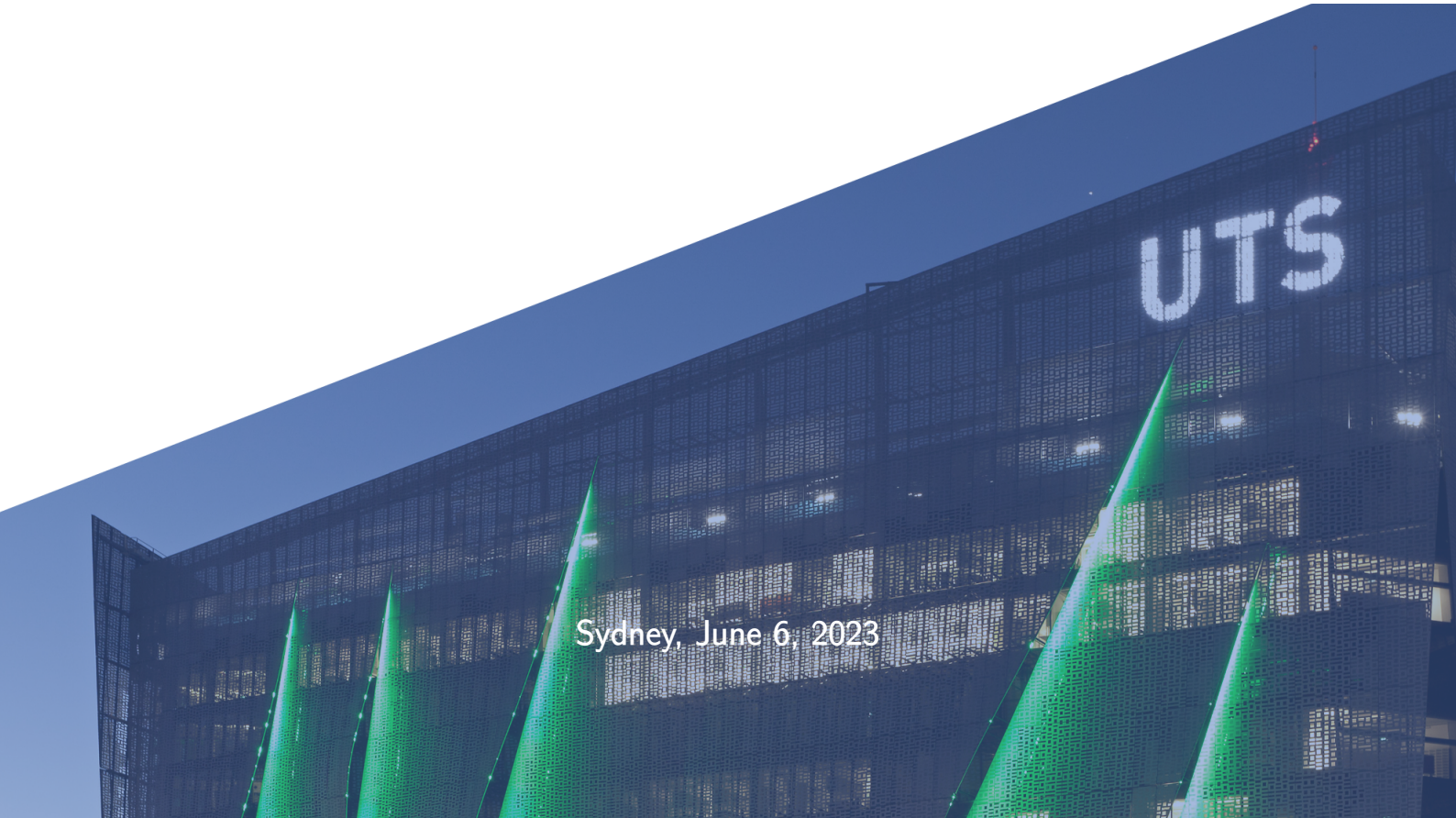
Project development
Injury prone pose detection

42028: Deep Learning and Convolutional Neural Networks (Autumn 2023)

