MSA UNIVERSITY
جامعة أكتوبر للعلوم الحديثة والآداب

UNIVERSITY of GREENWICH

October University for Modern Sciences and Art

Faculty of Computer Science

Graduation Project

Title: Horse-Posing Estimation

Supervisor: DR. Islam ElShaarawy

Name: Osama Khalil Abbas

Abdullah Amin Madi

ID's: 203039

203243

Table of Contents

Table of Figures

Table of Tables

Abstract

The advance of knowledge show many ways to understand the nature of the animal in the environment designated for it. This led to a strong desire to learn about some of the different patterns and behavior of many animals. One of the most important of these methods is how to estimate the situation of the animal in various situations. Based on this estimate many numerous studies carried out to determine some of the features that can extracted from an animal to identify many methods. This may help humans to know if an animal needs its help. Moreover, the pose estimation for living organisms done through the world of deep learning and deep networks. In addition, tracking animal status contributes to the classification of some unrecognized animals and divides them into classified and unclassified categories. This greatly contributed to provide a huge amount of data. By supporting processes that depend mainly on estimating the condition of the animal in different conditions to achieve the best results. Use many frameworks to track or estimate the status of different animals, the most prominent of which is the framework called SLEAP. An automatic tracking system relies on data classification and inferring other new data.

Chapter 1: Introduction

1.1 Introduction

In the modern technological age, with the flourishing of the world of deep learning and artificial intelligence. Taking advantage of the ability of computer vision to interact with the physical world, computer vision has contributed effectively to the development of estimating the attitude or behavior of many objects. As states by Söderström (2021) [10] by resorting to various fields of artificial intelligence and machine learning to understand and describe determine the content of videos and photos. Given the importance of status-tracking applications in biology and the environment, this carry out a huge role in helping people easily interact with animals. Although this process can be perform with great precision using the human element, this usually requires enormous time and cost and simple processing of some data. Accordingly, computer science has been rely upon to overcome these problems and obtain results that are more accurate. Also relying on some advanced tools that determine the frames of video clips and take into account the contrast in the shape and body of the animal. The horse is a powerful animal and has many uses throughout people's lives for example: racing horses and horse jumping, so the intended animal is the horse. At present, with remarkable development, there is a lack of applications to determine the behavior of animals. In addition, many of these applications may not include dealing with animals with relatively complex shapes and structures or with rare data. As mentioned by Mathis et al (2021) [5] some good results can based on many poses estimation. The situation has assessed for several animals of enormous importance to humans, the most important of which are (dogs, cats, horses, cows, etc.) Figure 1 show that.

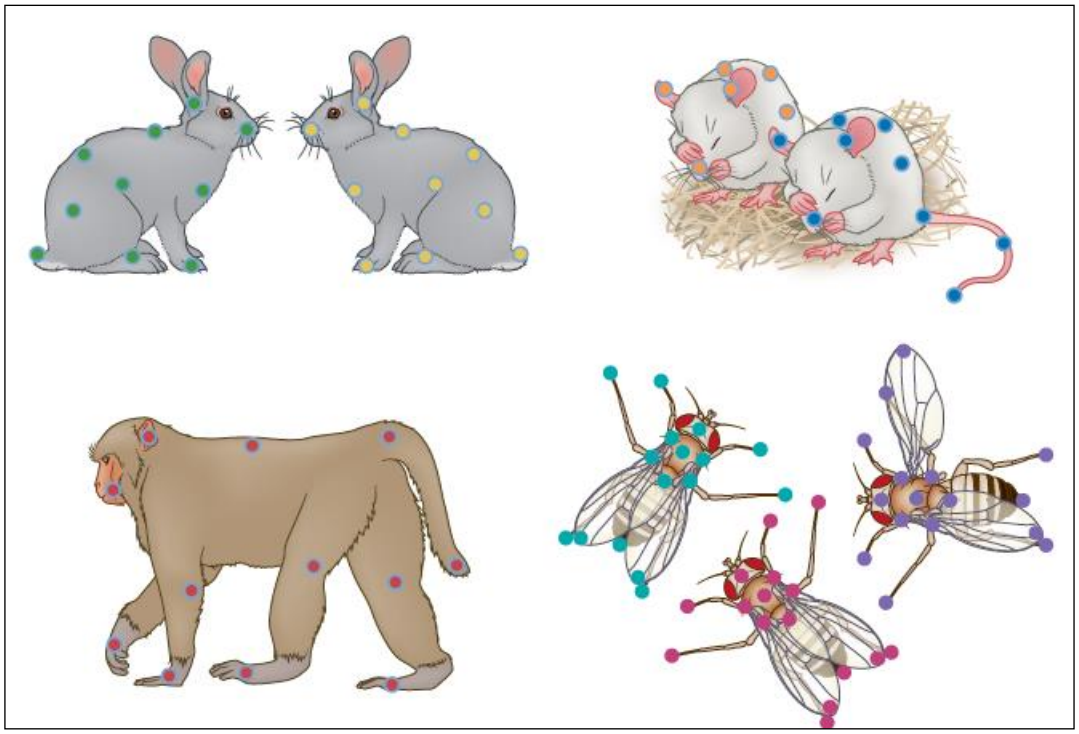


Figure 1 pose estimation for different animal.

Each work has strengths and weaknesses to show credibility of this work. First, the advantages of determining the behavior of horses. Pereira et al (2020) [6] state that the horse's behavior allows to recognize its nature better and monitor it in the environment in which it lives. It also contributes to identify injuries that may be inflicted on horses and working to treat them before they aggravate. Secondly, a gap may appear during the assessment of the status of real and artificial horses. In addition, there is a difference in identifying some parts of the horse's structure, and a defect may occur in the expected results.

1.2 Problem statement

Pose estimation faces many problems for four-legged animals, especially horses. These problems lie in the presence of many joints that are difficult to cope with in a unique profile, and as reported by Cao et al (2019) [1], the positions that are valued, such as running and jumping, may differ in different horses. In addition, problems arise while performing training on a variety of seeded samples. As said by Pereira et al (2020) [6], these problems appear clearly after training on the current classified data, which obtains the unseen data that cannot be identified with tags. Where those problems are reflected in the process of identifying the different behavior performed by the horse

1.3 Objective

The project should demonstrate the entire pipeline of the way of working. It receives labeled data extracted from the dataset. Then it based on the trained model to reuse in predicting the unlabeled data, which is a new video in which pose estimation has not done before. By use these results, it is easy to identify the different behavior of horses and identify their problems.

1.4 Motivation

This paper is concerned with the utmost importance of tracking the status of horses to design a map that shows how to deal with the world of this wonderful animal, and how to work on the primacy of knowing whether a horse needs treatment due to behavioral change. Due to the strong structure of this organism and the dependence on it in various fields, as mentioned previously, this prompted us to focus and pay great attention to improving the assessment of the situation for this animal to preserve its life

1.5 Thesis layout

Part two of the chapters will include a literature review, background, as well as the related work of other researchers' papers, Moreover, chapter 3 will be about the materials, which are data and tools used on the project, and there is will be methods for the project.

Chapter 2: Literature Review and Background

2.1 Background

Currently, technology plays an important role in determining much different behavior of the animal world. It contributes to reduce the gap between humans and animals. Accordingly, some modern technologies have employed to carry out these tasks. Hence the turning point in relying on computer vision, deep learning, and neural network. This algorithm works to estimate the situation and determine the different behaviors of horses.

2.1.1 Computer vision

Computer vision is a category of artificial intelligence that aims to build and reuse digital systems based on processing and interpreting visual data. It aims to reach computers with the highest degree of accuracy in identifying objects and people in digital images, videos and making appropriate decisions. Figure 2 show that. Furthermore, computer vision is approximately the same as the human eye. However, human vision gains a major advantage that makes it distinguish things from each other more accurately. Finally, computer vision aims to train machines on huge amounts of data to perform the required tasks accurately. This achieved by using two essential techniques of machine learning, CNN and SIFT. CNN takes into account the data in the training data, by extracting the most useful features for solving a given task. SIFT works to extract features from images by following several sequential stages, including Building the scale-space and Difference of Gaussian.

