-to-hand contact is displayed in table 3, 4, and 5 respectively.

Head centered

Table 3: Displays the original definition, variables selected, and features selected for head centered item of the MOS.

Original definition		Variables		Features
The head can be kept centered for at least 10 s; chin and sternum are in one line. Score atypical if the head cannot be centered, i.e., is tilted or in lateral position.	→	<u>Head centered angle:</u> The angle between the infant's head and body. <u>Nose-to-centre distance:</u> The distance from the infant's nose to the centre of the infant's head.	→	<u>Head centered time:</u> Percentage of the motion tracking data where both feature variables are less than a specified threshold. <u>Head centered percent:</u> Length of the longest segment of the motion tracking data where both feature variables are less than a specified threshold.

Head centered variable calculation

The variable head centered angle was calculated using regression lines representing the head and body of the infant. The head regression line is composed of the head top (number 1) and upper neck (number 5) body keypoints. The body regression line is composed of the upper chest (number 9) and mid pelvis (number 13) body keypoints. The angle between the two regression lines was calculated using the slope of the regression lines using equation 1. The absolute value of the angle is utilised as the feature only measures the degree of asymmetry.

$$Head\ centered\ angle = \tan^{-1}\left|\frac{Slope_{Body} - Slope_{Head}}{1 + Slope_{Body} * Slope_{Head}}\right| \qquad 1$$

The variable nose-to-centre distance is calculated using the infant's nose body keypoint and the head regression line to represent the centre of the infant's head. A tracking point called x head centre was created where the head regression line intercepts the y value of the nose body keypoint. The y value of the tracking point equals the y value of the nose body keypoint and the x value is calculated using the y value of the nose body keypoint, slope of the head regression line, and the intercept of the head regression line using equation 2. The nose-to-centre distance was calculated in the x-axis by subtracting the x head centre value from the x nose value using equation 3. An illustration of the head centered variables in the context of the motion tracking data is displayed in figure 2. The segment of python code executing the head centered angle and the nose-to-centre distance calculation is displayed in figure 3. See appendix B for full code implementation.

$$X_{Head\ centre} = \frac{(Y_{Nose} - Intercept_{Head})}{Slope_{Head}} \qquad 2$$

$$Nose - to - centre\ distance\ = |X_{Nose} - X_{Head\ centre}| \quad 3$$

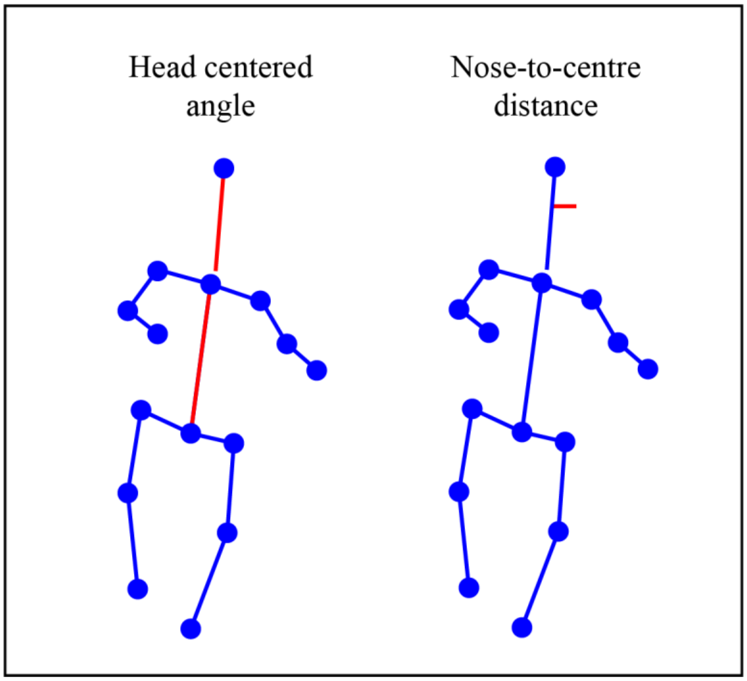
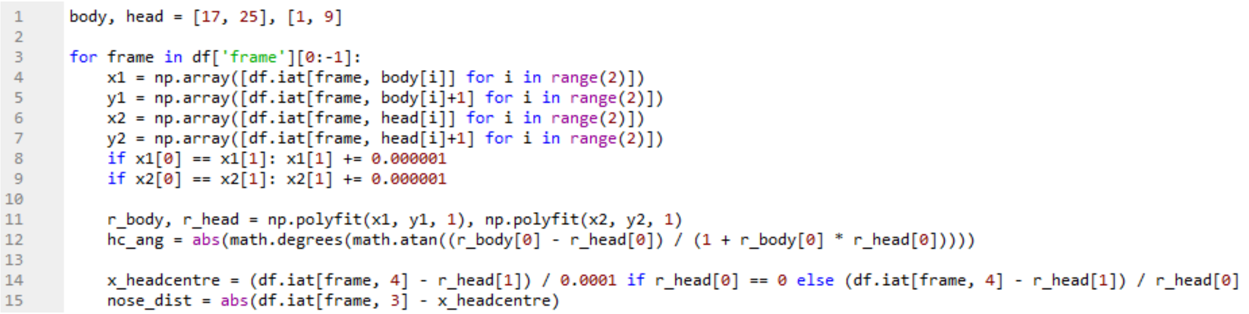


Figure 2: Illustration of the head centered variables. Head centered angle (left), with regression lines used for the angle calculation highlighted in red. Nose-to-centre distance (right), the distance measured between the nose body keypoint, and the head centre tracking point highlighted in red.



```
1       body, head = [17, 25], [1, 9]
2
3       for frame in df['frame'][0:-1]:
4           x1 = np.array([df.iat[frame, body[i]] for i in range(2)])
5           y1 = np.array([df.iat[frame, body[i]+1] for i in range(2)])
6           x2 = np.array([df.iat[frame, head[i]] for i in range(2)])
7           y2 = np.array([df.iat[frame, head[i]+1] for i in range(2)])
8           if x1[0] == x1[1]: x1[1] += 0.000001
9           if x2[0] == x2[1]: x2[1] += 0.000001
10
11          r_body, r_head = np.polyfit(x1, y1, 1), np.polyfit(x2, y2, 1)
12          hc_ang = abs(math.degrees(math.atan((r_body[0] - r_head[0]) / (1 + r_body[0] * r_head[0]))))
13
14          x_headcentre = (df.iat[frame, 4] - r_head[1]) / 0.0001 if r_head[0] == 0 else (df.iat[frame, 4] - r_head[1]) / r_head[0]
15          nose_dist = abs(df.iat[frame, 3] - x_headcentre)
```

Figure 3. The segment of python code used to extract the head centered variables from the motion tracking data with other unrelated elements of the code removed.

Figure 3 code explanation

1: Creates two separate lists containing the indexes of the x values of the body keypoints used to calculate the body regression line, and the head regression line.

3: Establishes a loop through the lines in the data frame, a particular segment, or the full motion tracking data. Each line of the data frame contains the motion tracking data from one frame of the recording. All variables for all features are calculated within the same loop in the full code.

4 – 7: Greatest NumPy arrays containing the coordinate values of the body keypoints used to calculate the regression lines. Separate arrays for the x and the y values for each regression line.

Indexes of the x values are stated in the lists from line one, and indexes of the y values are the index of the corresponding x index plus one.
8 & 9: Checks for and adjusts the edge cases where the x values of both body keypoints of the regression line are identical causing a problem with the calculation of the regression lines.
11: Calculates the slope and intercept of the regression lines using the NumPy.polyfit function. The variable r_body is the body regression line, and r_head is the head regression line. The variable name + [0] contain the slope, and the variable name + [1] contain the intercept.
12: Calculates the angle between the body and head regression line using equation 1. The angle is simultaneously converted from radians to degrees, and to the absolute value. The value is stored in the variable hc_ang.
14: Calculates the x value of the head centre tracking point using equation 2. Value is stored in the variable x_headcentre. The if statement is needed to act in edge cases where the slope of the regression line is exactly zero, performing the calculation with 0.0001 instead of the slope to avoid division by zero error. The variable df.iat[frame,4] contains the y value of the nose body keypoint in the current line in the motion tracking data. The value is stored in the variable x_headcentre.
15: Calculates the absolute distance between the x_centre and df.iat[frame,3]. The variable df.iat[frame,3] contains the x value of the nose body keypoint in the current line of the motion tracking data (equation 3). The value is stored in the variable nose_dist.

Body symmetry

Table 4: Displays the original definition, variables selected, and features selected for head centered item of the MOS.

Original definition		Variables		Features
An imaginary line through the shoulder joints and an imaginary line through the hip joints run parallel. Score atypical if this is not the case throughout the recording.	→	Body symmetry angle: The angle between one line spanning the infant's shoulders and a second line spanning the infants' hips.	→	Body symmetry time: Percentage of the motion tracking data where the feature variable value is lower than a specified threshold. Body symmetry percent: Length of the longest segment of the motion tracking data where the feature variable value is lower than a specified threshold. Body symmetry large asymmetry percent: Percentage of the motion tracking data where the feature variable value is higher than a specified threshold.

Body symmetry variable calculation

The body symmetry angle was calculated using two regression lines. One represents the infants' shoulders, and the other represents the infants' hips. The shoulder regression line is composed of body keypoint 6, 9, and 10 and the hip regression line is composed of body keypoint 13, 14, and 17. The angle between the two regression lines was calculated with the slope of the regression lines using equation 4. The absolute value of the angle is utilised as the feature only measures the degree of asymmetry. An illustration of the two regression lines in the context of the motion tracking data is displayed in figure 4. The segment of python code executing the body symmetry variable calculation is displayed in figure 5. See appendix B for full code implementation.

$$Body\ symmetry\ angle = \tan^{-1}\left|\frac{Slope_{Shoulder} - Slope_{Hip}}{1 + Slope_{Shoulder} * Slope_{Hip}}\right| \qquad 4$$

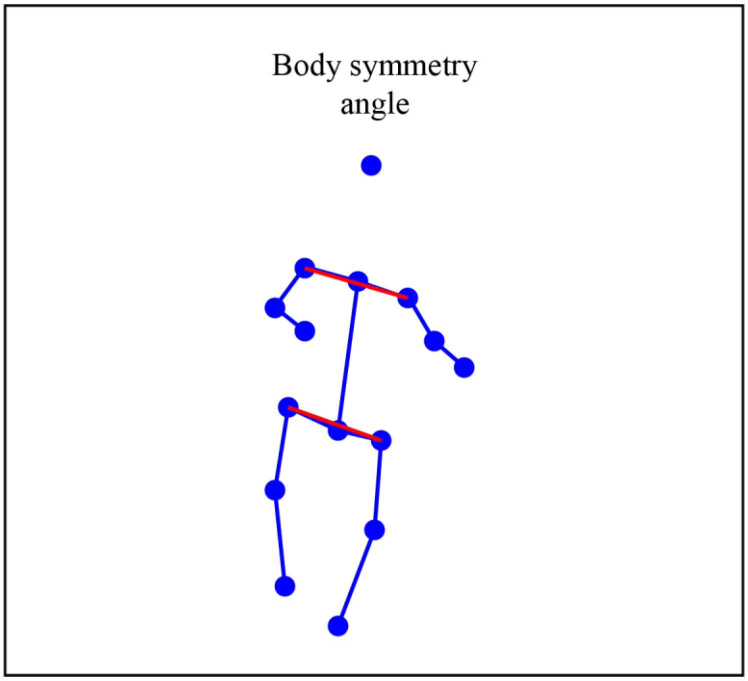
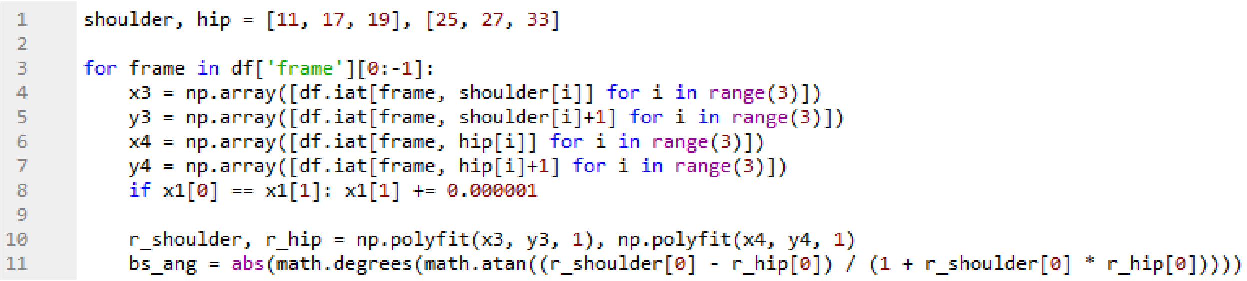


Figure 4: Illustration of the body symmetry variable with the regression lines used to calculate the angle highlighted in red.



```
shoulder, hip = [11, 17, 19], [25, 27, 33]

for frame in df['frame'][0:-1]:
    x3 = np.array([df.iat[frame, shoulder[i]] for i in range(3)])
    y3 = np.array([df.iat[frame, shoulder[i]+1] for i in range(3)])
    x4 = np.array([df.iat[frame, hip[i]] for i in range(3)])
    y4 = np.array([df.iat[frame, hip[i]+1] for i in range(3)])
    if x1[0] == x1[1]: x1[1] += 0.000001

    r_shoulder, r_hip = np.polyfit(x3, y3, 1), np.polyfit(x4, y4, 1)
    bs_ang = abs(math.degrees(math.atan((r_shoulder[0] - r_hip[0]) / (1 + r_shoulder[0] * r_hip[0]))))
```

Figure 5. The segment of python code used to extract the body symmetry variables from the motion tracking data with other unrelated elements of the code removed.

Figure 5 code explanation

1: Creates separate lists containing the indexes of the x values of the body keypoints used to calculate the regression lines.

3: Establishes a loop through the lines in the data frame, a particular segment, or the full motion tracking data. Each line of the data frame contains the motion tracking data from one frame of the recording. All variables for all features are calculated within the same loop in the full code.

4 - 7: Greatest NumPy arrays containing the coordinate values of the body keypoints used to calculate the regression lines. Separate arrays for the x and the y values for each regression line. Indexes of the x values are stated in the lists from line one, indexes of the y values are the index of the corresponding x index plus one.

8: Checks for and adjusts for edge cases where the x values of both body keypoints of the regression line are identical causing a problem with the calculation of the regression lines.

10: Calculates the slope and intercept of the regression lines using the NumPy.polyfit function. The variable r_shoulder is the shoulder regression line, and r_hip is the hip regression line. The variable name + [0] contain the slope, and the variable name + [1] contain the intercept.

11: Calculates the angle between the shoulder and hip regression lines using equation 4. The angle is simultaneously converted from radians to degrees, and to the absolute value. The value is stored in the variable bs_ang.

Hand-to-hand contact

Table 5: Displays the original definition, variables selected, and features selected for head centered item of the MOS.

Original definition		Variables		Features
Both hands are brought together in the midline and the fingers of both hands repetitively touch, stroke or grasp each other. Score atypical if asymmetrical, or if both hands are fisted.	→	Hand-to-hand distance: The distance between the right and left wrist body keypoints. Average hand-to-centre distance: The distance from the average wrist position to the centre of the infant's body.	→	Hand-to-hand time: Percentage of the motion tracking data where both feature variable values are lower than the specified threshold. Hand-to-hand percent: Length of the longest segment of the motion tracking data where both feature variable values are lower than the specified threshold.