Group 40 - EURO 2023

Full Name	Student ID
Jonathan Lys	24854604
Javier Sabin Ulloa	24651073
Joan Velja	24652164



Sydney, June 6, 2023

Contents

Abstract

This report aims to describe the implementation of a deep learning model tailor made for the detection of potentially injury prone poses in workplaces. The application takes as an input media files, be them videos or pictures, and outputs a score that represents what we call the *goodness of pose*, i.e., how close to a correct pose the given one is, along with additional information on how to avoid stressful postures for the person being analyzed. We propose as suitable industries for such application to be those in charge of warehouses, offices and so on. We believe the benefits generated by the adoption of such application would be two-fold: both for the employers and the overall welfare of the company.

Keywords

Deep learning, Pose estimation, Workplace safety, Prevention

1 | Introduction

Bad postures in a work environment can lead to various health issues and discomfort for employees. Some potential issues associated with bad postures include:

- **Musculoskeletal Disorders (MSDs)**: Prolonged poor posture can contribute to the development of MSDs such as back pain, neck pain, shoulder pain, and repetitive strain injuries.
- **Fatigue and Reduced Productivity**: Maintaining uncomfortable postures for extended periods can lead to muscle fatigue, decreased energy levels, and reduced overall productivity.
- **Increased Risk of Injury**: Poor postures can increase the risk of acute injuries, such as strains, sprains, and falls, as the body's balance and stability are compromised.
- **Joint and Nerve Compression**: Incorrect postures can cause compression on joints, nerves, and blood vessels, leading to conditions like carpal tunnel syndrome, thoracic outlet syndrome, and sciatica.
- **Reduced Respiratory Function**: Certain postures, especially those that involve slouching or hunching over, can restrict proper lung expansion, leading to shallow breathing and decreased oxygen intake.

Some examples of bad postures commonly observed in work environments include:

- Slouching or rounded shoulders
- Forward head posture (protruding neck)
- Sitting cross-legged or with legs crossed for prolonged periods
- Leaning forward or hunching over a desk
- Awkward wrist or hand positions during computer use
- Twisting or bending the spine while lifting heavy objects
- Standing or walking with poor alignment or weight distribution

It is important for employers and employees to be aware of these potential issues and take steps to promote good posture and ergonomic practices in the workplace. What we propose in this dissertation is a solution for employers to detect and tackle these potentially harmful positions, addressing the problem at its core without the insurgence of injuries, thus benefiting both the employer (who does not have to grant paid leave for injured employees and will also enjoy a higher productivity) and the employee (who will be able to correct himself while carrying out a specific task).

1.1 | Application

What we suggest for this application is a web-based service which allows the user to upload media files (videos or images) and obtain as an output the pose estimation through the rendering of a stick figure, along with a measure we engineered, called *goodness of pose*. This metric basically provides a score of how good a pose is, given a previous detection of the activity being carried out (i.e., standing, sitting, lifting weights, and so on). The backbone of the application will be explained into further detail in later stages of the report.

1.2 | Dataset

The models implemented for the task are multiple: Detectron2 and VideoPose3D are pre-trained and thus did not require training. Our proprietary model instead, the pose classifier, has been trained with a self-made dataset of 3D keypoints extracted from videos of us carrying out various activities.
The dataset includes 16 videos of good poses and 16 videos of bad poses from multiple perspectives, and their associated 3D keypoints, as numpy arrays. The shape of these arrays is $(N, 17, 3)$ with N being the number of frames, 17, the number of keypoints and the 3 dimensions.

Three scenarios were selected in accordance with the main poses leading to injury in the workplace, particularly those with relevant health risks: walking, lifting a box and sitting, as indicated in this Harvard article.