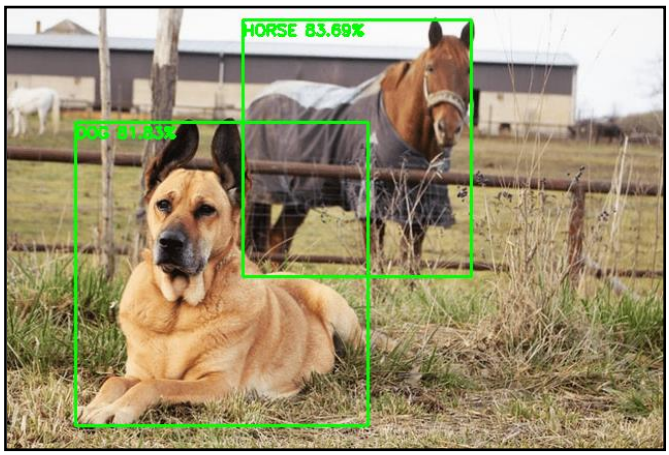


Figure 2 Object detection by computer vision.

2.1.2 Deep Learning

Deep Learning has become an increasingly popular method for pose estimation in recent years because it can accurately detect and track complex body movements. The heart of deep learning is the neural network. It simulates the way the human mind works in terms of learning and thinking. Neural networks draw their power from training data to learn and improve results over time. The neural network consists of an input layer, multiple hidden layers, and an output layer Figure 3 show that. All nodes in the layers act as linear regression models, and each node contains weights and biases. After passing through the input layer, weights are assign, and these weights contribute to determining the importance of the variables. These variables support the process of comparing inputs and outputs. The transition takes place in the next layer in the check. Therefore, we have a node that exceeded a certain limit, and it used as input for the next node. The process of passing data between layers of a neural network called network feeding. The majority of deep neural networks are automatic neural networks, and they work on using pre-trained data, as

inputs by reaching results (outputs) .There are different types of NNs (Neural Networks) which is feed forward NNs, Recurrent NNs, and Convolutional NNs.

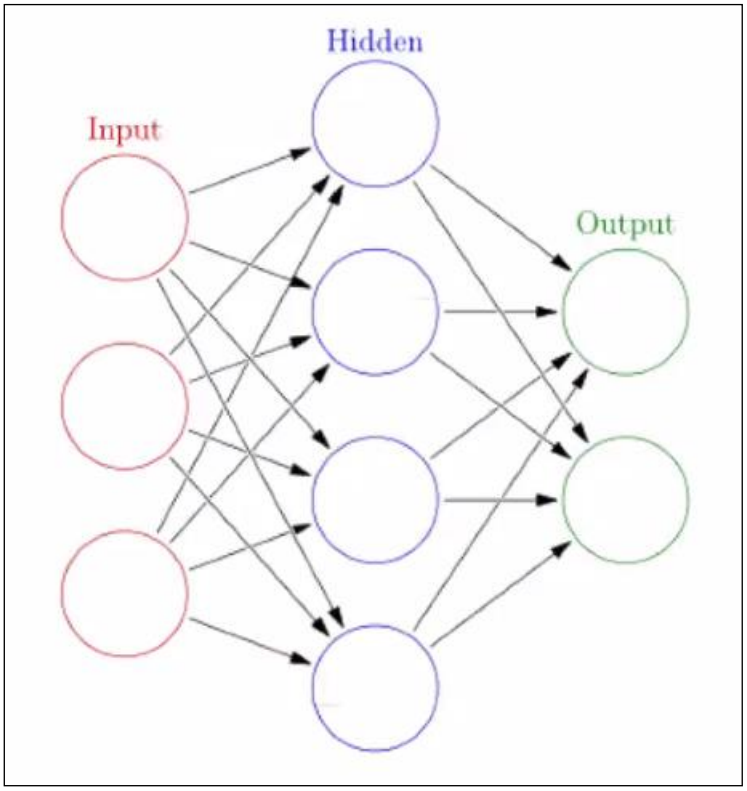


Figure 3 Neural network Layers.

2.1.3 SLEAP Framework

Advances in the science of machine learning are playing an important role in the development of numerous frameworks that address the challenges of understanding animal behavior in their natural environment. Here we present SLEAP, a machine learning framework for tracking animal behavior. This system is based on the use of classified data for training in unseen data discovery. It also has an easy-to-use graphical interface, a repeatable system with 30 uniform model structures and two

main methods for combining parts. As mentioned by Pereira et al (2022) [8] the SLEAP efficiency was tested on seven pieces of data from different organisms, including mice and bees, as it achieved a high accuracy of more than 800 frames per second and a transmission time of 3 milliseconds. There are two approaches to estimating the mode for multiple instances. The methods differ according to the animal and its body parts. The names of these methods do not refer to the way the camera is installed, but to the way it works. As said by Pereira et al (2020) [7] bottom-up approach first recognizes all the parts of the animal in the image and then links them together As shown in figure 4. It is characterized by going through the neural network once, leading to convergence fields and confidence maps that show the spatial relationship and interdependence between the parts of the animal's body. The confidence maps are used to indicate the coordinates of each part to be combined. The training process takes place in two stages. First, the neural network recognizes a linkage point, which is often in the middle. Second, a partial image of each animal is created to use as input for the second network. Secondly, as stated by Pereira et al (2020) [7] a top-down approach is used in which instances in each form are first discovered. The outputs of these instances are placed in the middle of each of them and instances may overlap as shown in figure 5. This process is very important because in the second phase it serves as an indicator of the parts that can be predicted in the image. A named part is selected that is in the center of the virtual box that surrounds the animal. The first stage of this approach targets the selected segment, which is trained to predict multiple confidence maps. The confidence maps are later converted into coordinates that are cropped from the image surrounded by the standard box. These images are referred to in the second phase to determine the instances that depend only on this image. Then the associated coordinates are extracted later.

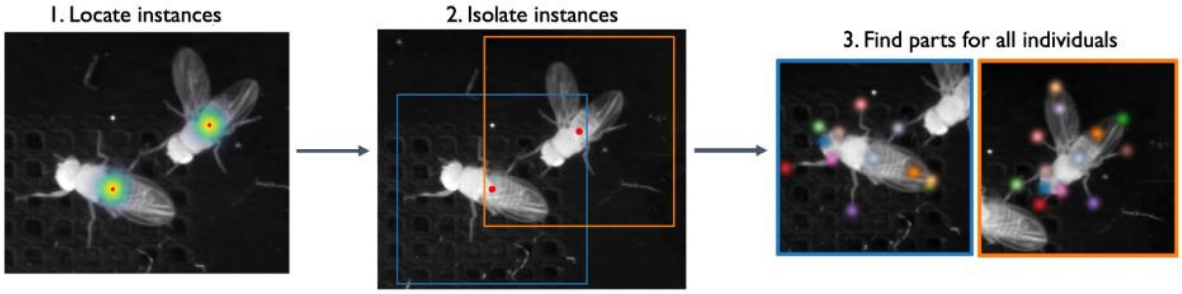


Figure 4 Top-Down approach

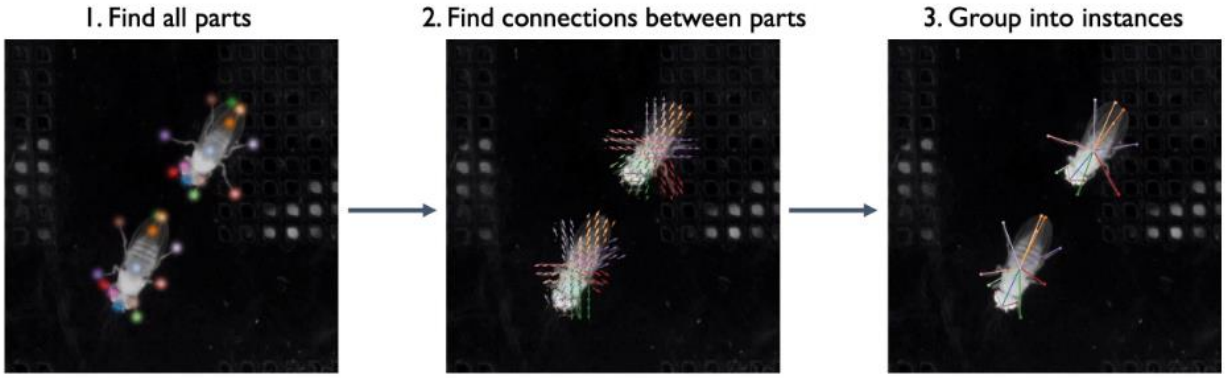


Figure 5 Bottom-Up approach

2.2 Previous work

In this section, we will present other research related to the work of various animals and human pose estimation approaches. We will also evaluate and analyze the literature review sources to be clear.