Hand-to-hand contact variable calculation

The variable hand-to-hand distance is the hypotenuse of the distance between the x values and the y values of the wrists (body keypoints 8 and 12) using equation 5.

$$Hand - to - hand\ distance\ = \sqrt{\left(X_{Right}\ -\ X_{Left}\right)^2\ +\ \left(Y_{Right}\ -\ Y_{Left}\right)^2} \qquad 5$$

To calculate the average hand-to-centre distance variable a new tracking point was created to represent the average position of both wrists, this was achieved using equations 6 and 7. A tracking point called x body centre was created where the body regression line intercepts the y value of the average hand position. The y value of x body centre equals the y value of the average hand position, and the x value of x body centre is calculated using the y value of the average hand position, slope of the body regression line, and the intercept of the body regression line using equation 8. The average hand-to-centre distance variable was then calculated by subtracting the x body centre value from the x value of the average hand position as outlined in equation 9. An illustration of the hand-to-hand distance and average hand-to-centre distance is displayed in figure 6. The segment of python code executing the hand-to-hand contact item variable calculation is displayed in figure 7. See appendix B for full code implementation.

$$X_{Average\ hand} = \frac{\left(X_{Right}\ +\ X_{Left}\right)}{2} \qquad 6$$

$$Y_{Average\ hand} = \frac{\left(Y_{Right}\ +\ Y_{Left}\right)}{2} \qquad 7$$

$$X_{Body\ centre}\ = \frac{\left(Y_{Average\ hand}\ -\ Intercept_{Body}\right)}{Slope_{Body}} \qquad 8$$

$$Average\ hand - to - centre\ distance\ = \left|X_{Average}\ -\ X_{Centreline}\right| \qquad 9$$

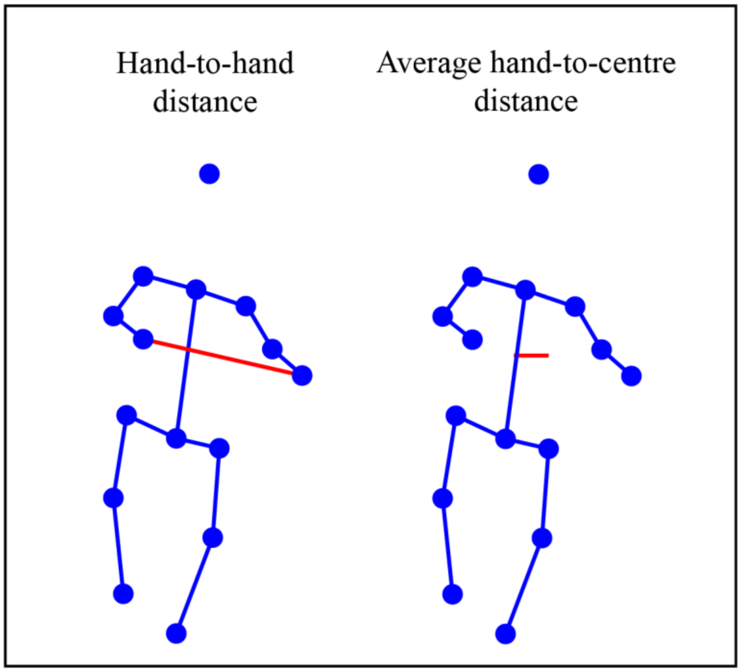
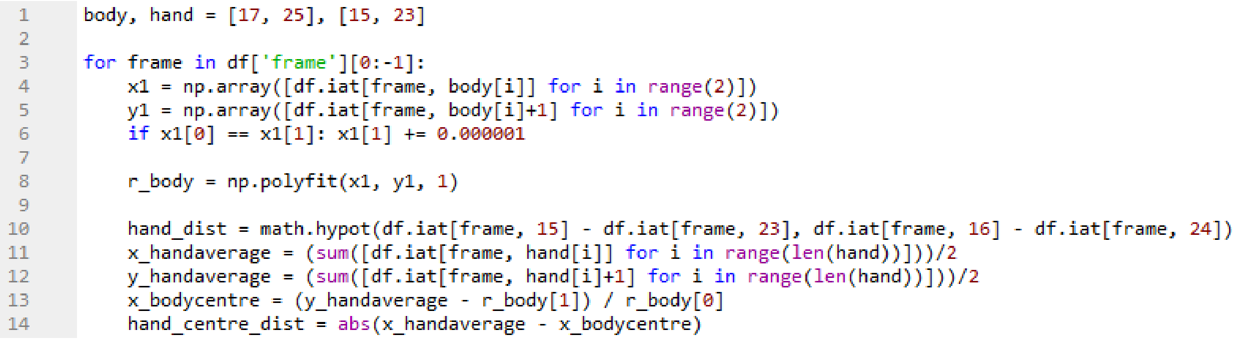


Figure 6: Illustration of the hand-to-hand contact variables. Hand-to-hand distance (left), with the distance measured between the wrist body keypoints highlighted in red. Average hand-to-centre distance (right), with the distance measured between the average hand position to the body centre tracking point highlighted in red.



```
1     body, hand = [17, 25], [15, 23]
2
3     for frame in df['frame'][0:-1]:
4         x1 = np.array([df.iat[frame, body[i]] for i in range(2)])
5         y1 = np.array([df.iat[frame, body[i]+1] for i in range(2)])
6         if x1[0] == x1[1]: x1[1] += 0.000001
7
8         r_body = np.polyfit(x1, y1, 1)
9
10        hand_dist = math.hypot(df.iat[frame, 15] - df.iat[frame, 23], df.iat[frame, 16] - df.iat[frame, 24])
11        x_handaverage = (sum([df.iat[frame, hand[i]] for i in range(len(hand))]))/2
12        y_handaverage = (sum([df.iat[frame, hand[i]+1] for i in range(len(hand))]))/2
13        x_bodycentre = (y_handaverage - r_body[1]) / r_body[0]
14        hand_centre_dist = abs(x_handaverage - x_bodycentre)
```

Figure 7. The segment of python code used to extract the hand-to-hand contact variables from the motion tracking data with other unrelated elements of the code removed.

Figure 7 code explanation

1: Creates separate lists containing the indexes of the x values of the body keypoints used to calculate the regression line and the average hand position.

3: Establishes a loop through the lines in the data frame, a particular segment, or the full motion tracking data. Each line of the data frame contains the motion tracking data from one frame of the recording. All variables for all features are calculated within the same loop in the full code.

4 & 5: Greatest NumPy arrays containing the coordinate values of the body keypoints used to calculate the regression line. Separate arrays for the x and the y values. Indexes of the x values are stated in the list from line one, the indexes of the y values are the index of the corresponding x index plus one.

6: Checks for and adjusts for edge cases where the x values of both body keypoints of the regression line are identical causing a problem with the calculation of the regression lines.
8: Calculates the slope and intercept of the regression line using the numpy.polyfit function. The variable r_body is the body regression line. The variable name + [0] contains the slope, and the variable name + [1] contains the intercept.
10: Calculates the hand-to-hand distance using the math.hypot function of the math library to execute equation 5. The value is stored in the variable hand_dist.
11. Calculates the average x position of the wrist body keypoints (equation 6). The value is stored in the variable x_handaverage.
12: Calculates the average y position of the wrist body keypoints (equation 7). The value is stored in the variable y_handaverage.
13: Calculates the x value of the new tracking point along the body regression line (equation 8) The value is stored as the variable x_bodycentre.
14: Calculates the absolute distance between the x_bodycentre and x_handaverage (equation 9). The value is stored in the variable hand_centre_dist.

2.4 – Variable- and feature threshold search

For the features to resemble the intended movement characteristic of the original MOS items, initial upper and lower bounds were established for each variable using a visual comparison of the original video and a digital rendering of the movement tracking data. A program was developed to create the digital rendering of the motion tracking data along with the current values for each variable. A side-by-side comparison of the original video and digital rendering at approximately the same frame can be seen in figure 8, and the full code for the program creating the digital rendering can be seen in Appendix C. The comparison was performed on a separate infant outside of the subject pool for this study. Subsequently, a grid search was performed for each feature to find the variable values best able to distinguish between the CP and non-CP groups. The grid search started with values evenly distributed between the selected upper and lower bounds for the variable. The area under the curve (AUC) of the receiver operating characteristic (ROC) curve was used to narrow the search towards the value or combination of values best able to distinguish between the groups. For features utilising the same variables, the variable thresholds are the same. The sensitivity and specificity for the full spectrum of possible thresholds were extracted with the ROC curve using the variable values providing the best AUC for the feature. The time or percentage threshold used to determine a positive and negative prediction for each feature was selected based on the best combined sensitivity and specificity, with both weighted equally. For the head centered time feature, a threshold was already established in the original definition of the item at 10 seconds. In this case, the sensitivity and specificity were analysed concurrently with the variable grid search to find values best suited for this threshold plus or minus two seconds.

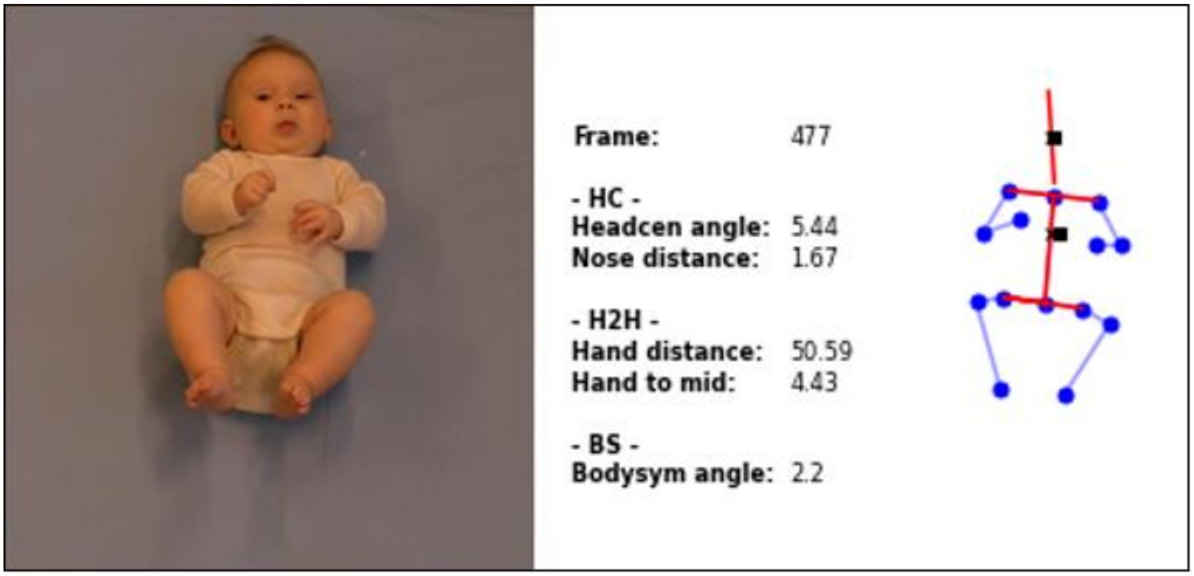


Figure 8: Illustration of the visual comparison of the initial threshold search. Original observational GMA video recording (left) and digital rendering of the video-based motion tracking data (right) of the same infant at approximately the same frame. The digital rendering includes the current frame number and the current value of each of the feature variables.

2.5 - Variable test

The value for each variable is compared to the threshold established in the grid search. If both values or single value in the case of body symmetry is within the specified threshold value, a “1” is appended to a list representing the feature. If either variables or the single variable in the case of body symmetry is not within the specified threshold value a “0” is appended to the same list (equation 10 and inverse of equation 10 for the large asymmetry variable). Any variable measuring distance between two body keypoints or other constructed tracking points in the motion tracking data (nose-to-centre distance, hand-to-hand distance, and average hand-to-centre distance) is transformed such that the variable states the percentage of the distance of the infant's upper body measured as the distance between the upper chest and mid pelvis body keypoints using equation 11. This was done to reduce the impact of video camera position and the size of the infant as confounding factors. The calculation of the variable values and the body distance variable is repeated for each line of the motion tracking data representing one frame of the video recording each.

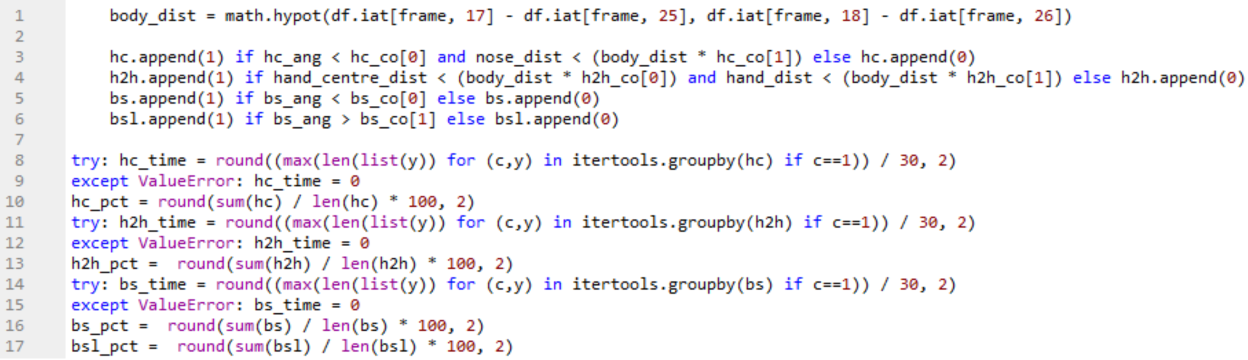
$$f(x) = \begin{cases} 0, & x < Threshold \\ 1, & x \geq Threshold \end{cases} \qquad 10$$

$$Body\ distance\ = \sqrt{\left(X_{Upper\ chest} - X_{Mid\ pelvis}\right)^2 + \left(Y_{Upper\ chest} - Y_{Mid\ pelvis}\right)^2} \qquad 11$$

2.6 - Feature value

Following the classification of the variables for all frames of the motion tracking data, the binary list is used to find both the percentage and length of the longest segment of the motion tracking data where the variable or variables in question are within the specified threshold. The percentage is calculated by summing all the items in the list, divided by the number of items in the list, and multiplied by 100 to convert it into percentage (equation 12). The longest time segment of the motion tracking data is calculated by using a function in the intertools library identifying the longest uninterrupted segment of "1" in the list and dividing by 30 frames per second to convert it to seconds (figure 9). The segment of python code executing the variable threshold test and feature value test is displayed in figure 9.

$$Percent\ feature\ value\ = \frac{\sum 1}{\sum 0 + \sum 1} * 100 \qquad 12$$



```
1      body_dist = math.hypot(df.iat[frame, 17] - df.iat[frame, 25], df.iat[frame, 18] - df.iat[frame, 26])
2  
3      hc.append(1) if hc_ang < hc_co[0] and nose_dist < (body_dist * hc_co[1]) else hc.append(0)
4      h2h.append(1) if hand_centre_dist < (body_dist * h2h_co[0]) and hand_dist < (body_dist * h2h_co[1]) else h2h.append(0)
5      bs.append(1) if bs_ang < bs_co[0] else bs.append(0)
6      bsl.append(1) if bs_ang > bs_co[1] else bsl.append(0)
7  
8  try: hc_time = round((max(len(list(y)) for (c,y) in itertools.groupby(hc) if c==1)) / 30, 2)
9  except ValueError: hc_time = 0
10 hc_pct = round(sum(hc) / len(hc) * 100, 2)
11 try: h2h_time = round((max(len(list(y)) for (c,y) in itertools.groupby(h2h) if c==1)) / 30, 2)
12 except ValueError: h2h_time = 0
13 h2h_pct =  round(sum(h2h) / len(h2h) * 100, 2)
14 try: bs_time = round((max(len(list(y)) for (c,y) in itertools.groupby(bs) if c==1)) / 30, 2)
15 except ValueError: bs_time = 0
16 bs_pct =  round(sum(bs) / len(bs) * 100, 2)
17 bsl_pct =  round(sum(bsl) / len(bsl) * 100, 2)
```

Figure 9. The segment of python code executing the variable threshold test and feature value test with other unrelated elements of the code removed.