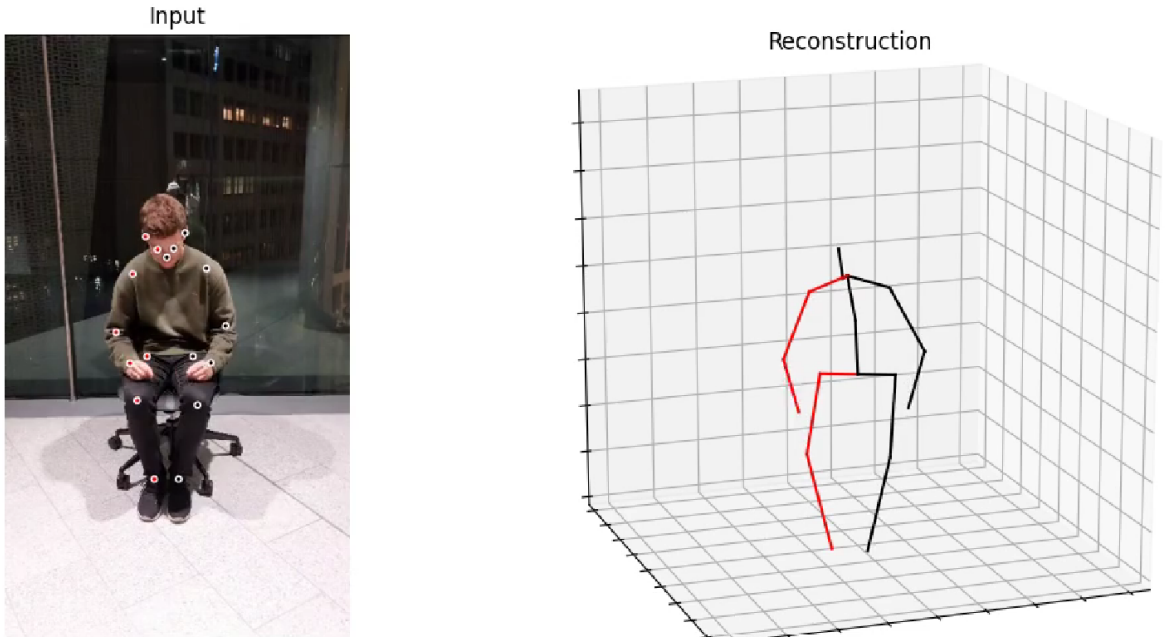
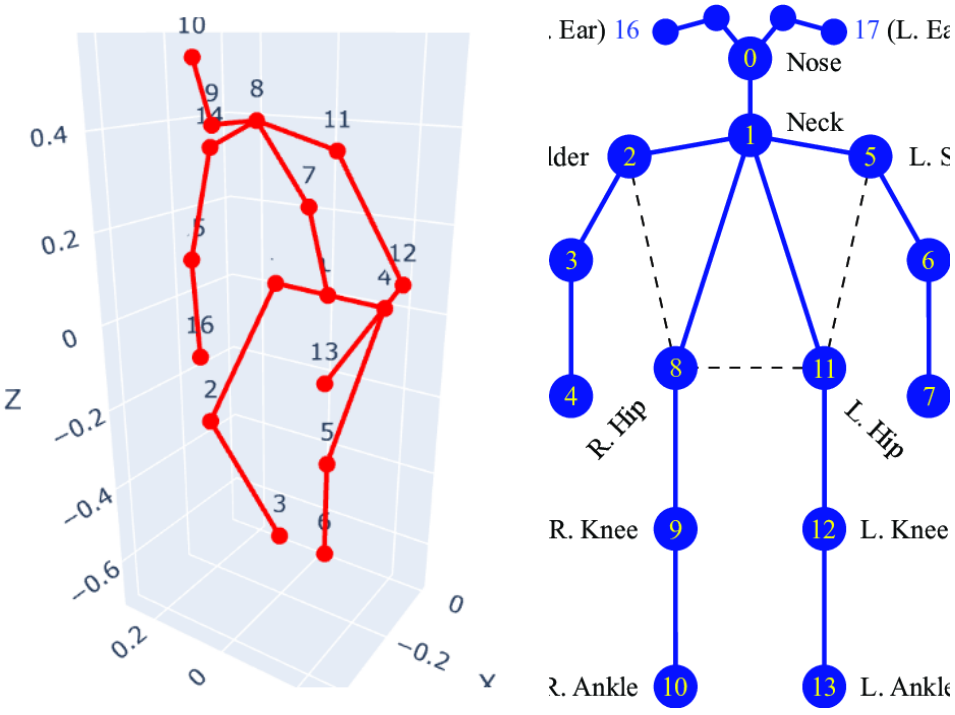


Figure 1.1: Input Video with 2D keypoints (left), 3D reconstruction (right)

The first image summarises the 2 necessary files/instances that are required to infer the 3D keypoints. It includes the raw input video with the addition of the predicted 2D keypoints.