2.2.1 Pose Classification of Horse Behavior in Video… [11]

2.2.1.1 Strategy and Structure

This thesis aims to deepen the role of technology in identifying animal behavior. Where they worked on classifying many horse behavior to improve its well-being. Horses do not have the ability for verbal expressions, so they require a great deal of experience and knowledge to understand their behavior. This paper based on the Deeplabcut framework toolbox like a leap, DeeperCut, OpenPose, and DeepPoseCut. It also based on the openCV framework it used to get metrics from the saved data. A network called the MLP (Multi-layer perceptron) is created to classify and examine random samples. Figure 6 show that

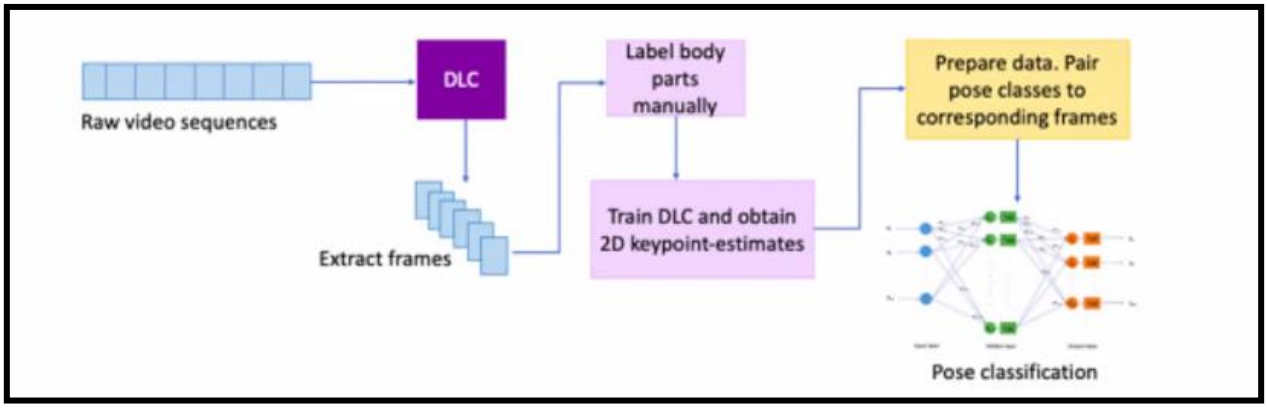


Figure 6 Structure of DLC Framework.

2.2.1.2 Data

These researchers used videos as the nature of their data, as they were 4 videos, each video is 45 minutes. This data were collected by installing 4 cameras around the horse's chest to record different movements of horses like standing, eating, and kicking. This data are recorded at 20 frames per second at a resolution of 1520 x 2688 and it is used as labeled data. Figure 7 show how data collected.

Figure 7 Some of horse picture for the collected data.

2.2.1.3 Method Evaluation

They were able, using DeepLabcut and 2D-Keypoints, to improve the classification of behaviors and the assessment of different positions of horses. Noticed that the DeepLabCut is able to identify a large percentage of the parts of the horse's body. This is with a small error rate of 54 pixels.

2.2.1.4 Results Evaluation

These researchers extracted a high percentage compared to the random score percentage and its accuracy equaled 87%. Some of them emphasized it was an impressive percentage because it contributed very beneficially and it may be influential in the future in terms of applications.

2.2.2 Equine Pain Behavior Classification via Self-Supervised Disentangled Pose Representation [3]

2.2.2.1 Strategy and Structure

Detecting horse pain is important for the horse's physical safety and early detection of diseases. Horses also express their problems through a change in behavior, facial expressions, and body movement. In addition, the owner's inexperience may hinder

the diagnosis of equine pain and impede the possibility of urgent medical intervention. Two approaches were used to perform this process. The first approach is to build a new display synthesis decoding architecture and learn about the fixed appearance of the horse in order to be able to recognize the new horse's body. As for the second approach, the results extracted from the trained architecture are relied upon to be used in building a multiple instance learning (MIL) models that classifies the different pains experienced by the horse. Figure 8 show that

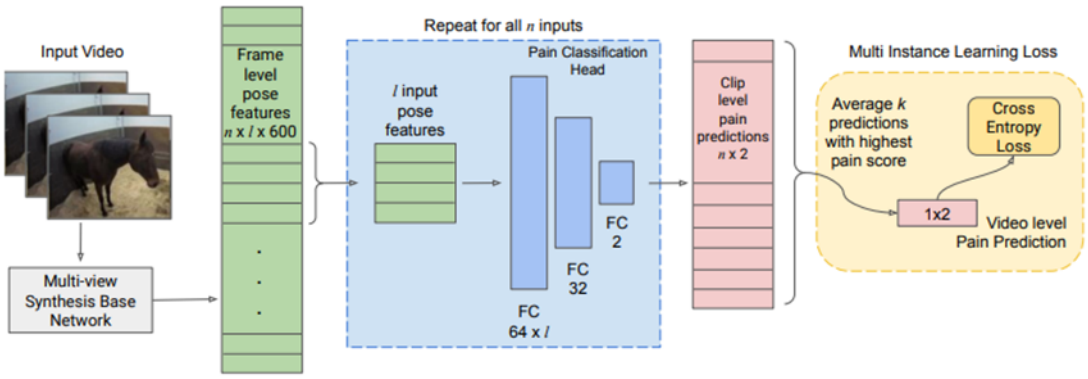


Figure 8 Model for horse pose estimation and MIL classification

2.2.2.2 Data

The Equine Pain (EOP) dataset contains eight horses that were monitored for 24 hours over the course of 16 days. These horses were injected with substances to induce joint pain, according to the protocol of Swedish legislation regarding animal experiments. As reported by Maheen Rashid et al, horses were injected with lipopolysaccharides (LPS) into the hock joint to cause lameness and inflammation of various degrees. Then, after more than an hour and a half, the horses are taken out to monitor the change in coordination of movements according to four different pain scales [13, 17, 63, 6]. A session with the highest average CPS score was selected as

the peak pain period. The closest two hours of the pain monitoring session were taken as evidence of pain, resulting in 128 hours of monitoring. Figure 9 show that

Figure 9 horse have a pain in his bones.

2.2.2.3 Method Evaluation

The network structure consists of two main layers coupled with ReLU, followed by a 2D main classification layer with or without pain. The sections that contain the horses with the highest pain prediction are averaged in order to obtain a high-level pain prediction. Also, the value of the loss is relied upon, as Maheen Rashid et al, The main goal of the design of the loss is that the horse that suffers from pain does not continuously express pain, and the absence of its painful behavior can be designed as not suffering from pain. The lower the value of the loss, the better the results, and the greater the loss, the lower the accuracy.

2.2.2.4 Results Evaluation

Classifying the situation of horses and identifying their different behaviors, especially pain, is one of the difficult problems. Where the human performance in the process of identifying pain related to horses through video clips achieved approximately 58%. Urging the researcher was able to classify many behaviors as pain or without pain or severe pain. It also achieved approximately 51% in bone

pain. As for the researchers in charge of this work, they achieved a good percentage in identifying and tracking the horse's behavior, which amounted to 60%.

2.2.3 Cross-Domain Adaptation for Animal Pose Estimation [1]

2.2.3.1 Strategy and Structure

This paper aims to present a method for estimating the state of the data inside and outside the field, in other words, trying to label the undefined data using the known label that has been trained. The motivation for this work was to reduce the process of labeling data on animals due to the scarcity of data. The Weak Domain Semi-Supervised Adaptation (WS-CDA) scheme is designed to mitigate the problem of mode estimation to better learn about common features across the domain Figure 10 show that. Then using the "pseudo-label-based progressive optimization" (PPLO) strategy designed to enhance the performance of the model to increase the amount of data.