Figure 9 code explanation

1: Calculates the distance between the upper chest- and mid-pelvis body keypoints using the math.hypot function of the math library to execute equation 11. The distance is stored in the variable as body_dist.

3 - 6: Tests each variable value against the variable threshold established in the grid search stored elsewhere in global lists. A "1" is added to a new list if both variables are within the threshold, and a "0" is added if any of the variables are not within the threshold (equation 10). These calculations are part of the loop that repeats for every line of the data frame.

8: Calculates the value of the head centered time feature. The segment tests the hc list (created in line 3) for the longest segment of uninterrupted "1" in the list. Divides the length of the segment by 30 to convert to seconds and stores the result rounded to two decimals in the variable hc_time.

9: Exception occurs in the previous calculation if no instance of 1 is found in the hc list, sets the variable hc_time to 0.

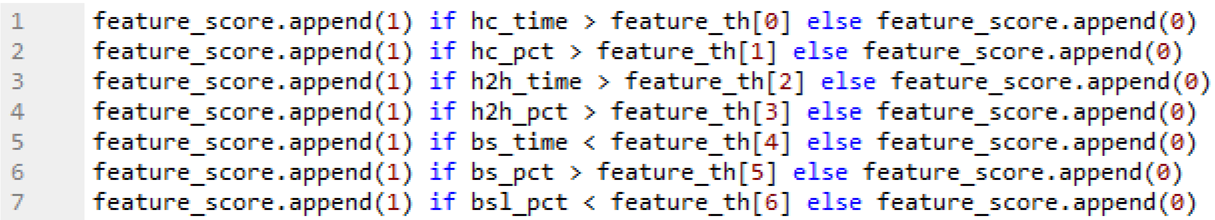
10: Calculates the head centered percent feature (equation 12). Divides the total number of items in the hc list by the total number of "1" in the hc list. Multiplied by 100 to convert to percentage, rounded to two decimals, and stored in the variable hc_pct.

11 - 17: Identical steps are performed to find the time- and percent features of the hand-to-hand contact item and the body symmetry item as was done on the head centered feature in lines 8 - 10. The values are stored in the variables h2h_time, h2h_pct, bs_time, bs_pct and bsl_pct respectively.

2.7 - Feature prediction score

Each feature is given a score of “1” or “0” depending on the feature value in comparison to the feature threshold. A “1” represents a negative CP prediction and a “0” represents a positive CP prediction. For features where a lower feature value is more indicative of a positive test, a “1” is allocated if the feature value is higher than the feature threshold (equation 13). For features where a higher value is more indicative of a positive test, a “1” is allocated if the feature value is lower than the feature threshold (inverse of equation 13). The segment of the python code executing the feature prediction test is displayed in figure 10.

$$f(x) = \begin{cases} 0, & x < Threshold \\ 1, & x \geq Threshold \end{cases} \qquad 13$$



```
1    feature_score.append(1) if hc_time > feature_th[0] else feature_score.append(0)
2    feature_score.append(1) if hc_pct > feature_th[1] else feature_score.append(0)
3    feature_score.append(1) if h2h_time > feature_th[2] else feature_score.append(0)
4    feature_score.append(1) if h2h_pct > feature_th[3] else feature_score.append(0)
5    feature_score.append(1) if bs_time < feature_th[4] else feature_score.append(0)
6    feature_score.append(1) if bs_pct > feature_th[5] else feature_score.append(0)
7    feature_score.append(1) if bsl_pct < feature_th[6] else feature_score.append(0)
```

Figure 10. The segment of the python code executing the feature prediction test with other unrelated elements of the code removed.

Figure 10 code explanation

1 - 7: Each line tests the previously calculated feature value against the corresponding feature threshold (equation 13). A “1” is appended to the feature_score list if a negative test result is observed and a “0” is appended to the same list if a positive test result is observed.

2.8 Program development and statistical analysis

All programs developed for this study were written in python programming language using spyder integrated scientific development environment. The main program for motion tracking data testing utilises the libraries pandas, NumPy, glob, math, and intertools. Pandas’ library is used for importing CSV files, exporting the main results as CSV and data frame management in the program. NumPy library is used for calculating the regression lines as well as constructing the needed arrays for the calculation. Glob library is used to iterate through all CSV files in specific folders. Math library is used for the inverse tangent-, radians to degrees- and hypotenuse calculation. Lastly, the intertools library is used to identify the length of the longest segment of uninterrupted identical items a list.

Results were analysed using IBM SPSS Statistics for Windows, version 27 (IBM Corp., Armonk, N.Y., USA). Test of normality was performed with the Shapiro-Wilk test and the statistical significance of the difference in distribution was tested with the Mann-Whitney U test. The difference in the feature score between the outcome groups was tested with the independent samples T-test. ROC curve analysis was used to find the variable threshold values for each feature, and the ROC curve coordinates were exported to Microsoft Excel to calculate the combined sensitivity and specificity for each possible threshold to find the largest combined total. Additionally, the ROC curve coordinates were used to provide the sensitivity and specificity of the individual feature scores and sum of features scores, which in conjunction with the number of positive and negative cases in the data set enables the calculation of accuracy. P-values less than 0.05 were considered statistically significant.

3.0 - Results

3.1 - Results of the threshold search

The threshold for each feature variable is displayed in table 6. The threshold value for each feature determining positive or negative prediction for the feature is displayed in table 7. Results for the training data set are displayed in appendix A.

Table 6: The variable threshold values developed in the grid search on the training data set.

Variable	Threshold
Head centered angle	12 degrees
Nose-to-centre distance	5 percent of body distance
Hand-to-hand distance	25 percent of body distance
Average hand-to-centre distance	6 percent of body distance
Body symmetry angle	8 degrees
Body symmetry large asymmetry angle	20 degrees

Table 7: The feature threshold values developed in the grid search on the training data set.

Feature	Threshold
Head centered time	> 11.5 seconds
Head centered percent	> 17.5 percent
Hand-to-hand time	> 1 second
Hand-to-hand percent	> 0.5 percent
Body symmetry time	< 7.5 seconds
Body symmetry percent	> 20 percent
Body symmetry large asymmetry	< 23.5 percent

3.2 - Main results

Shapiro Wilks test of normality indicates that no feature value is normally distributed (p = 0.000 - 0.009).

Head centered features

The AUC and distribution of the head centered features are displayed in figure 11. Head centered time in panel A and head centered percent in panel B. The AUC for the head centered time- and percent features was .611 (95% CI = .488 - 734), and .606 (95% CI = .478 - 733) respectively. The box plot indicates some difference in distribution between the CP and non-CP outcome groups for both features, although the difference was not statistically significant Mann-Whitney U p = .106 for the head centered time feature and p = .123 for the head centered percent feature.

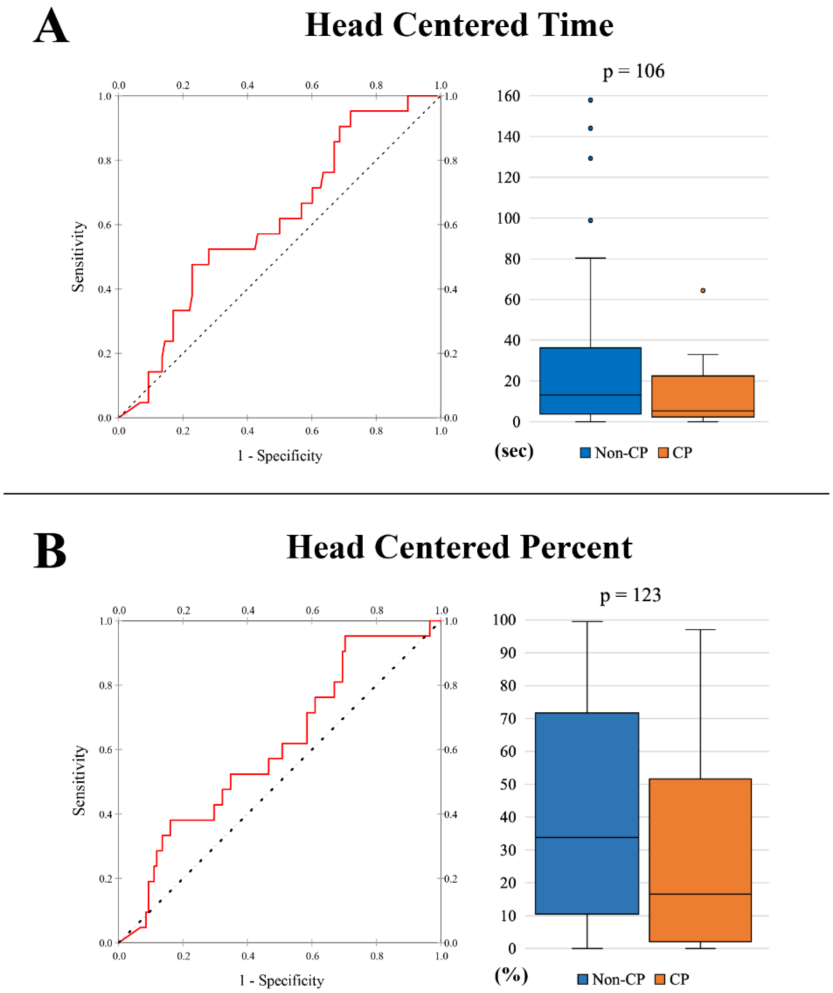


Figure 11: Displays the ROC curve and difference in distribution between the CP and non-CP outcome groups with the box plot for the head centered features. Head centered time (A) and head centered percent (B).

Hand-to-hand features

The AUC and distribution of the hand-to-hand contact features are displayed in figure 12. Hand-to-hand time in panel A and hand-to-hand percent in panel B. The AUC for the hand-to-hand time- and percent feature was .548 (95% CI = .425 - .672) and .551 (95% CI = .427 - .674) respectively. The box plot indicates a difference in distribution for both features, the difference was not statistically significant, Mann-Whitney U p = .402 for the time feature and p = .379 for the percent feature. Both hand-to-hand contact features are impacted by a low prevalence for the CP and non-CP outcome group, although the prevalence is slightly lower in the group that later developed CP. 77 (65.3%) of the non-CP outcome group and 15 (71.4%) of the CP outcome group measured no hand-to-hand contact in the motion tracking data.

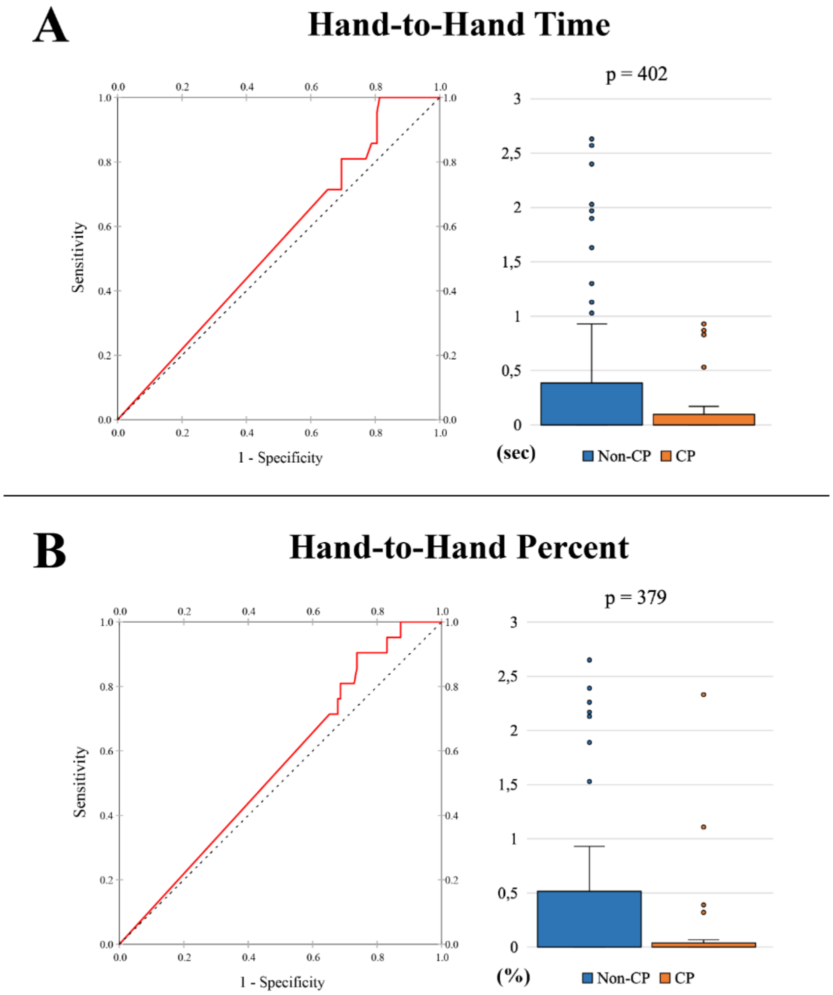


Figure 12: Displays the ROC curve and difference in distribution between the CP and non-CP outcome groups with the box plot for the hand-to-hand contact features. Hand-to-hand time feature (A) and hand-to-hand percent (B).

Body symmetry feature

The AUC and distribution of the body symmetry features are displayed in figure 13. Body symmetry time in panel A, body symmetry percent in panel B, and body symmetry large asymmetry in panel C. The AUC for the body symmetry time, body symmetry percent and body symmetry large asymmetry percent features was .474 (95% CI = .333 - .615), .534 (95% CI = .404 - .664), and .552 (95% CI = .417 - .688) respectively. The box plot indicates limited differences between the CP and non-CP outcome groups, with the largest difference displayed by the body symmetry percent feature. No difference in distribution was statistically significant Mann-Whitney U p = .704 for body symmetry time feature, p = .621 for body symmetry percent feature, and p = .446 for body symmetry large asymmetry feature.