It was also noted that the keypoints and their order were not consistent with the COCO keypoints format.



(a) 3D reconstruction of the previous input video **(b)** COCO keypoint format

Figure 1.2: Different keypoint formats

2 | Overview of the architecture

Our architecture is composed of three different neural networks.

2.1 | Flow Diagram/Flowchart/Work Flow

The model takes a video file as an input. The video has to contain one subject that is always fully visible. Then, 2D keypoints are extracted with a Detectron2 model. Then, those keypoints are passed through the VideoPose 3D model that extracts the 3D keypoints and generates a 3D reconstruction of the subject. This reconstruction is also added to the original video for comparison, this makes the first output of this workflow. The 3D keypoints are then utilised to generate the relevant angles used to assess the correctness of the pose.

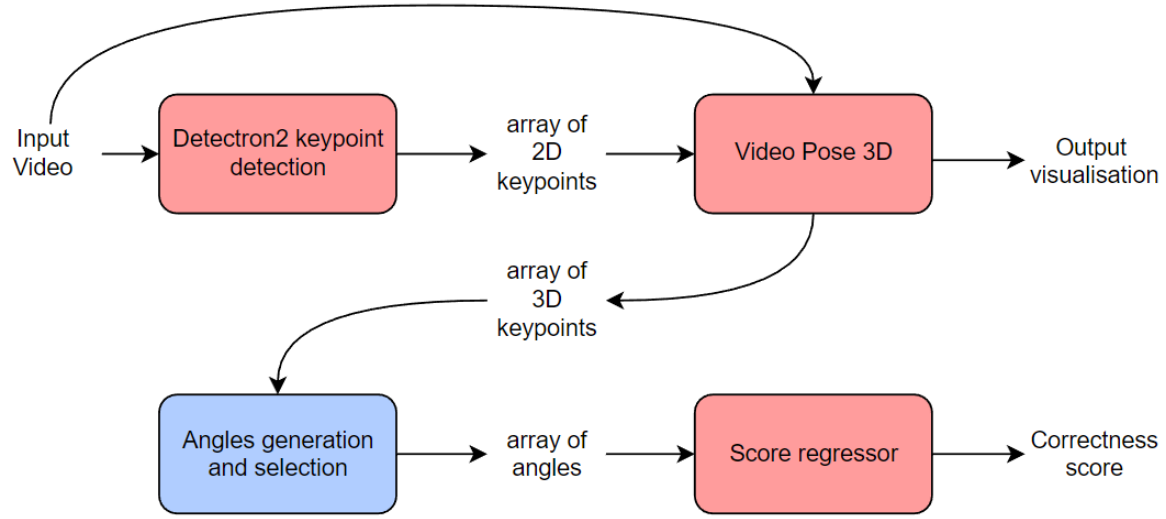


Figure 2.1: Complete flow diagram

2.2 | CNN/MLP Architecture Design

Once we had chosen to use the VideoPose3D model, we had two choices for the 2D keypoint extraction, the Detectron models and the Detectron2 models. For spatial precision purposes, we chose to work with the Detectron2 models. All the Detectron2 models are based on Faster R-CNN [1]. Looking at the Detectron2 model zoo, there are 4 models available for keypoints detection.

Name	Train time	Inference time	box AP	keypoint AP
R50-FPN	0.315	0.072	53.6	64.0
R50-FPN	0.316	0.066	55.4	65.5
R101-FPN	0.390	0.076	56.4	66.1
X101-FPN	0.738	0.121	57.3	66.0

Table 2.1: Metrics of the Detectron2 keypoint detection models

The inference time correspond to the inference on a single frame with a GPU. Given that we could run the models on non GPU devices, the inference time was not considered, as CPU inferences on a few seconds video took several minutes. The selected model is the X101-FPN (pretrained), because it has the best performance, and the overall goal was to get a good spatial precision. The backbone of this model is the ResNetXt101, introduced in [2]. The model combines an object detection task to identify the person and a keypoint detection task (17x2 regressions) to infer the 2D keypoints.
The VideoPose3D model uses a receptive field of 243 (3^5) frames in our case. It uses a series of 5 3x1 residual blocks. An example is given for a receptive field of 27 (3^3, using 3 residual blocks) frames. It should be noted that it is also a pretrained model, as 3D annotations are out of our reach.