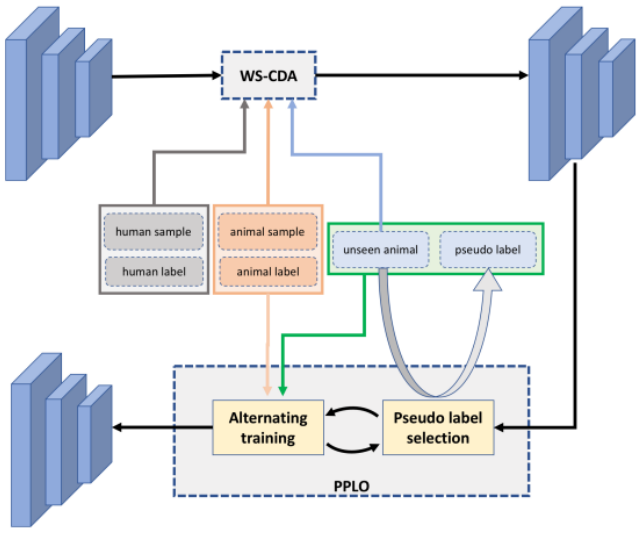


Figure 10 Workflow of system pose estimation using WS-CDA and PPLO

2.2.3.2 Data

Since data on animals are scarce, and if it is available, it may not contain the markers that the team relied on in tracking an animal's condition; researchers may be tempted to build evidence. Fortunately, for the researchers who did this work, the dataset of the classified instances from VOC2011 is publicly available. This data contains five available animals, which are dog, cat, horse, sheep, and cow. In this dataset, 5,517 instances of these five categories are distributed over 3,000 images and each instance contains 17 points.

2.2.3.3 Method Evaluation

To evaluate the performance of the model (WS-CDA), several different and varied data are entered from several fields, all of which use the AlphaPose framework. Then, 1117 cases are selected from the data set in order to test the model after training. The model showed a failure during its testing on animals, so the data was increased and data related to humans were merged with it to enhance the performance of the model.

2.2.3.4 Results Evaluation

The model showed great progress after increasing the data and relying on (PPLO) in the results of a huge amount of data. In addition to taking advantage of common features between human and animal structures to reduce model misleading. The model based on the AlphaPose framework achieved acceptable results with cats, dogs, and horses without relying on (PPLO) as the mAP was 37.6, 37.3, and 47.9. It also achieved, based on WS-CDA + PPLO, results of 42.3, 41.0, and 53.1.

Chapter 3: Material and Methods

3.1 Materials

In this part, we will outline what can be used to successfully complete this project. In terms of the quality of the data used, the working environment chosen and the tools used. Finally, we will present a diagram that proves the nature of the project work.

3.1.1 Data

3.1.1.1 Horse-10 Dataset

These data were used because it is appropriate in the process of estimating the condition of the horse. We uploaded a file consisting of 30 files, each containing about 300 to 400 images. All these files occupy a storage space of 900 MB. This data is used as labeled data. In addition, some videos are used to ensure the validity of the training, estimated at 5 videos. Figure 11 show that

Figure 11 show Horse-10 Dataset

3.1.1.2 Animal Kingdom Dataset

This is a huge and diversified data collection for the animal kingdom that includes numerous annotated activities to help researchers gain a better understanding of natural animal behavior. Animals in the wild have been captured at various times of the day in a variety of habitats with varying backdrops, angles, lighting, and weather conditions. More precisely, for the video grounding test, the dataset comprises 50 hours of annotated movies for discovering significant animal behavior snippets in lengthy videos, 30 000 sequences, for the multi-marker fine action identification job, and 33 000 frames for the posture estimation task, which correspond to With 850 species spread over six major animal classes, it has a diverse animal population. This because the data set comprises a large number of animals, so we applied a filter to the data to extract only horse-related information. We took around 1500 frames from it for various horses in various circumstances, which we used in the pose estimation procedure. Furthermore, we were able to collect over 100 movies that were used to assess the model's quality and utilize its results to decide behavior. Figure 12 show that

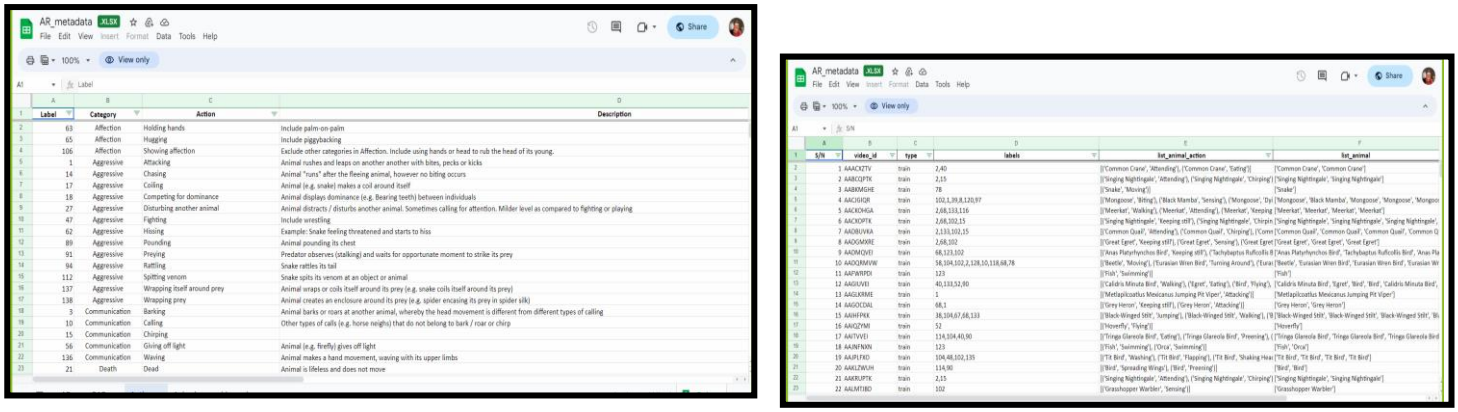


Figure12 Metadata of Animal Kingdom dataset

3.1.2 Tools

- Google Colaboratory: An analog interface for dealing with Python, and it is also used to deal with deep learning and computer vision problems.
- SLEAP: It is a framework based on implementing two pipelines, which are bottom-up and top-down with over 30 standard neural networks. He is considered a pioneer in the field of classification and tracking the behavior of many organisms.
- Python 3: An open-source, sophisticated programming language with a wide range of functions, on which the most important computer sciences, such as Deep Learning, are based.
- Scikit-Learn: is an essential Python library that is commonly used in machine learning projects. it is primarily concerned with machine learning tools, such as mathematical, statistical, and general-purpose algorithms, which serve as the foundation for many machine learning technologies.
- JMESPath: AWS CLI, AWS Python SDK, and AWS Lambda Powertools for Python employ JSON query language. Its methods are included to help you quickly deserialize typical encoded JSON payloads in Lambda functions.
- NumPy, Pandas, Matplotlib

3.1.3 Environment

- Cloud: GPU of Google Colaboratory.
- We use the GPU runtime in Google Colab, which includes a 2.20 GHz CPU Intel Xeon processor, its RAM is 13 GB, a Tesla K80 accelerator, and 12 GB of GDDR5 VRAM.

3.2 Methods

In this section, we will discuss how the process of our project has been done in pose estimation for horses. In addition, we will explain the solutions and strategies we used.