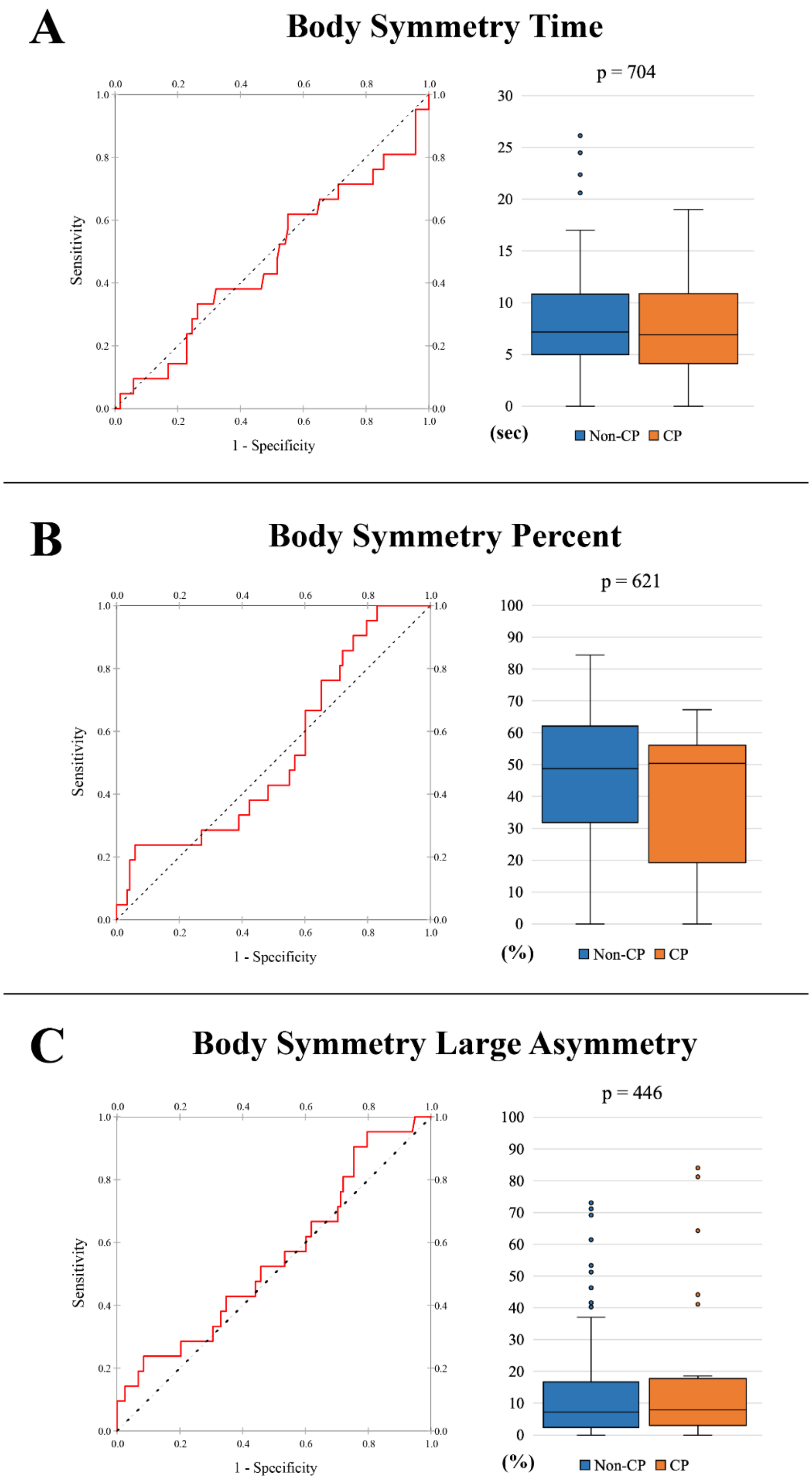


Figure 13: Displays the ROC curve and difference in distribution between the CP and non-CP outcome groups with the box plot for the body symmetry features. Body symmetry time feature (A), body symmetry percent (B), and body symmetry large asymmetry percent (C).

Feature prediction score

The results following conversion to binary CP prediction using the feature thresholds can be seen in table 8. Six features had positive predictive ability, the body symmetry time did not. The mean difference between the CP and non-CP outcome groups ranged from -.085 to .186. The mean difference for hand-to-hand time and the combined total was statistically significant, the hand-to-hand time mean difference was .186 (p = 0,031) and the mean difference for the combined total was .750 (p = 0.031). The mean difference for the remaining features difference was not statistically significant measured with the independent sample T-test. Both hand-to-hand features had high sensitivity but low specificity and both body symmetry percent and body symmetry large asymmetry had low sensitivity but high specificity. The sensitivity, specificity, and accuracy for each possible threshold for the combined total score of the test data set can be seen in table 9.

Table 8: The sensitivity, specificity, and accuracy for each feature on the left, and results of the independent sample T-test for each feature, and the combined total on the right

	Predictive power			**Independent sample T-test**			
Feature	**Sens. [1]**	**Spec. [2]**	**Acc. [3]**	**Mean non-CP N = 118**	**Mean CP N = 21**	**Mean diff**	**Sig.[4]**
HC time	57.1	54.2	54.7	.542	.429	.114	.351
HC pct	47.6	66.9	64.0	.669	.524	.146	.235
H2H time	100	18.6	30.9	.186	.000	.186	.031
H2H pct	90.5	26.3	36.0	.263	.095	.168	.098
BS time	38.1	53.4	51.1	.534	.619	-.085	.474
BS pct	23.8	89.8	79.9	.898	.762	.136	.080
BSL pct	23.8	84.7	75.5	.847	.762	.086	.404
Total	---	---	---	3.941	3.190	.750	.031

[1] Sensitivity (%). [2] Specificity (%). [3] Accuracy (%). [4] Two-sided p

Table 9: The sensitivity, specificity, and accuracy of the combined total score for all thresholds.

Threshold	**Sensitivity**	**Specificity**	**Accuracy**
0.5	0	99.2	84.2
1.5	14.3	95.0	82.8
2.5	28.6	81.4	73.4
3.5	57.4	61.0	60.4
4.5	85.7	34.7	42.4
5.5	95.2	16.1	28.0
6.5	100	6.8	20.8

4.0 – Discussion

The aim of this study was to develop an automatic assessment of the hand-to-hand contact, head centered, and body symmetry MOS items by developing movement features using quantifiable variables extracted from video-based motion tracking data. Additionally, analyse how accurate individual and sum of movement features differentiate and predict CP and non-CP outcomes in high-risk infants.

The results suggest that quantifying this category of MOS items with the intent of differentiating between CP outcome groups is feasible. The head centered and hand-to-hand contact features can to a reasonable extent differentiate between the CP and non-CP outcome groups, but any individual feature's ability to predict the development of CP in singular cases is limited. However, the predictive capability of multiple combined features surpasses that of any individual feature the combination is composed of. This reflects one of the findings of Einspieler 2019. They found that the absence of an individual movement pattern by itself could not predict the degree of functional limitations. The combination of several atypical movements and postural patterns often in conjunction with abnormal or cramped-synchronised movement character and the lack of FMs was predictive of GMFCS outcomes (15).

No previous studies using exclusively the score of the observed movement and or postural patterns of the MOS to predict the development of CP were found. However, a study investigating the relationship between the MOS at 3 to 5 months and neurological outcome found that a non-optimal MOS predicted CP with 100% accuracy, with all infants in the study who developed CP had a MOS between 5 and 7. Although this is promising, they also found a large variety in the quality of movement and posture among those with normal development. With only 35% of normally developing infants displaying optimal motor performance (37). A different study analysing the GMA and MOS among extremely preterm infants observed almost identical distribution within the FM category and the quality of other movements category. The posture category did not display the same similarity (38). Although this does indicate a correlation between the FM and movements categories, the study did not follow up with CP outcome.

4.1 - Head centered item

Among the group who later developed CP in this study, 57.1 % displayed atypical head centered time i.e., no segment of 11.5 seconds or more with head centered detected in the motion tracking data. This is almost identical to Einspieler 2019, which found that 60.1% of infants in their study (exclusively infants with later CP diagnosis) displayed atypical head centered posture (15). The similarity suggests that the head centered time feature developed in this study bears a satisfactory resemblance to the original head centered MOS item and its threshold for normal or atypical assessment. The head centered percent feature did not exhibit the same level of similarity, with 47.6% of the infants developing CP scoring atypical on the feature.

The method for extracting the nose-to-centre distance in this study is likely subject to inaccuracies in particular circumstances. The variable is measured exclusively in the x-axis of the motion tracking data causing the distance measured during frames where the head regression line is not exactly perpendicular to the x-axis to be inaccurate. This inaccuracy would increase with a larger deviation from the y-axis, and the distance measured would increase or decrease depending on the direction of a simultaneous rotation or tilt of the infant's head. Considering the head centered angle regularly surpasses the 12-degree threshold, the inaccuracy caused by this issue could be significant depending on the simultaneous position of the body regression line. Rather than calculating the distance exclusively in the x-axis, the line measured could be perpendicular from the head regression line towards the nose body keypoint. Altering the approach for variable extraction for this variable could make the feature more reliable in asymmetrical positions. The same inaccuracy is present for the average hand-to-centre distance variable for the hand-to-hand contact features, but to a lesser extent due to the body regression line being more in line with the x-axis in most cases. The method for calculating the head centered angle seems to be considerably more robust. Although it might be worth investigating if the variable is better suited to differentiate the outcome groups by including the upper chest body keypoint in the regression line calculation.

4.2 - Body symmetry item

The body symmetry features developed in this study did not resemble the distribution of normal and atypical body symmetry in Einspieler 2019. In this study, the body symmetry time, body symmetry percentage, and body symmetry large asymmetry features found 38.1%, 23.8%, and 23.8% respectively to be atypical among the positive CP group, compared to 65% in Einspieler 2019 (15). The body symmetry time feature was the only feature developed in this study with no predictive capability, with the CP group scoring higher than the non-CP group on the test data set. Although the body symmetry percentage and large asymmetry features were able to differentiate between the groups, the difference was small and not significant. Lastly, the body symmetry large asymmetry percentage feature provided the least mean difference among the features with a positive predictive capability.

Although the movement characteristics for the body symmetry item is concise and the body symmetry angle is a direct quantification of the MOS item, this suggests that the features developed should be re-evaluated. A possibility of a more accurate way of recreating this MOS item could be a feature calculating the average body symmetry angle throughout the motion tracking data using the body symmetry angle developed.

4.3 - Hand-to-hand contact item

In this study the prevalence of any hand-to-hand contact during the extent of the motion tracking data was low, 34,7% for the non-CP group, 28.6% for the CP group, and 33,8% combined. Although the prevalence is low, it is significantly higher than the 10% hand-to-hand contact observed in Einspieler et al 2019, which was regardless of normal or atypical classification of the observed feature. This might suggest that the variable thresholds developed in this study determining what is considered hand-to-hand contact were lower than for the GMA observers in Einspieler 2019 (15). Alternatively, the “repetitively” part of “both hands repetitively touch, stroke or grasp each other” in the MOS item definition could also be the cause for this discrepancy. In this present study hand-to-hand contact is registered regardless of length or repetitive nature as soon as both variable threshold values are simultaneously met. Another implication that might be derived from these findings, is that the similarity of occurrence in both outcome groups indicates that the eventual CP outcome has negligible influence on the prevalence of hand-to-hand contact. This would contradict the notion put forth in Einspieler 2019 that the low prevalence of hand-to-hand contact was due to the lack of visuomotor coordination caused by a sample consisting of exclusively positive CP outcomes (15).

The hand-to-hand contact item definition consists of two separate elements. First, whether the hands are brought together over the midline of the infant, and second, how the fingers touch once the hands are over the midline (15). In the original MOS, the classification of the item as normal or atypical is centred around the way the fingers are positioned during contact (15). The available detail in the motion tracking data makes it impossible to extract the position and movement of the fingers of the infants. A variation of the item was chosen in this study despite the inability to extract finger movement and because of a theory that arm movements against the line of gravity might be a relevant predictive factor.

4.4 - Variable and feature threshold

This study used CP outcome to develop thresholds for the variables and features. The exception was head centered time feature which had a pre-established threshold in the MOS item definition (15), and the initial upper and lower limitation for the variable threshold to maintain the intended movement characteristics of the MOS item. This removes most of the human subjectivity from the development of the variables and features. Developing the features based on the outcome without input from GMA experts has the advantage of eliminating the subjective thresholding of GMA experts or preferences of a GMA team (39). It might in turn remove the clinical relevance of the features established through previous observational research on GMA and MOS (10,15–18).

Like the variable and feature threshold selection, the combined sensitivity and specificity with both weighted equally is used as the main measure for predictive capability. Using accuracy as a measure of predictive capability or as a basis for selecting variable- or feature thresholds with the current data sets would bias towards high specificity at the expense of sensitivity. This is due to the training and testing motion tracking data sets consisting of 15.1% positive CP outcomes, which is to be expected in a representative group of high-risk infants with various high-risk factors (2). The approach used in this study to use sensitivity and specificity to determine thresholds does result in lower accuracy for each of the features, however, it eliminates this bias.

4.5 - Study limitations and future aspects

The scope of this current study is limited, quantifying 3 out of 36 total MOS items within the observed movement- and postural patterns categories, with the developed features displaying varying levels of significance differentiating the CP and non-CP outcome group distributions. Additionally, this study exclusively investigated one item subtype within the two categories, focusing on items that can be identified and extracted on any singular frame of the motion tracking data. Considering these limitations, it is unknown how the result of this current study translates to other items, particularly the compound movement items, and finer movement items containing the eyes, mouth, fingers, etc. Hence, future studies quantifying MOS items should include a wider selection of items and include items of the types not investigated in this study. The 19 body keypoints available in the video-based motion tracking data used in this study do not facilitate the extraction of finer movements of the mouth, eyes, and fingers. With further development of these systems, more MOS items could be more realistic targets for quantification in future studies on the subject. Regarding the more complex MOS items with compound movement spanning several frames of the motion tracking data, the currently available detail should be sufficient to facilitate extraction, but the complexity of these movements might require machine learning with GMA expert annotation to accomplish. Expert annotation would reintroduce the potential issue of inter-observer reliability, with differing subconscious thresholds between the observers doing the annotations. Although the inter-observer reliability is high when differentiating between normal FMs and non-normal FMs with experienced observers (20). A study investigating the inter-observer agreement within the four other categories of the MOS found a lower agreement for the observed movement and postural patterns than for FMs. Additionally, the agreement was based on the score of the category, which is decided based on the number of normal versus atypical items. Making the inter-observer agreement on individual items potentially lower due to potential chance agreements on the category (39).

The current study directly interpreted the score of individual features and the sum of the scores of the features developed. This rudimentary method is likely a suboptimal approach for the analysis of the combination of features. More sophisticated methods could produce more significant differences between the outcome groups. Methods such as partial least square regression, taking the covariation between features into account (40,41), have been used in some previous quantitative CP prediction studies (27,30). Future studies quantifying individual movement features of the MOS should investigate the possibility of clustering features to improve the capability.

Although most of the features developed in this current study did not achieve statistical significance, the results are promising and do justify further research and development. Improving or replacing features developed for the MOS items in this current study and adding features for additional items of the movement and postural patterns categories. As well as implementing better analytical methods for differentiating CP and non-CP outcome groups and predicting outcomes in individual cases using the combination of features. With continuing development of an automatic video-based assessment of individual movements, particularly three applications of the assessment should be investigated.