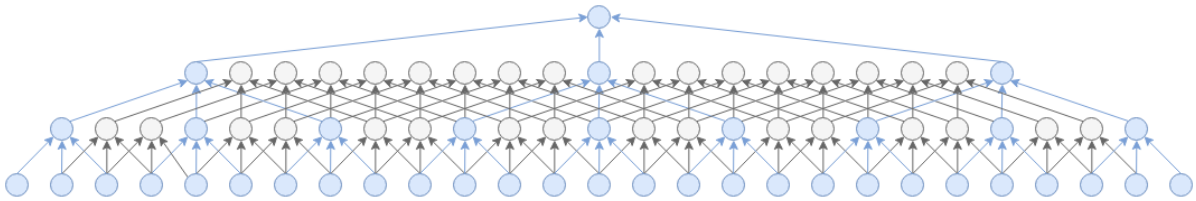


Figure 2.2: VideoPose 3D architecture - 27 frames receptive field

It might seem like a lot of connection are useless because they are not considered into the final inference, but according to the documentation: "For inference of very long sequences, this approach is very efficient as intermediate results are shared among successive frames." It should be noted that the convolution blocks use both past and future information to interpolate the present position, preventing the model to run live.
Finally, the MLP design was mostly the selection of the relevant features and feature engineering related to those features. Three features were created: Angles, Distance Ratios and Symmetry. For each frame, the following aspects were collected:

- Exact positions of the keypoints.
- Angles: 8 angles potentially leading to the aforementioned injury-prone poses were used attempting to cover a full spectrum of relevant angles:
  - Head and neck alignment (to derive a forward head posture). Shoulders and neck alignment (to derive rounded shoulders).
  - Left hip and knee alignment (for overall knee alignment).
  - Right hip and knee alignment (for overall knee alignment).
  - Hip, medullary cone and neck (for spinal alignment).
  - Hip, medullary cone and head (for overall forward standing position).
  - Right shoulder, elbow and wrist (for overall arm alignment).
  - Left shoulder, elbow and wrist (for overall arm alignment).
- Symmetry: Both the mean of all left joints (shoulder, elbow, wrist, hip, knee and ankle) and right joints was calculated. Then, the left joints mean was subtracted by the right joints mean, obtaining a simple measure of symmetry.
- Distance ratios: the distance between left and right shoulder was divided by the distance between left and right hip, obtaining a simple ratio that may be indicative of imbalances or misalignment in the upper body.

Positions perform great, but are probably biased by the training data i.e. the body orientation. Since angles on their own are almost perfectly predictive of pose correctness, they are selected due to lower proneness to bias with fewer features.
With an Occam's razor strategy in mind, different configurations of angles were attempted to simplify the model and analyze the most relevant combination of angles for top performance. A combination of four angles was found to yield 100% accuracy as well.
The selected angles are the neck/nose/head angle, neck/shoulders angle, hip/medullary cone/neck (the back) angle, and the hip/knees angle. The model is a simple MLP with 2 hidden layers of width 64 and 32, and ReLU activations.

In summary, our three models are the pretrained X101-FPN (ResNetXt101 backbone), the 3,3,3,3,3-VideoPose3D model (243 frames receptive field), and the MLP classifier. All the models were build in PyTorch, which allows the use of PyTorch optimised environment.

