Automatic Behavioural Monitoring in Thoroughbred horses through Pose Estimation

Jasmine Poon BVSc4
Working with Dr. Lazslo Talas at University of Bristol

BCET Beaufort Cottage Educational Trust

University of BRISTOL

Table of contents

01

Research Focus

Aims and Methodology

02

Pose Estimation

SLEAP

03

Ethogram

Behaviours we wish to monitor

04

Results

Analysis with use of Python and Boris

05

Conclusion

Challenges, improvements, what’s out there?

06

Next Steps

Learning new things and continuing research

RESEARCH FOCUS: To establish what behaviours we can automatically detect in horses using video monitoring and automatic pose estimation

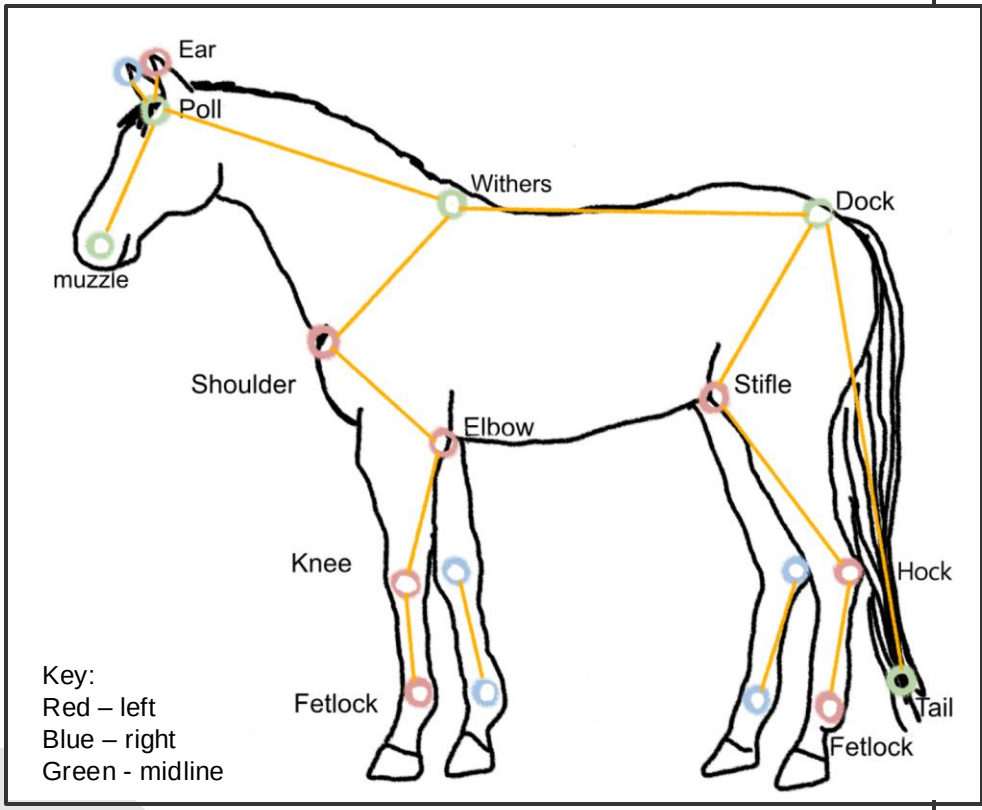
Objective: to map a horse using data points and use this to infer the behaviours observed in the video

Method: Deep learning for pose estimation – Mapping the horse
Identifying horse behaviours from data points – Coding with Python
Validation with observable behaviours in labelled video – BORIS
Reflection on AI accuracy and application in current and future time

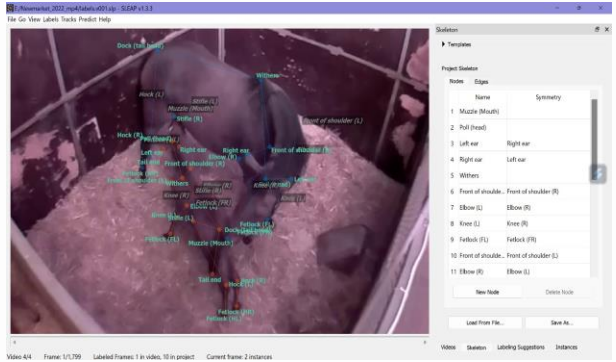
What is Pose Estimation?

Identifying and predicting the positions and orientation of objects.