3.2.1 System architecture Overview

The Horse-10 dataset and some videos were used to pose estimation the horses. First, a model is built using the SLEAP API framework and then trained with labeled data. Second, a video is inserted to obtain predicting labeled data in the input video. Finally, there is a video output with new bone structures in the horse that was predicted from the labeled data. Figure 13 show that

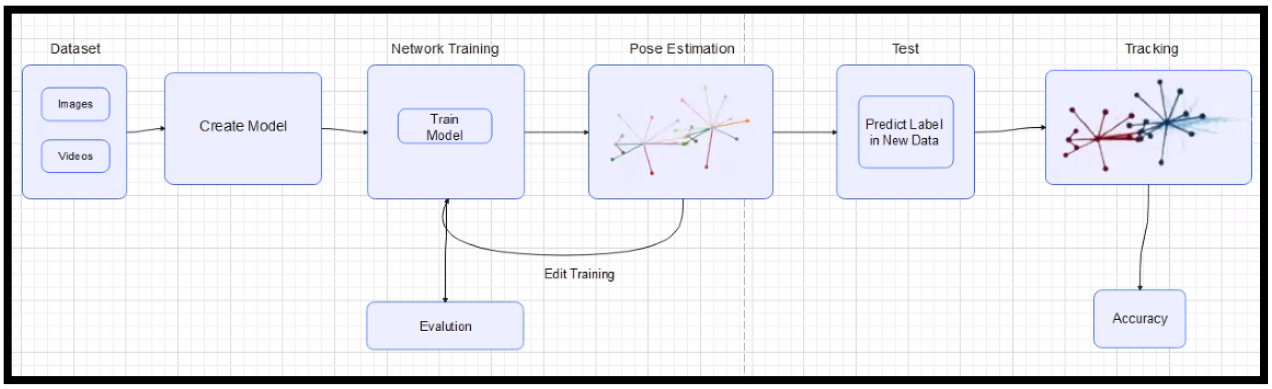


Figure 13 Pose Estimation system architecture Overview

Chapter 4: System Implementation

4.1 System Development

The development of this system includes several successive and different stages. In the beginning, we prepared the work environment to deal with Social LEAP Estimates Animal Poses (SLEAP). Then we created a simple model based on the data provided by developers such as flies and mice, which is the Benchmark data set, and this model achieved good results. Secondly, we started the process of collecting the necessary data to distinguish and track the behavior of horses, and this data was obtained from two sources: animal kingdom and Horse-10. This data was the beginning of the system development journey. Using the toolbox in the SLEAP GUI, we imported the data related to horses (Horse-10). SLEAP allows us to deal with different data formats such as ".slp", ".json", and ".h5". In addition to controlling the locations of labeled nodes and creating skeletons on the body of the horse for each frame in the video. Then labeled is selected frame randomly is based on the prediction process. Then we start the training process. Through the training process, SLEAP Framework allows us many pipelines such as multi-animal topdown, multi-animal bottom-up, and single-animal. It also provides high parameter control. A model was constructed using a multi-animal top-down pipeline and it achieved the desired results. In addition, taking advantage of the notebook created by the developers, we went to use google colab to train or re-train the previously created model. Thirdly, some files (best_model.h5) are used from the model to convert it into TensorFlow Lite Model. This is done by relying on the two libraries TensorFlow and NumPy to check the quality of the model. These models are exported in order to, employ them in different applications. Beginning with this step, we turned to a dataset called Animal King Dome. This dataset contains large groups of animals and is divided into several categories, so we made a filter on the JSON files. There are two main ways that made it easier for us to obtain the required

data: the first is by using JMESPath, and the second is by dealing directly with JSON files through the Python code. As for JMESPath, it is a query language that allows you to extract data declaratively from JSON documents. JMESPath includes official ABNF rules and a full set of compliance tests to ensure parity between libraries. It is similar to XPath for XML documents. This is an example of a typed query ({

"info":info,

"licenses":licenses,

"images":images,

"categories":categories,

"annotations":annotations[?contains(animal_subclass, 'Horse')]}).After that, the results of the filter are dealt with, as they are submitted to the SLEAP framework to build a model on them that will be used later to perform the prediction process. After the prediction results are obtained, they are converted into JSON files using the SLEAP command line. Here comes the role of direct interaction with JSON files using Python. It is a very important process that prepares the data for the stage of determining behavior. In this process, the coordinates of the available points in the frame are extracted and then linked to the multiple behaviours of the horses. Finally, we save this data in a CSV file. Then we are ready to build a multi-class, multi-output model using XGBoost. The term "Extreme Gradient Boosting" refers to this "XGBoost". A distributed scaling boost library that is optimised for speed, scalability, and portability is called XGBoost. Gradient boosting is used to develop machine learning algorithms. In order to quickly and accurately tackle a variety of data science challenges, it provides parallel tree boosting. We call some important libraries that help us complete this task, such as Pandas, Seaborn, Xgboost, sklearn.model_selection, sklearn.pipeline, Matplotlib, Numpy, sklearn.metrics, and sklearn.multioutput. Some of the previous libraries read and formatted the data saved in the CSV file that we created earlier, such as Pandas. Others show distributed data,

such as Matplotlib. In addition, the data is divided into parts for training data and others for test data through the sklearn.model_selection library. Thus, we have reached the stage where the parameters on which the XGBoost model depends and the fitting of the training data are prepared. Finally, the model is tested to verify its accuracy. Figure 14 show that

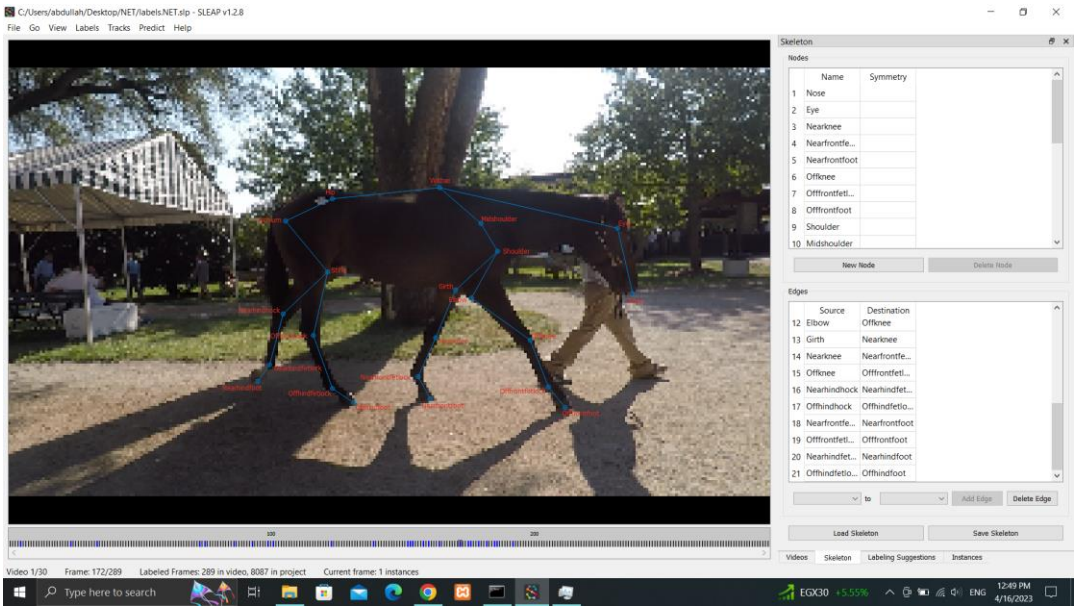


Figure (14) Output of horse pose estimation

4.2 System Structure

In this section, we will briefly explain the upcoming subsections. The first subsection shows our final system overview that we developed and also shows how the building process of the system has been done. The second subsection will clearly declare the system classes (class diagram) that were used to do a specific task on our project.