2.3 | GUI Design

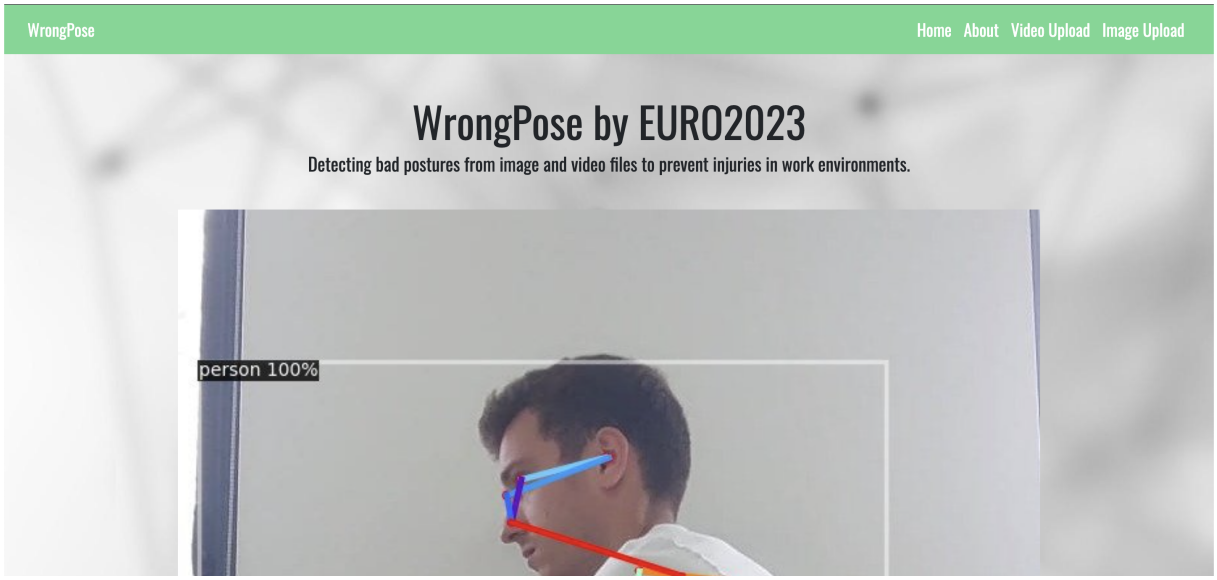


Figure 2.3: Homepage

The GUI is still WIP. The main functionalities are there already and have been implemented through the Wagtail CRM. The application loads at the above page, where a carousel of samples is presented, along with other information:

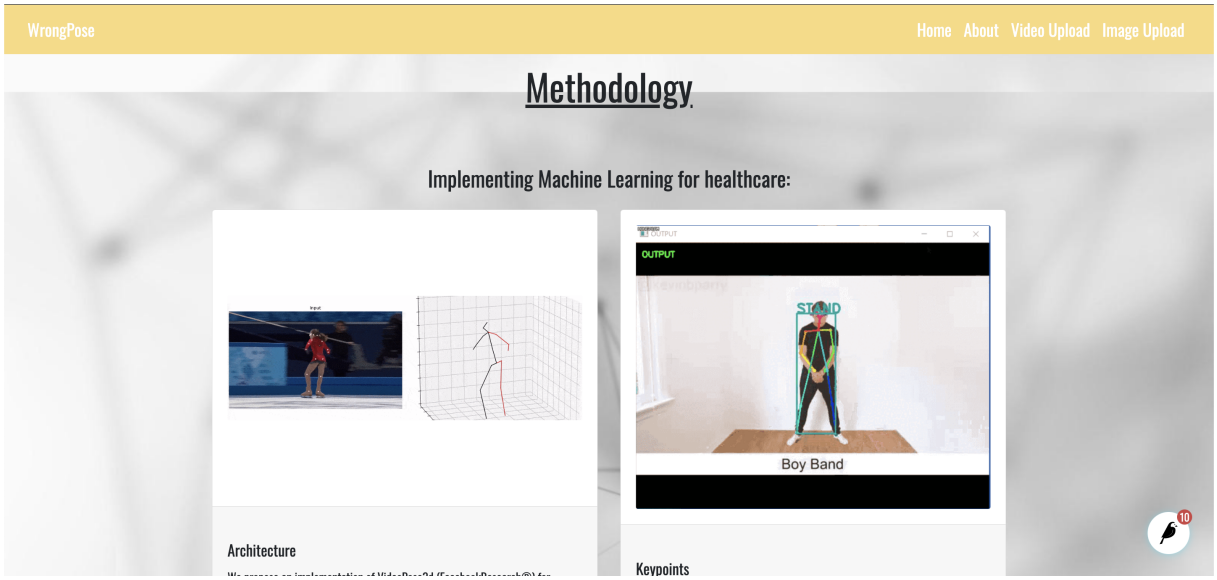
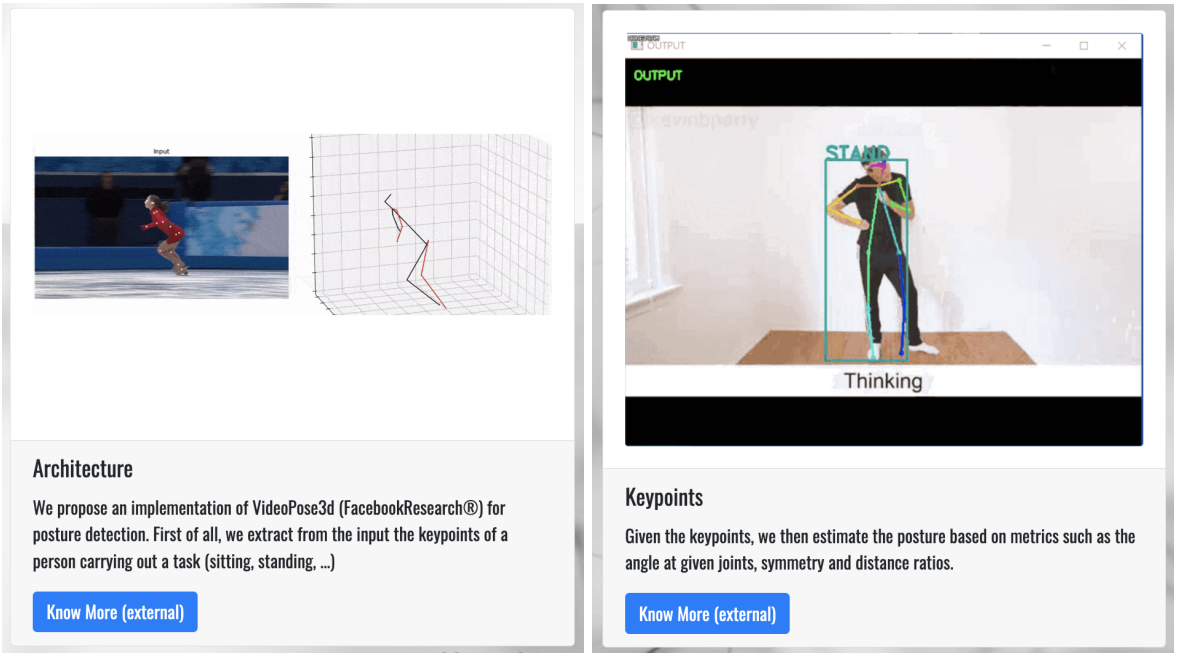


Figure 2.4: Methodology



(a) Architecture card **(b)** Keypoint explanation card

The methodology section explains briefly the functionality: 2.5a has a button redirecting to the VideoPose3d repository, whereas 2.5b addresses the keypoints used for the evaluation of the pose. The button links to this report.