Involves identifying key points and mapping their spatial relationship to reconstruct the posture of an object



SLEAP



Training

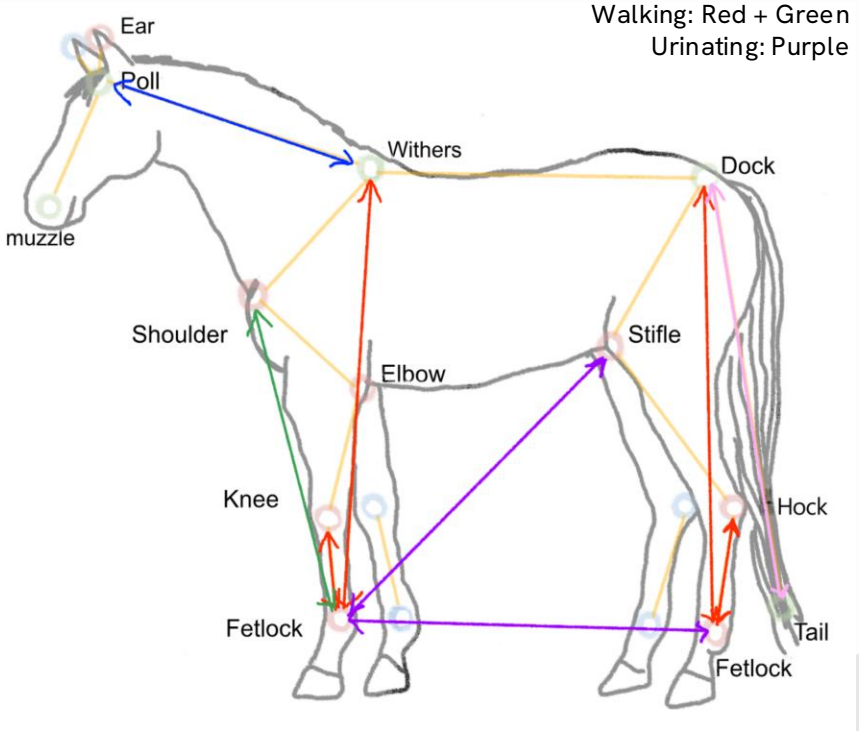
Training and inference loop

1. Label the horse(s) in each video frame using points from the pose estimation slide
2. Run training through all the labelled frames
3. SLEAP produces it's predictions of the points of the horse in new frames
4. Adjust and correct SLEAP's predictions
5. Re-train the model (SLEAP 'learns' from your corrections on it's predictions)
6. SLEAP produces new predictions after training
7. Continue loop of correcting new predictions and training the model
8. Once happy with the accuracy of predictions, predict the whole video and export a .csv file containing the data (points of the horse and their coordinates) of the predictions

Ethogram (how to convert behaviours into calculations of data points)

Standing up	Leg comparably extended. Longer wither/dock to fetlock distance compared to fixed knee/hock to fetlock distance
Lying down	Leg comparably flexed. Shorter wither/dock to fetlock distance compared to fixed knee/hock to fetlock distance
Head up	Poll above withers (y-axis – poll y value greater than withers)
Head down	Poll below withers (y-axis)
Tail flick	Rapid changes in distance between dock and tail
Walking	Changes not only in leg flexion but also fetlock position in relation to shoulders/stifle (x-axis)
Urinating	Distance between front fetlock and stifle compared to distance between front and hind fetlock

Key:
Standing/lying: Red
Head up/down: Blue
Tail flick: Pink
Walking: Red + Green
Urinating: Purple



Results from SLEAP

Example of what a frame from a predicted video looks like
(As shown, SLEAP creates a 'skeleton' that we define from the points of the horses)

Analysis

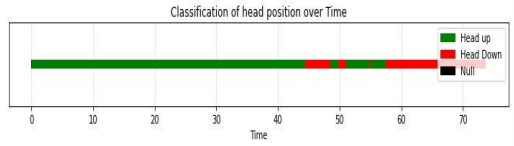
Python

Using Spyder, a python coding software and publicly available coding packages numpy and pandas to automatically run through calculations of data points

```
xDist_P_W = np.empty(N)
yDist_P_W = np.empty(N)
#Loop for calculating difference between P_x and W_x
for i in range(N):
    if array_P_x[i] == 0:
        xDist_P_W[i] = 0
    elif array_W_x[i] == 0:
        xDist_P_W[i] = 0
    else:
        xDist_P_W[i] = np.abs(array_P_x[i] - array_W_x[i])

#Loop for calculating difference between P_yand W_y
for i in range(N):
    if array_P_y[i] == 0:
        yDist_P_W[i] = 0
    elif array_W_y[i] == 0:
        yDist_P_W[i] = 0
    else:
        yDist_P_W[i] = array_P_y[i] - array_W_y[i]

#Loop for calculating difference in y value
for i in range(N):
    if array_P_y[i] == 0:
        yDist_P_W[i] = 0
    elif array_W_y[i] == 0:
        yDist_P_W[i] = 0
    else:
        yDist_P_W[i] = array_P_y[i] - array_W_y[i]

#Loop to create 2 new arrays, a new yDist_P_w with 0 points removed
# and an array of equal length with the corresponding time stamps
yDist_P_W_clean = []
yDist_P_W_clean_time = []

for i in range(N):
    if yDist_P_W[i] != 0:  # Skip 0 values
        yDist_P_W_clean.append(yDist_P_W[i])  # Append to the clean list
        yDist_P_W_clean_time.append(time[i])  # Append corresponding time


y_threshold = 0  # Set the threshold value for 'up' and 'down'
```

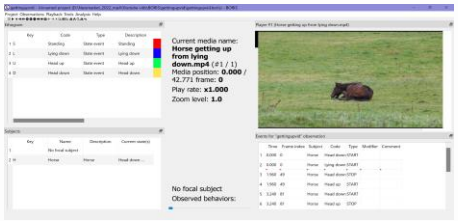
Creating graphs

Using the same coding environment, I am able to create graphs of behaviours against time using data points from SLEAP