First, whether a combined assessment of features score can differentiate between different GMFCS outcomes in infants with CP outcomes. Many studies have observed significant correlations between the movements observed during the FM period and eventual GMFCS classifications in positive CP outcome cases (15,25,30). Ihlen et al 2020 assessed the proportion of risk-related movements with video-based motion tracking using multivariate empirical mode decomposition. They found that the proportion of risk-related movements in cases with a GMFCS outcome of IV or V was significantly higher than that of cases with a GMFCS of I-III (30). Einspieler 2019 found a strong association between the MOS and GMFCS classification. In their study, no infant with a positive CP diagnosis achieved an optimal MOS (25 - 28), but 3% had a high score of 20 - 24. Among these, nine were classified with a GMFCS I, and the remaining three were classified with GMFCS II. Furthermore, a MOS over 14 was associated with lower functional disability with a GMFCS outcome of I or II, while infants with eventual GMFCS IV or V typically scored less than eight. Additionally, they found that excluding the FM category had a negligible impact on the predictive capacity regarding GMFCS outcomes. Excluding the FM category from the MOS increases the combined impact of observed movement- and postural patterns in the analysis from 29% to 50%. This suggests that the two categories should have some predictive capability regarding GMFCS outcomes in positive CP cases (15).

Second, investigating the possibility of quantifiably recreating previous observations regarding specific movements and the development of certain CP subtypes. Previous studies have found associations between a reduction in certain segmental movements on one side and the development of unilateral CP (42–44). The association was also observed in Einspieler et al 2019 with 63% of the infants who later developed unilateral CP showing a higher frequency of movements in the unaffected arm, with the remainder similar quantity of segmental movements. They also found that 58% of the infants later diagnosed with dyskinesia displayed circular arm movements, and that infants who were diagnosed with dyskinesia had more atypical postural patterns than those diagnosed with spasticity (15).

Third, there are numerous studies on the development of an automatic assessment of FM, many of them also based on video recording during the prominent FM period (25–30). There could be possibilities for combining a fully developed version of the automatic video-based assessment of movement and postural patterns with an automatic video-based assessment of FMs. Essentially creating a fully quantified video-based version of the MOS, performing all parts of the assessment based on a single video recording. Any such combination would benefit from both the assessment of FMs and the assessment of individual movement features utilising the same method for interpreting. Making the CIMA model in Ihlen 2020 the most applicable of the previously mentioned studies as it utilises similar video-based motion tracking (30).

Conclusion

The aim was to develop and test an automatic assessment of the hand-to-hand contact, head centered, and body symmetry items of the MOS by developing movement features using quantifiable variables extracted from video-based motion tracking data on high-risk infants. Additionally, analyse how accurate individual and sum of movement features differentiate and predict CP and non-CP outcomes in high-risk infants. The results show that individual movement and postural features of the category examined can be quantified and some features can differentiate CP and non-CP outcome groups. Individual features had limited predictive capability in individual cases, however, combining features elevated the predictive capability. The head centered features closely mirrored that of previous findings, the remaining features might warrant re-evaluation. The scope of the study was limited and should be replicated with a larger set of features including more varieties in item type. Future studies should also investigate if the combination of quantified features can predict GMCFS outcome and if specific quantified movement features can predict CP subtype.

Appendix A. Result for the training data set

Shapiro Wilks test of normality indicates that no feature value is normally distributed (p = 0.000 - 0.021).

Head centered

The AUC and distribution of the head centered features on the training data set are displayed in figure A1. Head centered time in panel A and head centered percent in panel B. The AUC for the head centered time- and percent features was .653 (95% CI .578 - 728), and .637 (95% CI 559 - 716) respectively. The box plot indicates a large difference in distribution between the CP and non-CP outcome groups for both features Mann-Whitney U p = .000 for the head centered time feature and p = .001 for the head centered percent feature. The confidence interval and Mann-Whitney U test indicate statistically significant difference in between the CP and non-CP groups for both features.

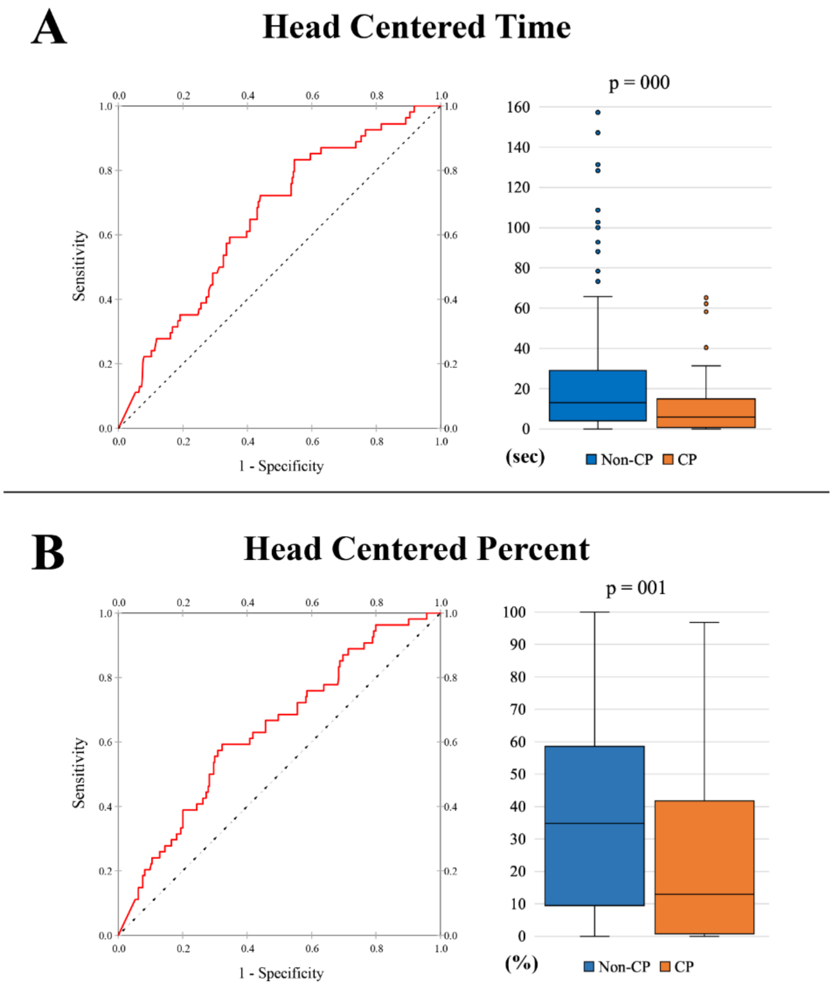


Figure A1: Displays the ROC curve and difference between the CP and non-CP outcome groups with the box plot for the head centered features for the training data set. Head centered time feature (A) and head centered percent (B).

Hand-to-hand contact

The AUC and distribution of the hand-to-hand contact features are displayed in figure A2. Hand-to-hand time in panel A and hand-to-hand percent in panel B. The AUC for the hand-to-hand time- and percent feature was .561 (95% CI .485 - .637) and .560 (95% CI .484 - .636) respectively. The box plot indicates a difference in distribution for both features Mann-Whitney U p = .087 for the time feature and p = .091 for the percent feature. The confidence interval and Mann-Whitney U test indicate the difference between the CP and non-CP groups was not statistically significant for any of the features.

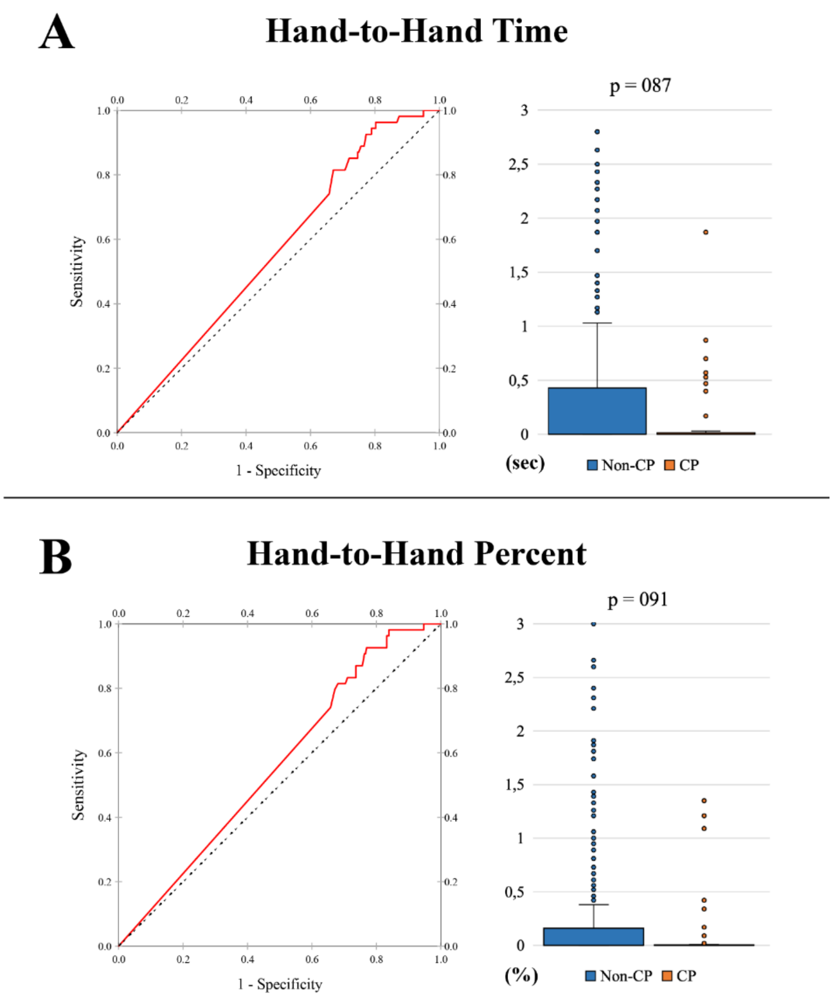


Figure 12: Displays the ROC curve and difference between the CP and non-CP outcome groups with the box plot for the hand-to-hand contact features for the training data set. Hand-to-hand time feature (A) and hand-to-hand percent (B).

Body symmetry

The AUC and distribution of the body symmetry features are displayed in figure A3. Body symmetry time in panel A, body symmetry percent in panel B, and body symmetry large asymmetry in panel C. The AUC for the body symmetry time, body symmetry percent and body symmetry large asymmetry percent features was .563 (95% CI .475 - .651), .522 (95% CI .436 - .609), and .508 (95% CI .419 - .598) respectively. The box plot indicates limited differences between the CP and non-CP outcome groups. No difference in distribution was statistically significant Mann-Whitney U p = .142 for body symmetry time, p= .600 for body symmetry percent, and p = .845 for body symmetry large asymmetry. The confidence interval and Mann-Whitney U test indicate the difference between the CP and non-CP groups was not statistically significant for any of the features.

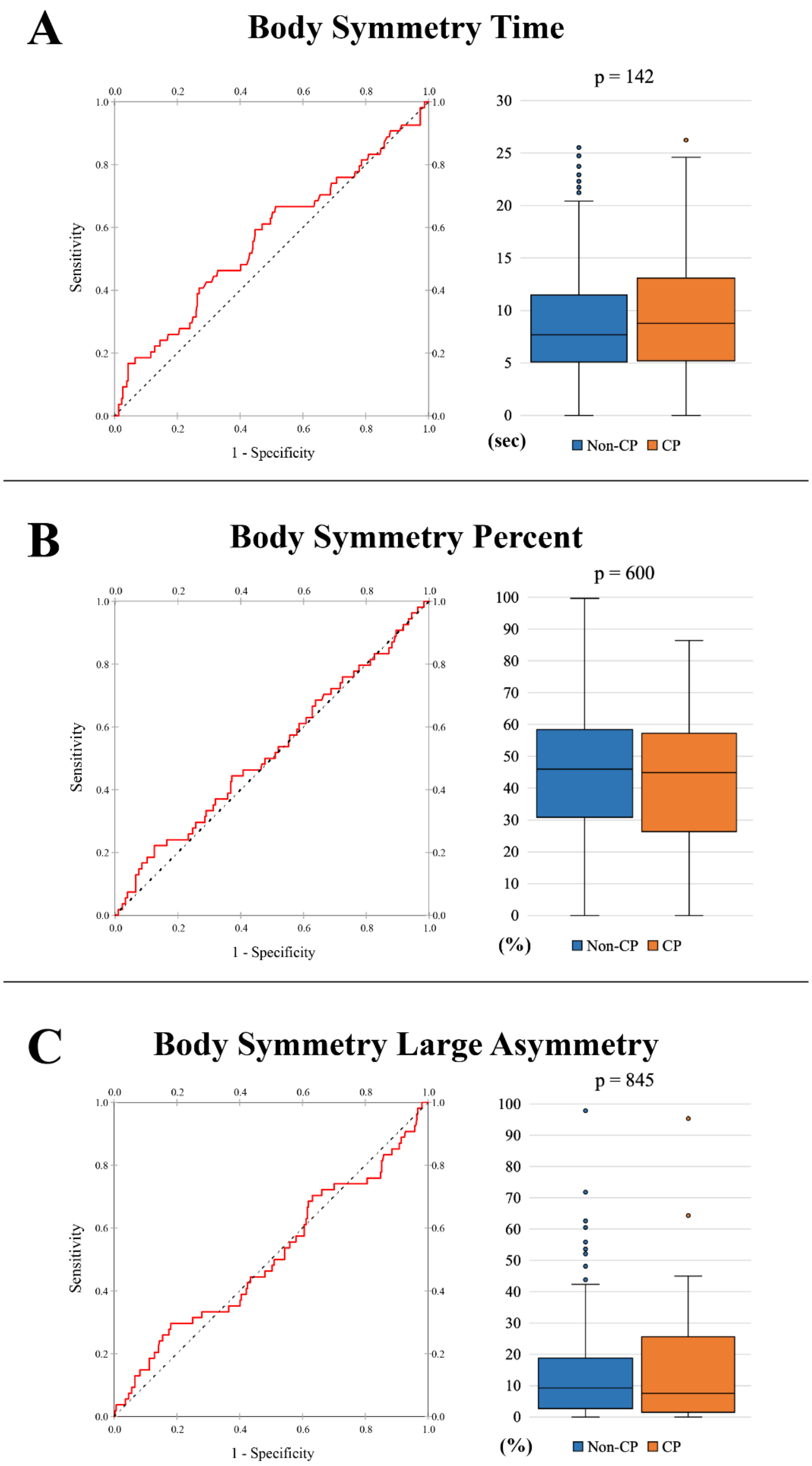


Figure A3: Displays the ROC curve and difference between the CP and non-CP outcome groups with the box plot for the body symmetry features for the training data set. Body symmetry time feature (A), body symmetry percent (B), and body symmetry large asymmetry percent (C).

Feature test

The results following conversion to binary CP prediction using the feature threshold values can be seen in table A1. Most features had a statistically significant difference in mean score between the CP and non-CP outcome groups except body symmetry percent and body symmetry large asymmetry tested with independent sample t-test. The sensitivity, specificity, and accuracy for each of possible threshold for the combined total score of the test data set can be seen in table A2.