4.2.1 System Overview

This system consists of six main stages. The first stage is responsible for preparing the data and ensuring that it can be used on the SLEAP Framework. SLEAP allows us to use the previously prepared data (labelled data) or create a user-labelled one. This process is done by writing the names of the nodes (eye, shoulder, knee, etc.) that are important in the horse's body, and then they are linked together until they form a skeleton. However, through our use of the two previously mentioned data sets, we rely on a skeleton consisting of 23 points distributed over the horse's body, which was prepared accurately. In addition, we do a filtering of the data in the Animal Kingdom data set, because it is a huge data set consisting of more than one million and two hundred thousand images of multiple and different animals, to extract from it data related to horses only. The second stage is divided into several steps. First, we prepare a labelled suggestion data set in order to use it later to evaluate the model. Secondly, we prepare some parameters in order to use them in creating the model. For example, in the process of selecting a pipeline, we resorted to choosing the top-down pipeline based on the test results that we carried out on the other pipeline and based on developer recommendations. The pipeline top-down is characterized by the fact that the model that was created contains two files (centered and centroid), and these files contain a high parameter that improves the model

training process, including optimization, which contains batch size and epochs. Epochs of 200 and batch sizes of 40 were used. In addition to the augmentation that works to control noise, scaling, and brightness. Finally, the process of choosing an algorithm from among the many diverse algorithms such as Unet, Leap, Resnet, and Unet was used. After that, all changes are saved, and then we extract training job packages. zip to be used in the process of training the model on Google Colab. The third stage is that the model uses labelled suggestions in order to predict and evaluate the learning outcomes of the model. In addition, the unseen data can be used to evaluate the performance of the model and identify the new data. Also, the model can be retrained again or some of the main parameters can be changed in case of dissatisfaction with the results provided by the model. In the fourth stage, after completing the training and prediction and making sure of their correctness, the model is used in a simple flask app that work in ake any videos for horse from the laptop or PC and start the process of predicting or tracking the attached video, then it get a list of frames that predicted from the given video to convert it to a new video to be watched later. The fifth stage is relying on the trained model in order to predict the new data and extract the predicted coordinates, and then converting them into JSON files, followed by linking the behavior to each frame that was previously predicted, and saving this data in a CSV file. The sixth and final stage, in which the coordinates that were predicted in the previous stage are used in order to start classifying the behavior of horses. Figure 15 show the system overview

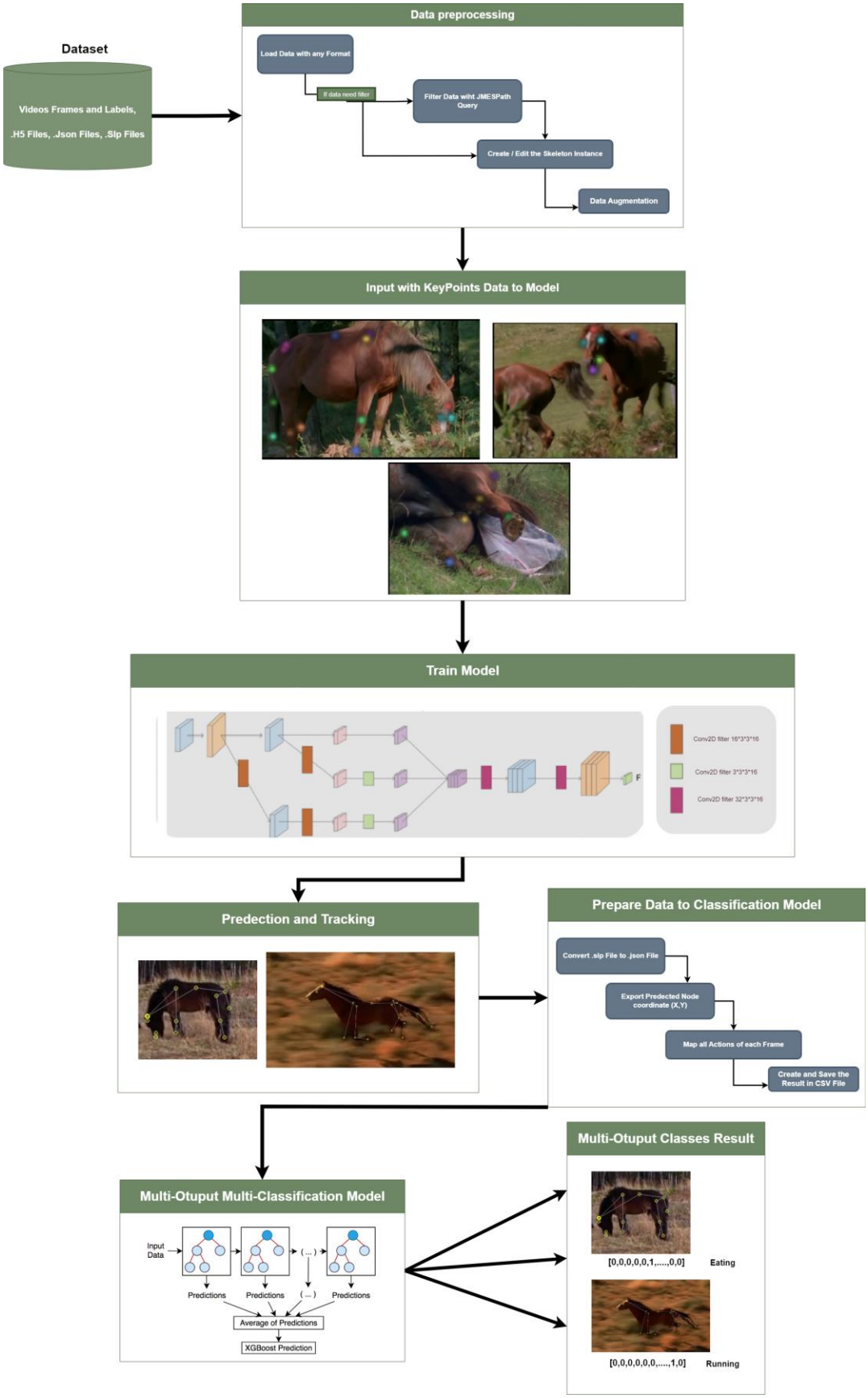


Figure 15 show the system overview

4.2.2 Convolution Layer

Conv2D is a 2D image processing conv layer of the TensorFlow Keras API. Performs a wrapping operation on the input image and generates the feature map as a result. Many deep learning models use the Conv2D layer for image classification, object recognition, and segmentation tasks. Conv2D is used as part of the deep learning pipeline in the SLEAP framework for multi-animal posture tracking. SLEAP tracks the positions of several animals in movies using convolutional neural networks (CNNs). To analyze the incoming video frames and predict the situation, the framework uses a combination of Conv2D layers and other types of layers. Conv2D is used in SLEAP to visually extract items of interest from animal posture data. It works with two-dimensional input data, such as photos or feature maps, and extracts feature using a set of learnable filters. Conv2D collects local patterns, edges, and textures by swiping these filters across the input data, which are critical for recognizing and tracking animal position landmarks or joints. SLEAP's Conv2D function provides flexibility and customization choices. The number and size of filters, stride length, padding choices, and activation functions are all configurable by the user. These settings can be adjusted to improve Conv2D's performance and accuracy for certain datasets and jobs. In the SLEAP structure, Conv2D is often followed by further layers such as pooling layers, normalization layers, and fully linked layers. The deep neural network's backbone is a hierarchical arrangement of layers that allows for precise position estimation and tracking. Overall, Conv2D is an important component of the SLEAP system, allowing the extraction of critical characteristics from input photos for analyzing and tracking animal behavior. SLEAP is a robust platform for behavioral research and comprehension because of its versatility and customization possibilities.

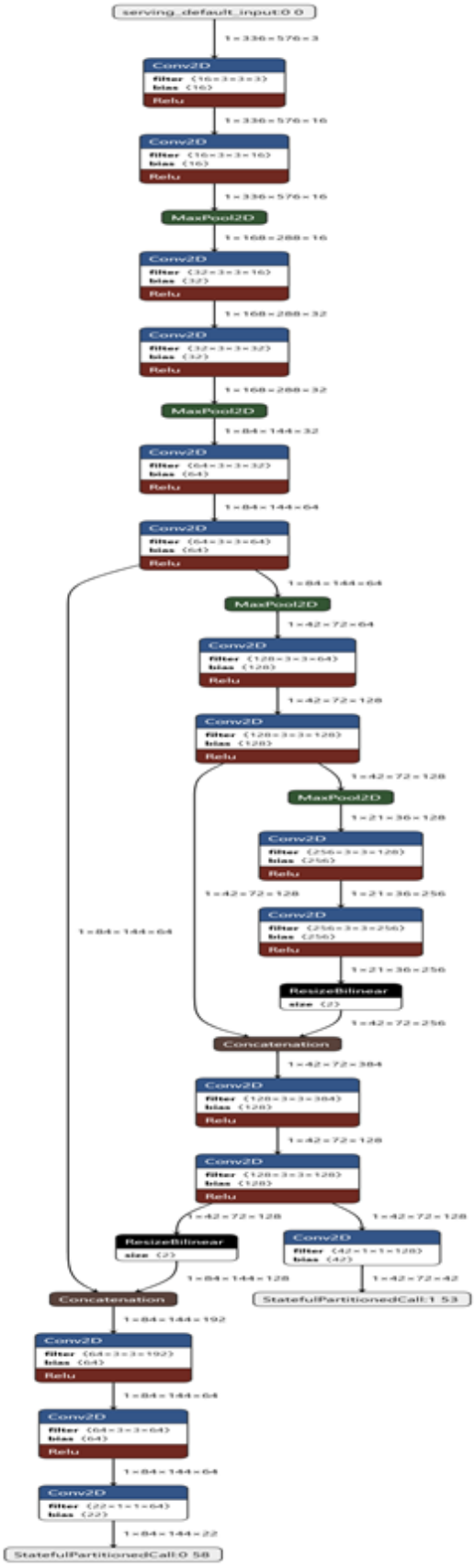


Figure 16 SLEAP pose estimation Conv 2D layer

4.3 System Running

In this part, we will show the input and output of every component in our final system.