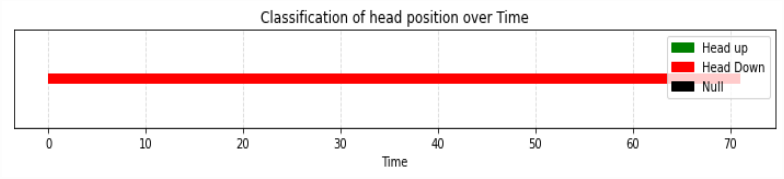
BORIS

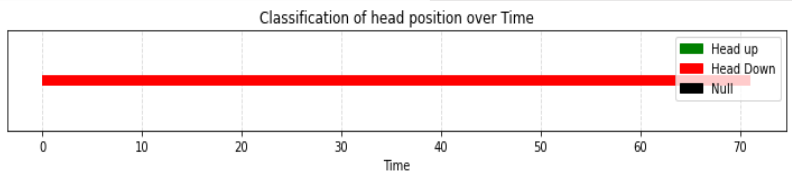
A software that allows manual labelling of behaviours occuring during a video and converts it into a graph against time

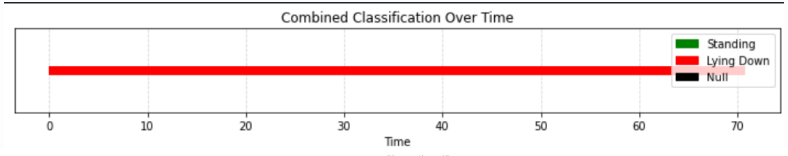


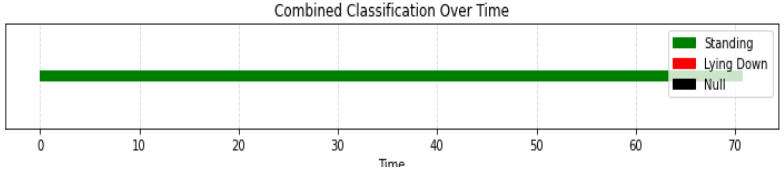
FOAL

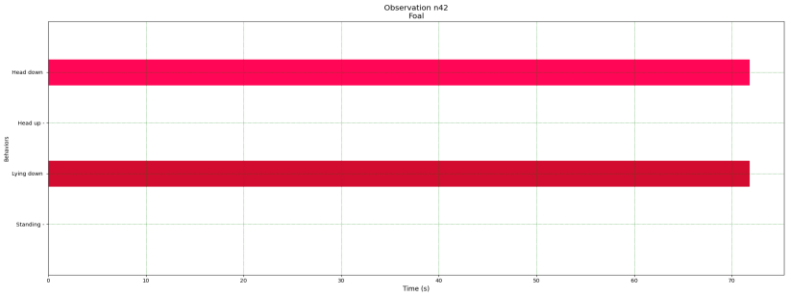
MARE

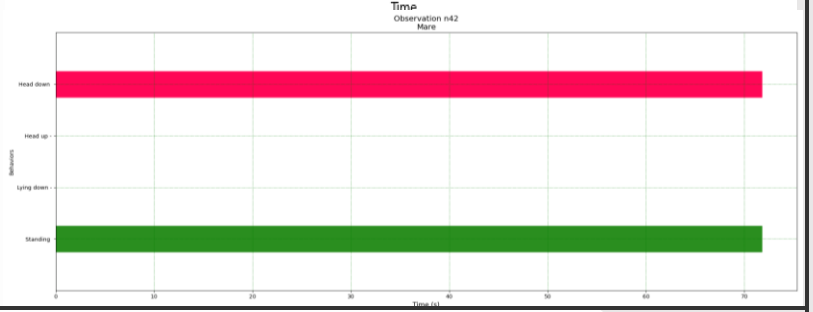












Example of behaviour analysis conducted

Top two graphs: from Python and SLEAP data, Bottom graph from BORIS by observing video

Challenges

Labelling in SLEAP

Ideas to make it more user friendly, improving labelling efficiency

Working in different computers

Limitations in multiple people working on the same project due to video file path.

Better Pose Estimation

Eliminating background and allowing single horse tracking using programmed video editing loop?
More data points?

Skeleton vs Silhouette

Is pose estimation and training AI to detect a horse's skeleton the better way?
Sleip
Novostable

Next steps

1. Improving user interface for labelling

We are writing our own Python package tailored specifically to label horses

2. More data

Monitoring horses at Langford using low-cost open-source cameras

3. Increasing ethogram complexity

Adding more detailed behaviours (e.g. signs of stress) and describing them mathematically

Literature review

- Pereira, TD, et al. (2022). SLEAP: A deep learning system for multi-animal pose tracking. *Nature Methods*, **19**, 486-495.
- Friard, O, & Gamba, M. (2016). BORIS: a free, versatile open-source event-logging software for video/audio coding and live observations. *Methods in Ecology & Evolution,* **7**(11), 1325–1330.