Table A1: The sensitivity, specificity and accuracy for each feature on the left, and results of the independent sample T-test for each feature and the combined total on the right

	Predictive power			Independent sample T-test			
Feature	Sens. [1]	Spec. [2]	Acc. [3]	Mean non-CP N = 304	Mean CP N = 54	Mean diff	Sig. [4]
HC time	70.4	56.6	58.7	.566	.296	.270	.001
HC pct	59.3	66.5	65.4	.664	.407	.257	.001
H2H time	96.3	18.4	30.2	.184	.037	.147	.007
H2H pct	92.6	22.7	33.2	.227	.074	.153	.010
BS time	66.7	48.4	51.1	.484	.333	.150	.041
BS pct	18.5	87.5	77.1	.875	.815	.060	.233
BSL pct	29.6	81.9	74.0	.819	.707	.115	.050
Total	---	---	---	3.819	2.667	1.152	.001

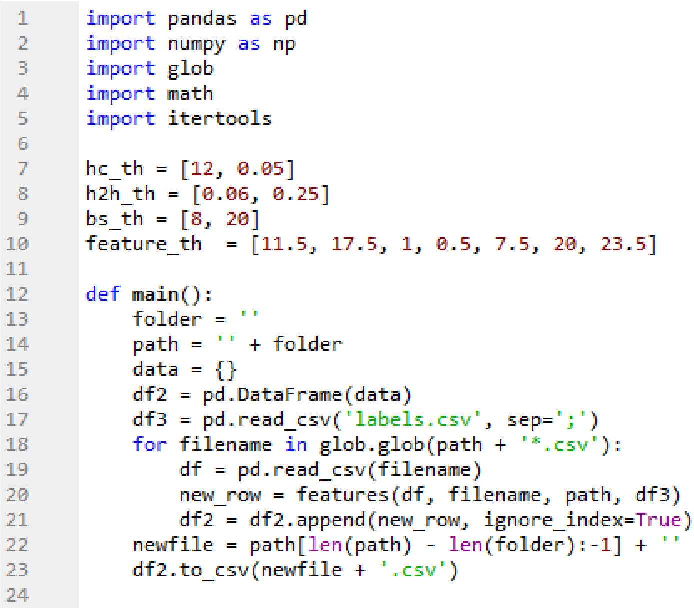
[1] Sensitivity. [2] Specificity. [3] Accuracy. [4] Two-sided p

Table A2: The sensitivity, specificity, and accuracy of the combined total score.

Threshold	Sensitivity	Specificity	Accuracy
0.5	5.6	100	85.8
1.5	13.9	97.0	84.4
2.5	48.1	79.6	74.9
3.5	74.1	59.5	61.7
4.5	94.4	29.6	39.4
5.5	98.1	11.8	24.9
6.5	100	4.3	18.7

Appendix B. Full code for the main test program

The full version of the main python code used for variable calculation, feature value calculation and feature score calculation is displayed in figure B1, B2 and B3. A short explanation of the code below each figure.



```
1   import pandas as pd
2   import numpy as np
3   import glob
4   import math
5   import itertools
6
7   hc_th = [12, 0.05]
8   h2h_th = [0.06, 0.25]
9   bs_th = [8, 20]
10  feature_th  = [11.5, 17.5, 1, 0.5, 7.5, 20, 23.5]
11
12  def main():
13      folder = ''
14      path = '' + folder
15      data = {}
16      df2 = pd.DataFrame(data)
17      df3 = pd.read_csv('labels.csv', sep=';')
18      for filename in glob.glob(path + '*.csv'):
19          df = pd.read_csv(filename)
20          new_row = features(df, filename, path, df3)
21          df2 = df2.append(new_row, ignore_index=True)
22      newfile = path[len(path) - len(folder):-1] + ''
23      df2.to_csv(newfile + '.csv')
24
```

Figure B1: The first segment of the main code developed to calculate and test the features.

Code explanation for figure B1

1 - 5: Imports the necessary python libraries.

7 - 10: Lists containing the threshold values for all variables and features.

13: Dictates which folder the program is to be executed on e.g., training, test or validation.

14: States the path to the folder.

15: Creates an empty dictionary.

16: Creates a new data frame including the empty dictionary from 15.

17: Imports the labels csv file containing the CP or non-CP outcome of each subject in all data sets as a new data frame.

18: Creates a loop through all csv files in the folder designated by the folder and path variables. Allowing all csv files to be tested one by one.

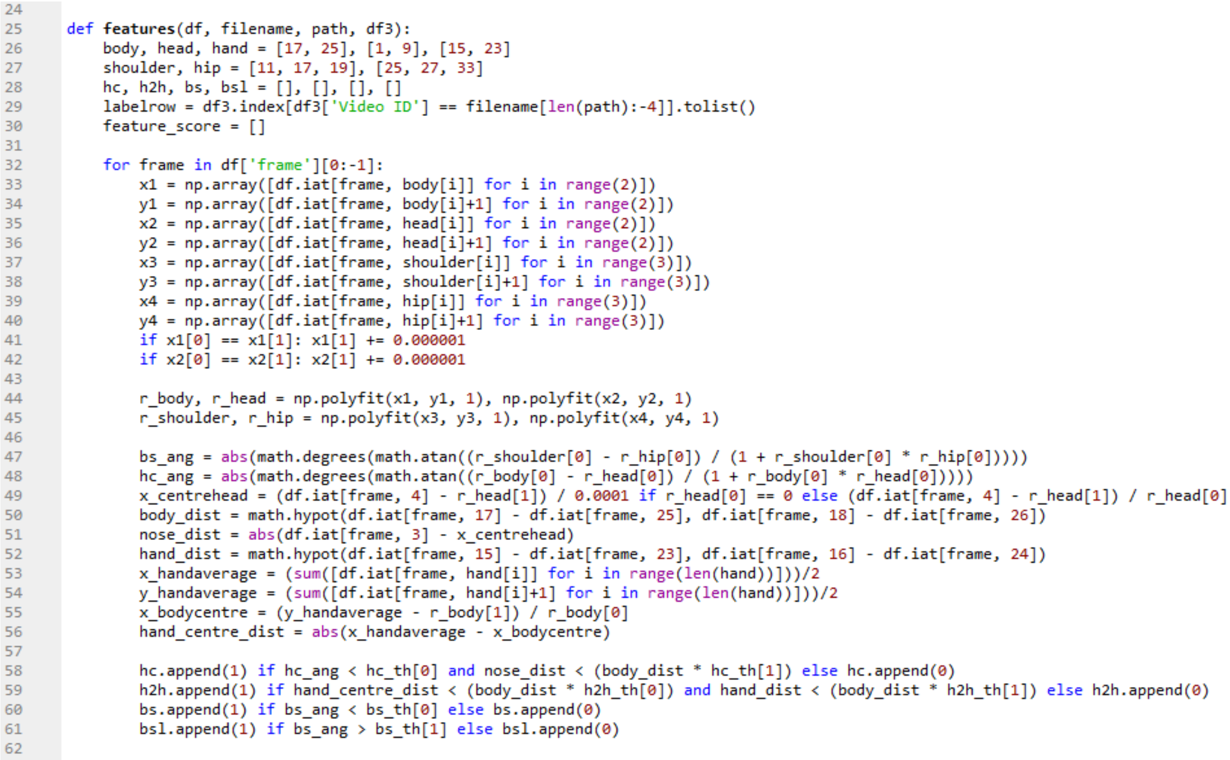
19: Opens the current csv file in the loop from the previous line. Information is stored in the variable df.

20: Sends the data frame of the current csv file to be tested in the function starting at line 25. All feature scores and feature values are returned as the dictionary variable new_row.

21: New_row is added to the data frame created in line 15. The program then loops back to line 18 and repeats with the next csv file in the folder.

22: Dictates the name output file.

23: The data frame from line 15 is exported as a csv.



```
24
25  def features(df, filename, path, df3):
26      body, head, hand = [17, 25], [1, 9], [15, 23]
27      shoulder, hip = [11, 17, 19], [25, 27, 33]
28      hc, h2h, bs, bsl = [], [], [], []
29      labelrow = df3.index[df3['Video ID'] == filename[len(path):-4]].tolist()
30      feature_score = []
31
32      for frame in df['frame'][0:-1]:
33          x1 = np.array([df.iat[frame, body[i]] for i in range(2)])
34          y1 = np.array([df.iat[frame, body[i]+1] for i in range(2)])
35          x2 = np.array([df.iat[frame, head[i]] for i in range(2)])
36          y2 = np.array([df.iat[frame, head[i]+1] for i in range(2)])
37          x3 = np.array([df.iat[frame, shoulder[i]] for i in range(3)])
38          y3 = np.array([df.iat[frame, shoulder[i]+1] for i in range(3)])
39          x4 = np.array([df.iat[frame, hip[i]] for i in range(3)])
40          y4 = np.array([df.iat[frame, hip[i]+1] for i in range(3)])
41          if x1[0] == x1[1]: x1[1] += 0.000001
42          if x2[0] == x2[1]: x2[1] += 0.000001
43
44          r_body, r_head = np.polyfit(x1, y1, 1), np.polyfit(x2, y2, 1)
45          r_shoulder, r_hip = np.polyfit(x3, y3, 1), np.polyfit(x4, y4, 1)
46
47          bs_ang = abs(math.degrees(math.atan((r_shoulder[0] - r_hip[0]) / (1 + r_shoulder[0] * r_hip[0]))))
48          hc_ang = abs(math.degrees(math.atan((r_body[0] - r_head[0]) / (1 + r_body[0] * r_head[0]))))
49          x_centrehead = (df.iat[frame, 4] - r_head[1]) / 0.0001 if r_head[0] == 0 else (df.iat[frame, 4] - r_head[1]) / r_head[0]
50          body_dist = math.hypot(df.iat[frame, 17] - df.iat[frame, 25], df.iat[frame, 18] - df.iat[frame, 26])
51          nose_dist = abs(df.iat[frame, 3] - x_centrehead)
52          hand_dist = math.hypot(df.iat[frame, 15] - df.iat[frame, 23], df.iat[frame, 16] - df.iat[frame, 24])
53          x_handaverage = (sum([df.iat[frame, hand[i]] for i in range(len(hand))]))/2
54          y_handaverage = (sum([df.iat[frame, hand[i]+1] for i in range(len(hand))]))/2
55          x_bodycentre = (y_handaverage - r_body[1]) / r_body[0]
56          hand_centre_dist = abs(x_handaverage - x_bodycentre)
57
58          hc.append(1) if hc_ang < hc_th[0] and nose_dist < (body_dist * hc_th[1]) else hc.append(0)
59          h2h.append(1) if hand_centre_dist < (body_dist * h2h_th[0]) and hand_dist < (body_dist * h2h_th[1]) else h2h.append(0)
60          bs.append(1) if bs_ang < bs_th[0] else bs.append(0)
61          bsl.append(1) if bs_ang > bs_th[1] else bsl.append(0)
62
```

Figure B2: The first segment of the main code developed to calculate and test the features.

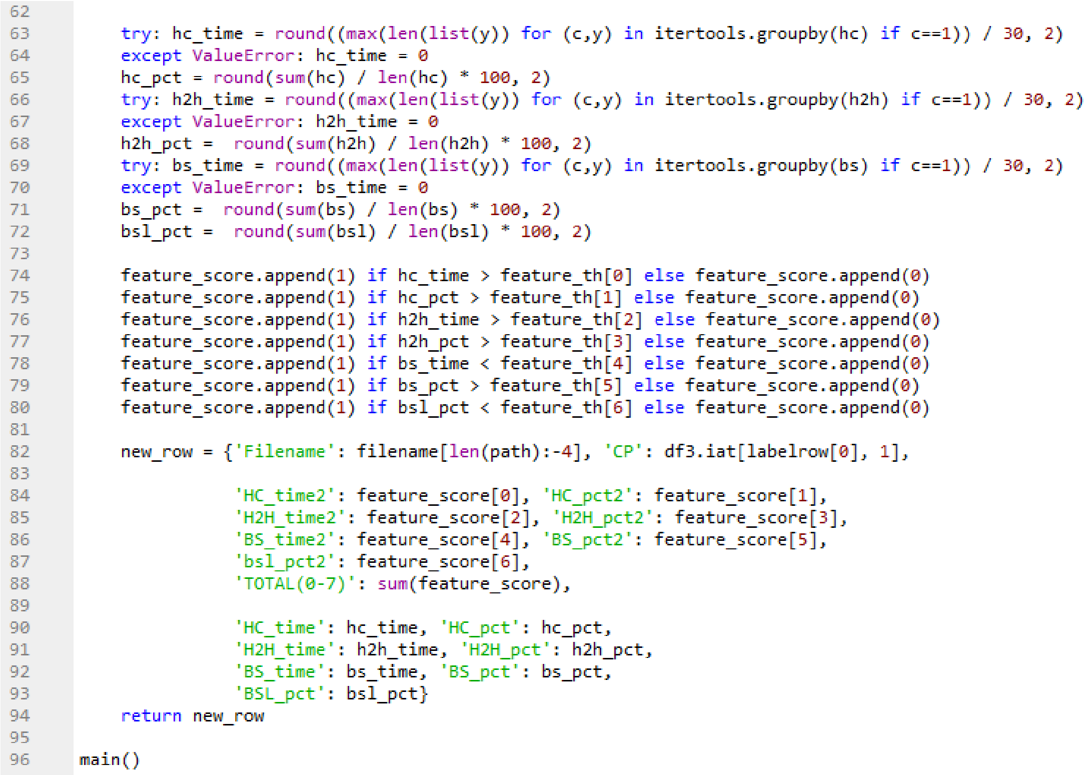
Code explanation for figure B2

25: Loop previously mentioned in figure 3, 5 and 7.

26 - 30: Creates the index lists and empty lists used during the variable, feature value and feature score calculations.

29: Reads the CP or non-CP outcome of the current file. Saved in the variable lablerow.

32 - 61: Fully integrated version of the code explained in and below figure 3, 5 7, 9 and 10.



```
62
63      try: hc_time = round((max(len(list(y)) for (c,y) in itertools.groupby(hc) if c==1)) / 30, 2)
64      except ValueError: hc_time = 0
65      hc_pct = round(sum(hc) / len(hc) * 100, 2)
66      try: h2h_time = round((max(len(list(y)) for (c,y) in itertools.groupby(h2h) if c==1)) / 30, 2)
67      except ValueError: h2h_time = 0
68      h2h_pct =  round(sum(h2h) / len(h2h) * 100, 2)
69      try: bs_time = round((max(len(list(y)) for (c,y) in itertools.groupby(bs) if c==1)) / 30, 2)
70      except ValueError: bs_time = 0
71      bs_pct =  round(sum(bs) / len(bs) * 100, 2)
72      bsl_pct =  round(sum(bsl) / len(bsl) * 100, 2)
73
74      feature_score.append(1) if hc_time > feature_th[0] else feature_score.append(0)
75      feature_score.append(1) if hc_pct > feature_th[1] else feature_score.append(0)
76      feature_score.append(1) if h2h_time > feature_th[2] else feature_score.append(0)
77      feature_score.append(1) if h2h_pct > feature_th[3] else feature_score.append(0)
78      feature_score.append(1) if bs_time < feature_th[4] else feature_score.append(0)
79      feature_score.append(1) if bs_pct > feature_th[5] else feature_score.append(0)
80      feature_score.append(1) if bsl_pct < feature_th[6] else feature_score.append(0)
81
82      new_row = {'Filename': filename[len(path):-4], 'CP': df3.iat[labelrow[0], 1],
83
84                 'HC_time2': feature_score[0], 'HC_pct2': feature_score[1],
85                 'H2H_time2': feature_score[2], 'H2H_pct2': feature_score[3],
86                 'BS_time2': feature_score[4], 'BS_pct2': feature_score[5],
87                 'bsl_pct2': feature_score[6],
88                 'TOTAL(0-7)': sum(feature_score),
89
90                 'HC_time': hc_time, 'HC_pct': hc_pct,
91                 'H2H_time': h2h_time, 'H2H_pct': h2h_pct,
92                 'BS_time': bs_time, 'BS_pct': bs_pct,
93                 'BSL_pct': bsl_pct}
94      return new_row
95
96  main()
```

Figure B3: The first segment of the main code developed to calculate and test the features.