4.3.1 Data Preprocessing

This process aims to extract data related to horses only to be used in the assessment process. This data consists of a large group of animals that we do not intend to use in estimating their conditions and tracking their movements. The inputs for this process are a JSON file that contains multiple categories and images of animals. The output from this process is a JSON file containing only horse-related data.

4.3.2 Build model

This process aims to create a model that tracks all the important parts of the horse's body, based on a previously built skeleton. Where the inputs of this process are the labeled video frame, showing the important parts of the horse's body. Depending on the parameter specified for the model, we initialize the labeled video frame by crop size, scaling, and brightness. In addition, max stride scans the image in order to focus on the important features in the image within the stride boundary. The output is a model that predicts and tracks horses.

4.3.3 Tracking instance

This process aims to work on tracking videos of horses by using the model that was created before, to show the instance on the horse's frame. Where the entries are videos and a model created before. The outputs are ".slp" files containing the frame on which the instance is shown.

4.3.4 Prepare data for horse behavior

The input to this step is a ".slp" file and contains prediction and tracking outputs. It is then converted into a JSON file in order to extract the x and y coordinates of the

points and then add the behavior of this frame. The output is a CSV file containing the x-axis and y-axis for each point, and the action is dependent on the frame.

4.3.5 Classification of horse behavior

This step aims to identify several behaviors that the horse performs. The input consists of the CSV file generated from the previous step. Null values are removed and replaced by -1 and then divided into 80% training data and 20% test data, then models are created, some of which are machine learning and the others deep learning. The outputs are the various behaviors of the horse that are identified.

Chapter 5: Results and Evaluation

5.1 Testing Methodology

In this part, we will explain and evaluate the different methodologies for model performance in the forecasting process that result from the use of a particular type of data set. We will clarify the methodology used for the pose estimation model for horses and the classification behavior XGBoost model for horses. In the beginning, the data used to estimate the situation for horses is divided into two types, namely testing and training. In the training process, the model is trained on a part of the data set at epoch = 200 and batch size = 4. As a result, the model is trained better, the data set is fed into a form, and then it is completed. The testing process on the other part of the data set is... Secondly, the data used to classify the horse's behavior is also divided into two types, as we mentioned in the previous methodology, but in the training approach process. For the model, the data fitting principle is used to ensure that the data is trained accurately and that the amount of loss is small in order to obtain better results. The next step is testing, and from here, a prediction of the fortress' behavior is made by using the other part of the data and some fitting processes.

5.2 Results

In the result section, we will show the different result cases, which is the best, acceptable, and worst results.

5.2.1 Best Results Cases

Firstly, the pose estimation model using the top-down pipeline generated the best result. This model is trained on an epoch of 200 and a batch size of 10. Additionally,

in the backbone section for adjusting the model, it had crop size =16 and max stride = 16. Figure 11 shows some random results for the best model. If you witness these results, it will be clear that the results are excellent because the majority of the skeleton on the horse is correct. For example, pictures 1 and 2 on Figure 17 show that most of the skeleton is in the correct position, due to the lighting, the environment in which the horse is located, and the direction of photography. Secondly, the best model of classification of horse behavior using the XGboost classifier and Gradient Boosting Regression, they classifies over 10 classes for horse actions like running, walking, eating, giving birth, hissing, etc. Additionally, the parameter used to train this classify model is max depth = 6 for XGboost and max depth = 3 for GBR , the objective was "multi: softmax" , and the number of classes is 145.

Picture (1) Picture (2)

Figure 17 Show Best model results

5.2.2 Acceptable Results Cases

To start, model a pose estimate using the same pipeline as a best-case scenario. This model was trained with batch size = 5 and epoch = 100. It also has max stride = 12

and crop size = 12 in the backbone section for model customization. depicts some random results of the accepted model. When you see these results, you will see that the model recognized some horses of large size but did not recognize the sizes of small and medium horses. It also recognized horses with dark colors and the proportion of the skeleton, some of which appeared in the correct place, while others did not appear in their correct places. Pictures (1) and (2) in Figure 18 show that most of the skeleton was in a correct and incorrect position, also the picture show the model work in dark horses and assessment of the position in an acceptable way, but not correct in a large proportion. Second of all, the acceptable model for classify horse behavior by using Multi label Perceptron and Random Forest classifiers. The parameter was activation='relu', and max_iter=300 for MLP while the Random forest parameter was number of classes = 145 and number of features = 46. This model show intermediate results.

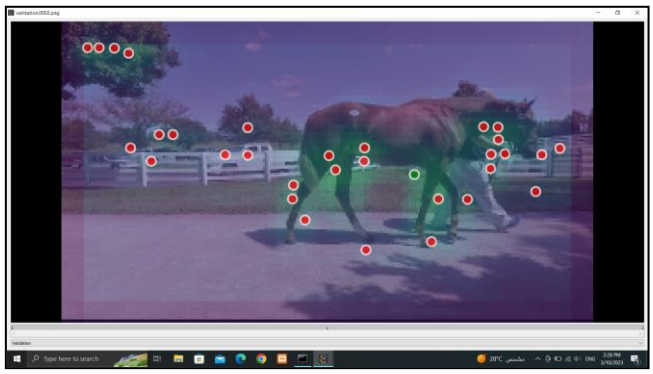
Picture (1)

Picture (2)

Figure 18 Show Acceptable model results

5.2.3 Worst Results Cases

Here is the model of pose estimation using the bottom-up pipeline. This model was trained with batch size = 2 and epoch = 80. It also has min scaling = 0.9 and max scaling = 1.1; in addition, there was a Gaussian noise mean of 5.0 in the section of model edit (data augmentation). Figure 13 displays some random results from the worst model. When you look at these results, you will see that the model cannot detect the majority of horse body parts. It also shows the wrong skeleton in the horse's body, like the nose appearing in the mouth and the shoulder appearing in the thigh of the horse. Pictures (1) and (2) in Figure 19 show that most of the skeleton was pose estimated in the wrong way. Second, the worst model of classification of horse behavior using the Supported Victor Machine Multiple and Logistic Regression classifier. This model get a bad result, because they did not recognize classes perfectly, and it make an overfitting problem.



Picture (1)

Picture (2)

Figure 19 Show Acceptable model results

5.2.4 Limitations

Many Limitations affect the functioning of the system. First, the amount of data available in the process of estimating the status of the horse and identifying its behavior. The more diverse the data and the more it contains different types of horses, the higher the quality of the desired results. Moreover, it may contain limited environments in which the horse was photographed, and this may affect the ability of the model to learn from the images, in addition to the distance of the camera from the horse and the angle of shooting. Secondly, excessive use of parameters during the model creation process may cause some problems when training this model as shown in the results of the previous different cases. Third, there is a hardware limitation, which is that during the process of creating the model, we may need to retrain the model because of the excessive use of the limited GPU and CPU. Fourth, in the process of classifying the horse's behavior, there are still problems with the multiplicity of actions that the horse performs that will be used later as the class for a classifier.

5.3 Evaluation

5.3.1 Accuracy

With regard to the part based on deep learning and neurons, the following table shows all the experiments that we conducted to obtain the best result. In addition, the table shows the method used to examine and evaluate the models individually. The best result of all these experiments achieved mean Average precision (mAP) = 0.6275335333156785, mean Average Recall (mAR) = 0.7566243194192376, and distance error (95%) = 7.9192803176613005.