Code explanation for figure B3

63 - 80: Fully integrated version of the code explained in and below figure 3, 5 7, 9 and 10.

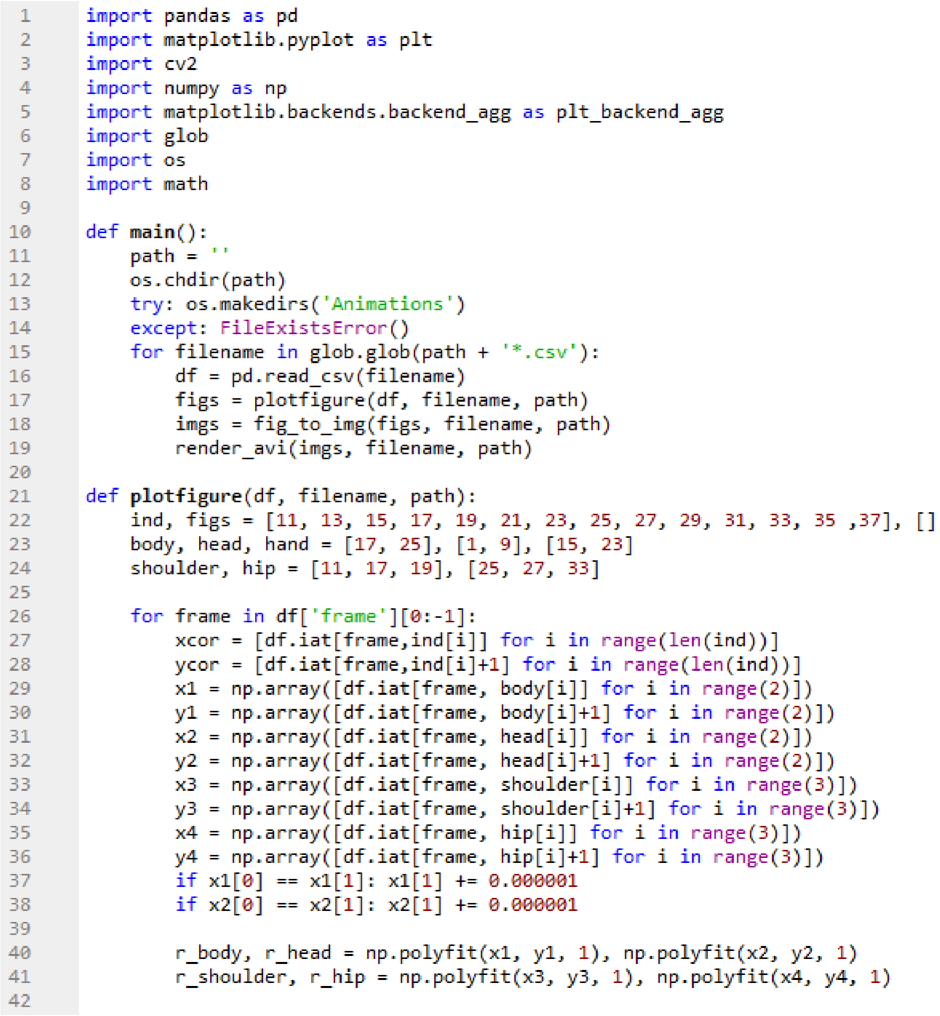
82 - 93: Creates the new_row variable that is returned and added to the new data frame. Contains the Video ID, CP or non-CP outcome as 1 or 0, feature scores, total score and feature values of the current file.

94: Returns the new_row variable to line 20 of the main segment.

96: Initialises the code starting at main in line 12.

Appendix C. Full code creating the digital rendering of the motion tracking data

The program used to convert the motion tracking data to a video representation is displayed in figure C1, C2 and C3. A short explanation of the code below each figure. The videos created was used to compare to an original video to determine upper and lower bounds for the variables before the grid search to ensure similarity between the original MOS item and the features developed here.



```
1   import pandas as pd
2   import matplotlib.pyplot as plt
3   import cv2
4   import numpy as np
5   import matplotlib.backends.backend_agg as plt_backend_agg
6   import glob
7   import os
8   import math
9
10  def main():
11      path = ''
12      os.chdir(path)
13      try: os.makedirs('Animations')
14      except: FileExistsError()
15      for filename in glob.glob(path + '*.csv'):
16          df = pd.read_csv(filename)
17          figs = plotfigure(df, filename, path)
18          imgs = fig_to_img(figs, filename, path)
19          render_avi(imgs, filename, path)
20
21  def plotfigure(df, filename, path):
22      ind, figs = [11, 13, 15, 17, 19, 21, 23, 25, 27, 29, 31, 33, 35 ,37], []
23      body, head, hand = [17, 25], [1, 9], [15, 23]
24      shoulder, hip = [11, 17, 19], [25, 27, 33]
25
26      for frame in df['frame'][0:-1]:
27          xcor = [df.iat[frame,ind[i]] for i in range(len(ind))]
28          ycor = [df.iat[frame,ind[i]+1] for i in range(len(ind))]
29          x1 = np.array([df.iat[frame, body[i]] for i in range(2)])
30          y1 = np.array([df.iat[frame, body[i]+1] for i in range(2)])
31          x2 = np.array([df.iat[frame, head[i]] for i in range(2)])
32          y2 = np.array([df.iat[frame, head[i]+1] for i in range(2)])
33          x3 = np.array([df.iat[frame, shoulder[i]] for i in range(3)])
34          y3 = np.array([df.iat[frame, shoulder[i]+1] for i in range(3)])
35          x4 = np.array([df.iat[frame, hip[i]] for i in range(3)])
36          y4 = np.array([df.iat[frame, hip[i]+1] for i in range(3)])
37          if x1[0] == x1[1]: x1[1] += 0.000001
38          if x2[0] == x2[1]: x2[1] += 0.000001
39
40          r_body, r_head = np.polyfit(x1, y1, 1), np.polyfit(x2, y2, 1)
41          r_shoulder, r_hip = np.polyfit(x3, y3, 1), np.polyfit(x4, y4, 1)
42
```

Figure C1: The first segment of the code developed to create video rendering of the motion tracking data.

Code explanation for figure C1

1 - 8: Imports all necessary python libraries.

11: States the path to the file or files that is the source for the video or videos.

12 - 14: Creates a new folder called “Aminations” in the folder containing the files.

15: Creates a loop through all csv files in the folder designated by the path variable. Allowing all csv files to be rendered one by one.

16: Creates a data frame containing the data from the current csv file of the loop.

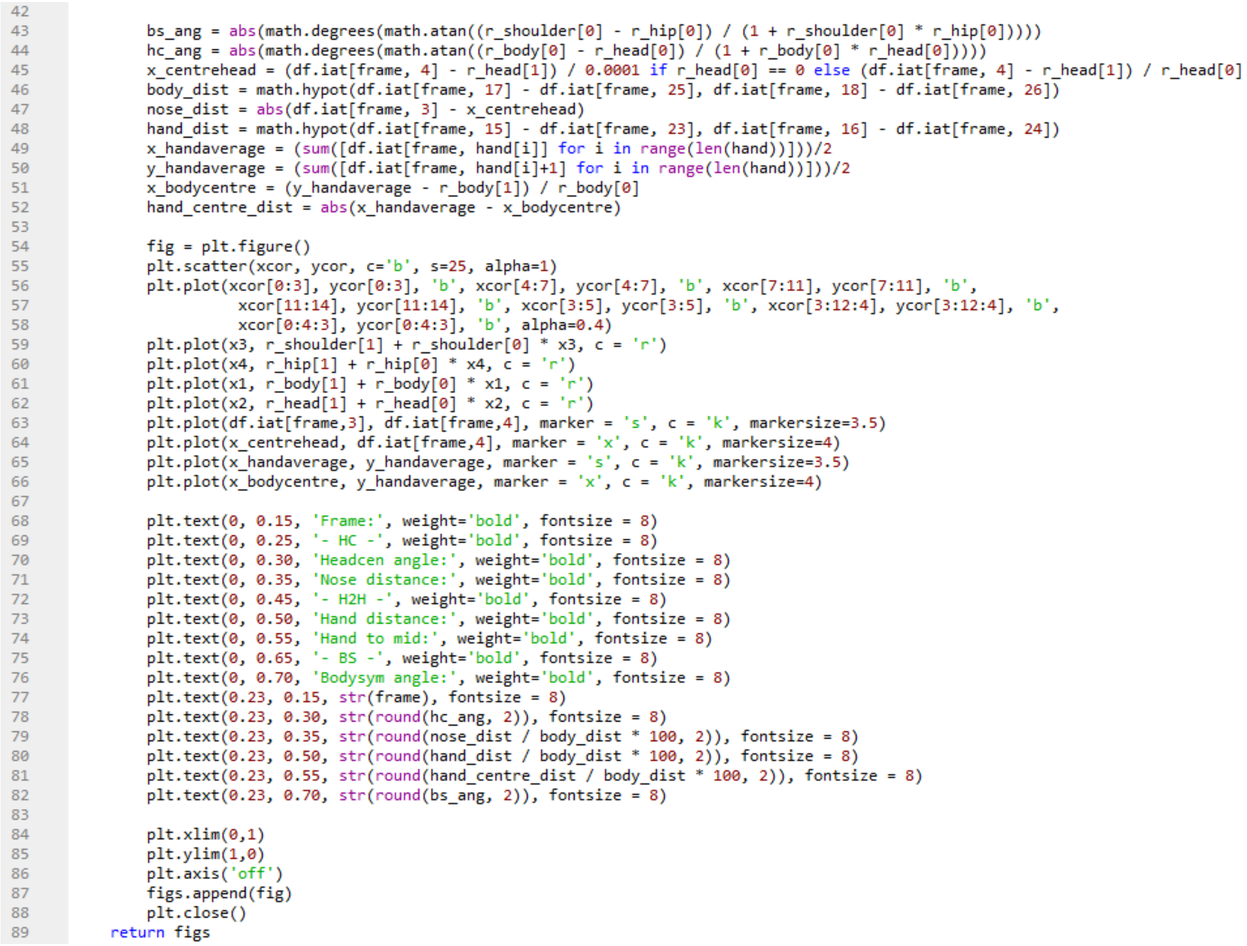
17: Sends the csv file to the plotfigure function starting at line 21. A list of is returned containing a figure per frame of the motion tracking data, stored in the variable figs.

18: The list of figures is sent to the fig_to_img function starting at line 91. A list of images is returned and stored in the variable imgs.

19: Sends the list of images to the render_avi function starting at line 103.

22: Creates a list containing the x value indexes of all body keypoints used to plot the full figure, and an empty list storing the figures created.

23 - 41: Same as in the main code and the code explained in figure 3, 5 and 7.



```
42
43          bs_ang = abs(math.degrees(math.atan((r_shoulder[0] - r_hip[0]) / (1 + r_shoulder[0] * r_hip[0]))))
44          hc_ang = abs(math.degrees(math.atan((r_body[0] - r_head[0]) / (1 + r_body[0] * r_head[0]))))
45          x_centrehead = (df.iat[frame, 4] - r_head[1]) / 0.0001 if r_head[0] == 0 else (df.iat[frame, 4] - r_head[1]) / r_head[0]
46          body_dist = math.hypot(df.iat[frame, 17] - df.iat[frame, 25], df.iat[frame, 18] - df.iat[frame, 26])
47          nose_dist = abs(df.iat[frame, 3] - x_centrehead)
48          hand_dist = math.hypot(df.iat[frame, 15] - df.iat[frame, 23], df.iat[frame, 16] - df.iat[frame, 24])
49          x_handaverage = (sum([df.iat[frame, hand[i]] for i in range(len(hand))]))/2
50          y_handaverage = (sum([df.iat[frame, hand[i]+1] for i in range(len(hand))]))/2
51          x_bodycentre = (y_handaverage - r_body[1]) / r_body[0]
52          hand_centre_dist = abs(x_handaverage - x_bodycentre)
53
54          fig = plt.figure()
55          plt.scatter(xcor, ycor, c='b', s=25, alpha=1)
56          plt.plot(xcor[0:3], ycor[0:3], 'b', xcor[4:7], ycor[4:7], 'b', xcor[7:11], ycor[7:11], 'b',
57                   xcor[11:14], ycor[11:14], 'b', xcor[3:5], ycor[3:5], 'b', xcor[3:12:4], ycor[3:12:4], 'b',
58                   xcor[0:4:3], ycor[0:4:3], 'b', alpha=0.4)
59          plt.plot(x3, r_shoulder[1] + r_shoulder[0] * x3, c = 'r')
60          plt.plot(x4, r_hip[1] + r_hip[0] * x4, c = 'r')
61          plt.plot(x1, r_body[1] + r_body[0] * x1, c = 'r')
62          plt.plot(x2, r_head[1] + r_head[0] * x2, c = 'r')
63          plt.plot(df.iat[frame,3], df.iat[frame,4], marker = 's', c = 'k', markersize=3.5)
64          plt.plot(x_centrehead, df.iat[frame,4], marker = 'x', c = 'k', markersize=4)
65          plt.plot(x_handaverage, y_handaverage, marker = 's', c = 'k', markersize=3.5)
66          plt.plot(x_bodycentre, y_handaverage, marker = 'x', c = 'k', markersize=4)
67
68          plt.text(0, 0.15, 'Frame:', weight='bold', fontsize = 8)
69          plt.text(0, 0.25, '- HC -', weight='bold', fontsize = 8)
70          plt.text(0, 0.30, 'Headcen angle:', weight='bold', fontsize = 8)
71          plt.text(0, 0.35, 'Nose distance:', weight='bold', fontsize = 8)
72          plt.text(0, 0.45, '- H2H -', weight='bold', fontsize = 8)
73          plt.text(0, 0.50, 'Hand distance:', weight='bold', fontsize = 8)
74          plt.text(0, 0.55, 'Hand to mid:', weight='bold', fontsize = 8)
75          plt.text(0, 0.65, '- BS -', weight='bold', fontsize = 8)
76          plt.text(0, 0.70, 'Bodysym angle:', weight='bold', fontsize = 8)
77          plt.text(0.23, 0.15, str(frame), fontsize = 8)
78          plt.text(0.23, 0.30, str(round(hc_ang, 2)), fontsize = 8)
79          plt.text(0.23, 0.35, str(round(nose_dist / body_dist * 100, 2)), fontsize = 8)
80          plt.text(0.23, 0.50, str(round(hand_dist / body_dist * 100, 2)), fontsize = 8)
81          plt.text(0.23, 0.55, str(round(hand_centre_dist / body_dist * 100, 2)), fontsize = 8)
82          plt.text(0.23, 0.70, str(round(bs_ang, 2)), fontsize = 8)
83
84          plt.xlim(0,1)
85          plt.ylim(1,0)
86          plt.axis('off')
87          figs.append(fig)
88          plt.close()
89      return figs
```

Figure C2: The first segment of the code developed to create video rendering of the motion tracking data.

Code explanation for figure C2

43 - 52: Same as in the main code and the code explained in figure 3, 5 and 7.

54 - 82: Plots all information on the figure.

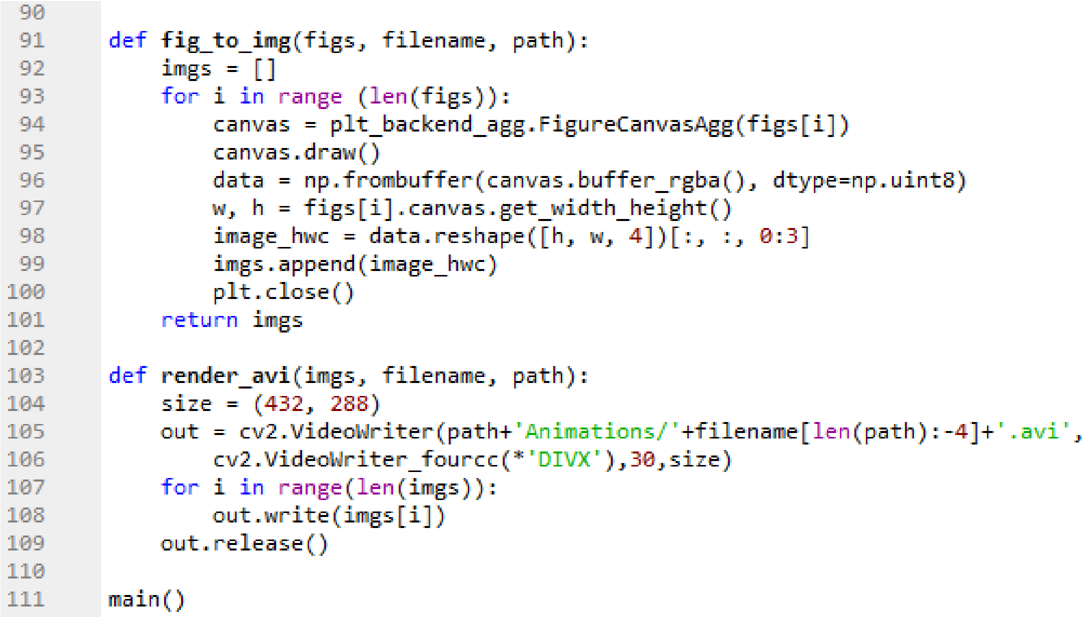
84 & 85: Flips the y axis of the plot to orient the figure correctly.

86: Removes the axis from the figure.

87: Adds the current figure to the figs list.

88: Closes the current figure before it loops back to calculate and plot the next figure.

89: Returns the list of figures to the main segment of the code.



```
 90
 91  def fig_to_img(figs, filename, path):
 92      imgs = []
 93      for i in range (len(figs)):
 94          canvas = plt_backend_agg.FigureCanvasAgg(figs[i])
 95          canvas.draw()
 96          data = np.frombuffer(canvas.buffer_rgba(), dtype=np.uint8)
 97          w, h = figs[i].canvas.get_width_height()
 98          image_hwc = data.reshape([h, w, 4])[:, :, 0:3]
 99          imgs.append(image_hwc)
100          plt.close()
101      return imgs
102
103  def render_avi(imgs, filename, path):
104      size = (432, 288)
105      out = cv2.VideoWriter(path+'Animations/'+filename[len(path):-4]+'.avi',
106          cv2.VideoWriter_fourcc(*'DIVX'),30,size)
107      for i in range(len(imgs)):
108          out.write(imgs[i])
109      out.release()
110
111  main()
```

Figure C3: The first segment of the code developed to create video rendering of the motion tracking data.

Code explanation for figure C3

91 - 100: Converts the figures received to images.

101: Returns the list of images to the main function.

103 - 109: Combines the images to a video and exports the video as an avi file.

111: Starts the program by initiating the main function.

References

1. Rosenbaum P, Paneth N, Leviton A, Goldstein M, Bax M, Damiano D, et al. A report: the definition and classification of cerebral palsy April 2006. Dev Med Child Neurol Suppl. 2007 Feb;109:8–14.
2. Oskoui M, Coutinho F, Dykeman J, Jetté N, Pringsheim T. An update on the prevalence of cerebral palsy: a systematic review and meta-analysis. Dev Med Child Neurol. 2013;55(6):509–19.
3. Novak I, Morgan C, Adde L, Blackman J, Boyd RN, Brunstrom-Hernandez J, et al. Early, Accurate Diagnosis and Early Intervention in Cerebral Palsy: Advances in Diagnosis and Treatment. JAMA Pediatr. 2017 Sep 1;171(9):897–907.
4. Palisano R, Rosenbaum P, Walter S, Russell D, Wood E, Galuppi B. Development and reliability of a system to classify gross motor function in children with cerebral palsy. Dev Med Child Neurol. 1997;39(4):214–23.
5. Hadders-Algra M, Gramsbergen A. Discussion on the clinical relevance of activity-dependent plasticity after an insult to the developing brain. Neurosci Biobehav Rev. 2007 Jan 1;31(8):1213–9.
6. Morgan C, Fetters L, Adde L, Badawi N, Bancale A, Boyd RN, et al. Early Intervention for Children Aged 0 to 2 Years With or at High Risk of Cerebral Palsy: International Clinical Practice Guideline Based on Systematic Reviews. JAMA Pediatr. 2021 Aug 1;175(8):846–58.
7. Einspieler C, Prechtl HFR. Prechtl's assessment of general movements: a diagnostic tool for the functional assessment of the young nervous system. Ment Retard Dev Disabil Res Rev. 2005;11(1):61–7.
8. Øberg GK, Jacobsen BK, Jørgensen L. Predictive Value of General Movement Assessment for Cerebral Palsy in Routine Clinical Practice. Phys Ther. 2015 Nov;95(11):1489–95.
9. Prechtl HF, Einspieler C, Cioni G, Bos AF, Ferrari F, Sontheimer D. An early marker for neurological deficits after perinatal brain lesions. Lancet Lond Engl. 1997 May 10;349(9062):1361–3.
10. Einspieler C, Peharz R, Marschik PB. Fidgety movements – tiny in appearance, but huge in impact. J Pediatr (Rio J). 2016 May 1;92(3, Supplement 1):S64–70.
11. Dimitrijević L, Bjelaković B, Čolović H, Mikov A, Živković V, Kocić M, et al. Assessment of general movements and heart rate variability in prediction of neurodevelopmental outcome in preterm infants. Early Hum Dev. 2016 Aug;99:7–12.
12. Crowle C, Loughran Fowlds A, Novak I, Badawi N. Use of the General Movements Assessment for the Early Detection of Cerebral Palsy in Infants with Congenital Anomalies Requiring Surgery. J Clin Med. 2019 Aug 22;8(9):E1286.
13. Brogna C, Romeo DM, Cervesi C, Scrofani L, Romeo MG, Mercuri E, et al. Prognostic value of the qualitative assessments of general movements in late-preterm infants. Early Hum Dev. 2013 Dec;89(12):1063–6.
14. Støen R, Boswell L, de Regnier RA, Fjørtoft T, Gaebler-Spira D, Ihlen E, et al. The Predictive Accuracy of the General Movement Assessment for Cerebral Palsy: A Prospective, Observational Study of High-Risk Infants in a Clinical Follow-Up Setting. J Clin Med. 2019 Oct 25;8(11):E1790.
15. Einspieler C, Bos AF, Krieber-Tomantschger M, Alvarado E, Barbosa VM, Bertoncelli N, et al. Cerebral Palsy: Early Markers of Clinical Phenotype and Functional Outcome. J Clin Med. 2019 Oct;8(10):1616.
16. Einspieler C, Marschik PB, Pansy J, Scheuchenegger A, Krieber M, Yang H, et al. The general movement optimality score: a detailed assessment of general movements during preterm and term age. Dev Med Child Neurol. 2016;58(4):361–8.
17. Barbosa VM, Smith EV, Bos A, Cioni G, Ferrari F, Guzzetta A, et al. Psychometric Properties of the General Movement Optimality Score using Rasch Measurement. J Appl Meas. 2020 Jan 1;21(1):17–37.
18. Barbosa VM, Einspieler C, Smith E, Bos AF, Cioni G, Ferrari F, et al. Clinical Implications of the General Movement Optimality Score: Beyond the Classes of Rasch Analysis. J Clin Med. 2021 Jan;10(5):1069.

19. Maitre N. Skepticism, cerebral palsy, and the General Movements Assessment. Dev Med Child Neurol. 2018;60(5):438–438.

20. Peyton C, Pascal A, Boswell L, deRegnier R, Fjørtoft T, Støen R, et al. Inter-observer reliability using the General Movement Assessment is influenced by rater experience. Early Hum Dev. 2021 Oct 1;161:105436.

21. Cabon S, Porée F, Simon A, Rosec O, Pladys P, Carrault G. Video and audio processing in paediatrics: a review. Physiol Meas. 2019 Feb;40(2):02TR02.

22. Marcroft C, Khan A, Embleton ND, Trenell M, Plötz T. Movement Recognition Technology as a Method of Assessing Spontaneous General Movements in High Risk Infants. Front Neurol [Internet]. 2015 [cited 2022 May 11];5. Available from: https://www.frontiersin.org/article/10.3389/fneur.2014.00284

23. Meinecke L, Breitbach-Faller N, Bartz C, Damen R, Rau G, Disselhorst-Klug C. Movement analysis in the early detection of newborns at risk for developing spasticity due to infantile cerebral palsy. Hum Mov Sci. 2006 Apr;25(2):125–44.

24. Hadders-Algra M. Neural substrate and clinical significance of general movements: an update. Dev Med Child Neurol. 2018;60(1):39–46.

25. Adde L, Helbostad JL, Jensenius AR, Taraldsen G, Grunewaldt KH, Støen R. Early prediction of cerebral palsy by computer-based video analysis of general movements: a feasibility study. Dev Med Child Neurol. 2010 Aug;52(8):773–8.

26. Rahmati H, Aamo OM, Stavdahl Ø, Dragon R, Adde L. Video-based early cerebral palsy prediction using motion segmentation. In: 2014 36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society. 2014. p. 3779–83.

27. Rahmati H, Martens H, Aamo OM, Stavdahl Ø, Støen R, Adde L. Frequency Analysis and Feature Reduction Method for Prediction of Cerebral Palsy in Young Infants. IEEE Trans Neural Syst Rehabil Eng. 2016 Nov;24(11):1225–34.

28. Stahl A, Schellewald C, Stavdahl Ø, Aamo OM, Adde L, Kirkerod H. An Optical Flow-Based Method to Predict Infantile Cerebral Palsy. IEEE Trans Neural Syst Rehabil Eng. 2012 Jul;20(4):605–14.

29. Orlandi S, Raghuram K, Smith CR, Mansueto D, Church P, Shah V, et al. Detection of Atypical and Typical Infant Movements using Computer-based Video Analysis. Annu Int Conf IEEE Eng Med Biol Soc IEEE Eng Med Biol Soc Annu Int Conf. 2018 Jul;2018:3598–601.

30. Ihlen EAF, Støen R, Boswell L, de Regnier RA, Fjørtoft T, Gaebler-Spira D, et al. Machine Learning of Infant Spontaneous Movements for the Early Prediction of Cerebral Palsy: A Multi-Site Cohort Study. J Clin Med. 2020 Jan;9(1):5.

31. Groos D, Adde L, Støen R, Ramampiaro H, Ihlen EAF. Towards human-level performance on automatic pose estimation of infant spontaneous movements. Comput Med Imaging Graph. 2022 Jan 1;95:102012.

32. Groos D, Lars A, et al. Development and Validation of a Deep-Learning Model to Predict Cerevral Palsy From Spontaneous Movements in High-Risk Infants. JAMA Netw Open. 2022;(In review).

33. Ferrari F, Plessi C, Lucaccioni L, Bertoncelli N, Bedetti L, Ori L, et al. Motor and Postural Patterns Concomitant with General Movements Are Associated with Cerebral Palsy at Term and Fidgety Age in Preterm Infants. J Clin Med. 2019 Aug 8;8(8):1189.

34. Morgan C, Romeo DM, Chorna O, Novak I, Galea C, Del Secco S, et al. The Pooled Diagnostic Accuracy of Neuroimaging, General Movements, and Neurological Examination for Diagnosing Cerebral Palsy Early in High-Risk Infants: A Case Control Study. J Clin Med. 2019 Nov 5;8(11):E1879.

35. Pascal A, Govaert P, Ortibus E, Naulaers G, Lars A, Fjørtoft T, et al. Motor outcome after perinatal stroke and early prediction of unilateral spastic cerebral palsy. Eur J Paediatr Neurol EJPN Off J Eur Paediatr Neurol Soc. 2020 Nov;29:54–61.

36. Aker K, Thomas N, Adde L, Koshy B, Martinez-Biarge M, Nakken I, et al. Prediction of outcome from MRI and general movements assessment after hypoxic-ischaemic encephalopathy in low-

income and middle-income countries: data from a randomised controlled trial. Arch Dis Child Fetal Neonatal Ed. 2022 Jan;107(1):32–8.

37. Yuge M, Marschik PB, Nakajima Y, Yamori Y, Kanda T, Hirota H, et al. Movements and postures of infants aged 3 to 5months: To what extent is their optimality related to perinatal events and to the neurological outcome? Early Hum Dev. 2011 Mar 1;87(3):231–7.
38. Sharp M, Coenen A, Amery N. General movement assessment and motor optimality score in extremely preterm infants. Early Hum Dev. 2018 Sep 1;124:38–41.
39. Fjørtoft T, Einspieler C, Adde L, Strand LI. Inter-observer reliability of the ‘Assessment of Motor Repertoire--3 to 5 Months’ based on video recordings of infants. Early Hum Dev. 2009 May;85(5):297–302.
40. Wold S, Sjöström M, Eriksson L. PLS-regression: a basic tool of chemometrics. Chemom Intell Lab Syst. 2001 Oct 28;58(2):109–30.
41. Abdi H. Partial least squares regression and projection on latent structure regression (PLS Regression). WIREs Comput Stat. 2010;2(1):97–106.
42. Cioni G, Bos AF, Einspieler C, Ferrari F, Martijn A, Paolicelli PB, et al. Early Neurological Signs in Preterm Infants with Unilateral Intraparenchymal Echodensity. Neuropediatrics. 2000 Dec;31(5):240–51.
43. Guzzetta A, Mercuri E, Rapisardi G, Ferrari F, Roversi MF, Cowan F, et al. General Movements Detect Early Signs of Hemiplegia in Term Infants with Neonatal Cerebral Infarction. Neuropediatrics. 2003 Apr;34(2):61–6.
44. Guzzetta A, Pizzardi A, Belmonti V, Boldrini A, Carotenuto M, D’Acunto G, et al. Hand movements at 3 months predict later hemiplegia in term infants with neonatal cerebral infarction. Dev Med Child Neurol. 2010 Aug;52(8):767–72.