Dataset	Approach	Backbone	mAP	mAR	Error (95%)	Error (50%)
Horse(10)	BU	ResNet	0.0012105926445087 18	0.012736490664028688	123.96117695021458	81.508243756032 56
Horse(10)	TD	UNet	0.1172715437788789	0.23963295587661068	8.25272935843475 4	1.41941460707125 3
Animal K.D	TD	UNet	0.6275335333156785	0.7566243194192376	7.91928031766130 05	2.24155007665779 36

Table 1 Show the SLEAP models Accuracy

For all of our experiments on behavioral classification using machine learning. These algorithms get their strength from the results predicted by the good model we created to identify and track important points in the horse's body. We achieved the best experience using XGBoost Multi-Output Classifier 84.7% using 145 class and achieved using 6 main class 91.68%.

Input Data	Model	Split size	Nu.Class	Nu.Features	Nu.estimators	Max_ Depth	Training Accuracy	Testing Accuracy	Another distribution of data	Problem
CSV	SVM	0.2	1	46	-	-	99.52%	99.52%		Over Fitting
CSV	Random Forest	0.2	145	46	100	-	99.86%	83.35%	8.2%	Need More Data
CSV	MLP	0.2	145	46	1000	-	75.21%	62.93%	6.75%	Need More Data
CSV	MOC XGB	0.2	145	46	200	6	99.57%	84.7%	10.8%	Need More Data
CSV	MOR GBR	0.2	145	46	1000	3	98.92%	92.11%	78.79%	Need More Data
CSV	MOC XGB	0.2	6	46	200	6	99.76%	91.68%	36.93%	Need More Data

Table 2 Show the behavior models Accuracy

5.3.2 Time performance

Since SLEAP depends mainly on the power of the GPU, so we went to work on Google Colab. Our main goal in this transformation was to speed up the work of creating the model and evaluating the results. Where the process of creating the model on the personal computer that contains 4 GB GPU and 8 GB, CPU took several hours and the process was done slowly at the rate of epoch every five minutes. While in the Google Colab, which gives you 12 GB CPU and 15 GB GPU, the process is faster, as it takes 31s/epoch and 125ms/step. In addition, it allows you to increase the packages used, which increases the speed of the training process.

Chapter 6: Conclusion and Future Work

6.1 Conclusion

To summarize, a system was created in this thesis to pose estimation and classify different horse behaviors. After researching several approaches to determine the most advantageous approach, the decision was made to pursue the deep learning approach, computer vision, neural networks, and other algorithms in order to construct a successful system that is added as an appropriate system for posing estimate horses and classifying behavior horses. First, we estimated the position of the horses, where we used the SLEAP framework and strategy was a top-down strategy (pipeline) for this task because, after the pose estimation process, the horse's behavior will be classified. As for the algorithms that were used for this process, they are deep learning and machine learning, specifically UNET and Resent. They are known algorithms for the tasks of estimating the status of animals. This algorithm proved its ability to produce great results for estimating the position of the horse. Secondly, we come to the task of classifying the behavior of horses, which depends on the previous task, as we mentioned. In this process, we used the XGBoost model, and this model proved tremendous results, as it recognized more than 10 classes, which are different behaviors of horses. In different environments. Finally, the aim of these operations is the ability to track and distinguish the daily activities of the horse's life.

6.2 Problem Issues

6.2.1 Technical issues:

Firstly, training the model on Google Colab needs Run type (GPU). Sometimes the training process of the model is interrupted, and this is due to excessive use of the GPU. We solved this problem by training the model in stages so that the GPU does not occur. Secondly, twice the capabilities of the personal device, such as the CPU

8GB and the GPU 4GB, and course these are somewhat acceptable specifications for dealing with the SLEAP framework. The problem was that the CPU space becomes full during the training process, as this affects the response speed of the device, which may cause a malfunction in the model training process. To solve this problem, we used Google Colab. To conclude, while the flask app running in the Google Colab notebook it does not give us the permission to run another cell of code.

6.2.2 Scientific issues:

In the beginning, before the process of training the model, an adjustment was made in the calling inside augmentation. If it was supplied, the training process would slow down, and the points on the horse would appear randomly and not in their correct places, due to the excessive enlargement of the image. Secondly, using the crop size excessively may cut off important parts of the image, which also leads to a wrong training process and incorrect results. Finally, when the max stride increases, this leads to greater receptivity at the expense of increasing the number of trainable parameters in the model and slows down the training of the model.

6.2 Future work

We hope in the near future to add many things that will make this project more accurate and realistic. The previously mentioned models must be developed to reach good results. Like the model that relies on the RESNET algorithm, we seek to use other updates from the RESNET, for example, RESNET101 and RESNET152. We also seek to improve pipelines such as single animal and other bottom-up and use them in dealing with big data to achieve good results. In addition, do not forget that the amount of data and annotations related to the positions and classifications of

horses must be increased. It is considered a good improvement because it allows dealing with different breeds of horses and the diverse environments in which they exist. Real-time Pose Estimation can also be added, which allows results to be provided at the same time, which can be used later by veterinarians or during horse races. Finally yet importantly, work on developing an easy and smooth user interface to make it easier for users to access and analyze results.

References

[1] Cao, J., Tang, H., Fang, H.-S., Shen, X., Lu, C., & Tai, Y.-W. (2019). Cross-Domain Adaptation for Animal Pose Estimation. Cornell University - ArXiv. https://doi.org/10.48550/arxiv.1908.05806.

[2] Dai X, Li S, Zhao Q, Yang H. Animal Pose Estimation Based on 3D Priors. Applied Sciences. 2023; 13(3):1466. https://doi.org/10.3390/app13031466.

[3] Rashid, M., Broomé, S., Ask, K., Hernlund, E., Andersen, P.H., Kjellström, H., & Lee, Y.J. (2021). Equine Pain Behavior Classification via Self-Supervised Disentangled Pose Representation. 2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV), 152-162.

[4] Johnson, S., & Everingham, M. (2015). Clustered Pose and Nonlinear Appearance Models for Human Pose Estimation. British Machine Vision Conference.

[5] Li, Chen & Lee, Gim. (2021). From Synthetic to Real: Unsupervised Domain Adaptation for Animal Pose Estimation. 1482-1491. 10.1109/CVPR46437.2021.00153.

[6] Mathis, A., Biasi, T., Schneider, S., Yuksekgonul, M., Rogers, B., Bethge, M., & Mathis, M. W. (2021). Pretraining boosts out-of-domain robustness for pose estimation. In Proceedings of the IEEE/CVF winter conference on applications of computer vision (pp. 1859-1868).

[7] Pereira, T. D., Shaevitz, J. W., & Murthy, M. (2020). Quantifying behavior to understand the brain. Nature neuroscience, 23(12), 1537–1549. https://doi.org/10.1038/s41593-020-00734-z.

[8] Pereira, T.D., Tabris, N., Li, J., Ravindranath, S., Papadoyannis, E.S., Wang, Z.Y., Turner, D., McKenzie-Smith, G.C., Kocher, S.D., Falkner, A.L., Shaevitz, J.W., & Murthy, M. (2020). SLEAP: Multi-animal pose tracking. bioRxiv.

[9] Pereira, T.D., Tabris, N., Matsliah, A. *et al.* SLEAP: A deep learning system for multi-animal pose tracking. *Nat Methods* **19**, 486–495 (2022). https://doi.org/10.1038/s41592-022-01426-1.

[10] Perez, M., & Toler-Franklin, C. (2023). CNN-Based Action Recognition and Pose Estimation for Classifying Animal Behavior from Videos: A Survey. arXiv preprint arXiv:2301.06187.

[11] Söderström, M. (2021). Pose Classification of Horse Behavior in Video : A deep learning approach for classifying equine poses based on 2D keypoints (Dissertation). Retrieved from http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-304512.

[12] Karashchuk, P., Rupp, K. L., Dickinson, E. S., Walling-Bell, S., Sanders, E., Azim, E., Brunton, B. W., & Tuthill, J. C. (2021). Anipose: A toolkit for robust markerless 3D pose estimation. Cell reports, 36(13), 109730. https://doi.org/10.1016/j.celrep.2021.109730.