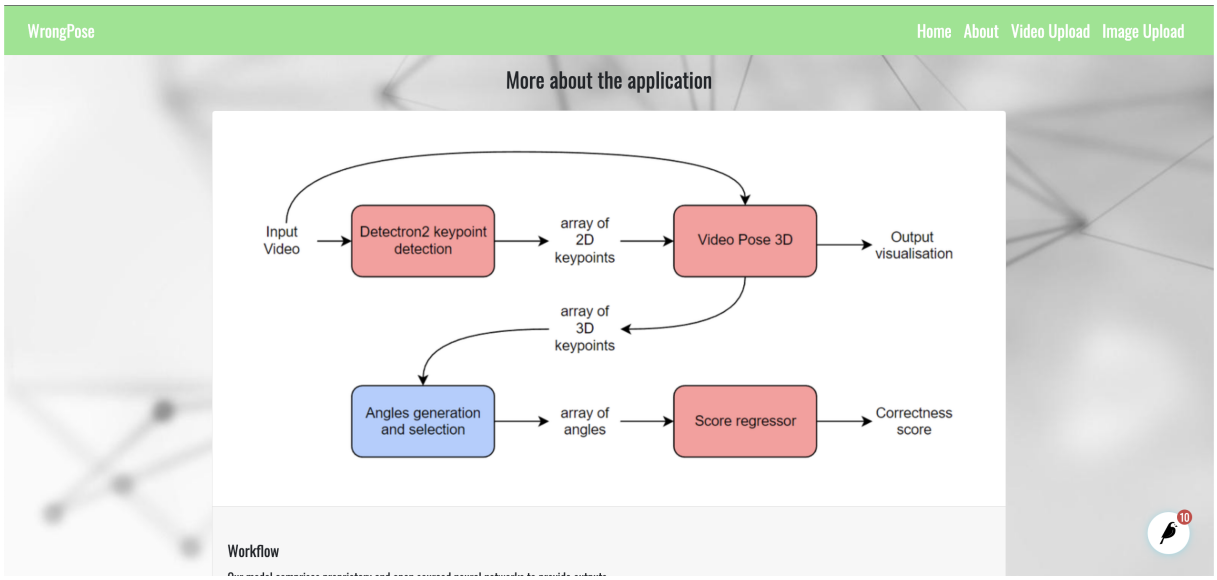


Figure 2.6: Workflow

2.6 instead addresses the brief workflow provided in this report.

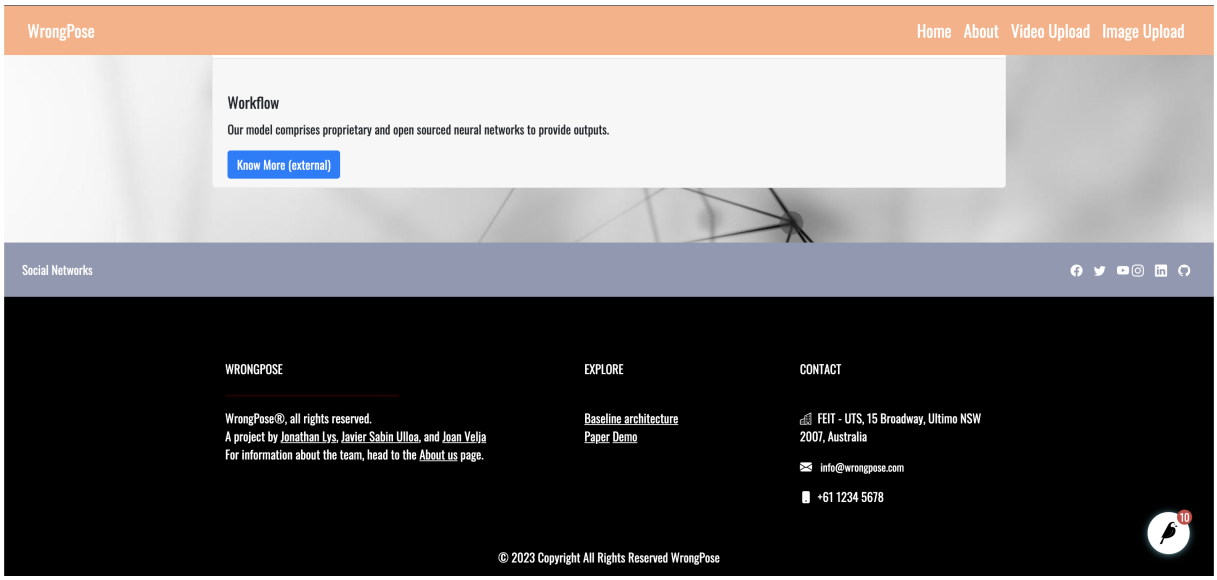


Figure 2.7: Footer

Finally, 2.7 contains social media links, general information about us, with links to our LinkedIn pages, this report, a demonstration video of the app and contacts.

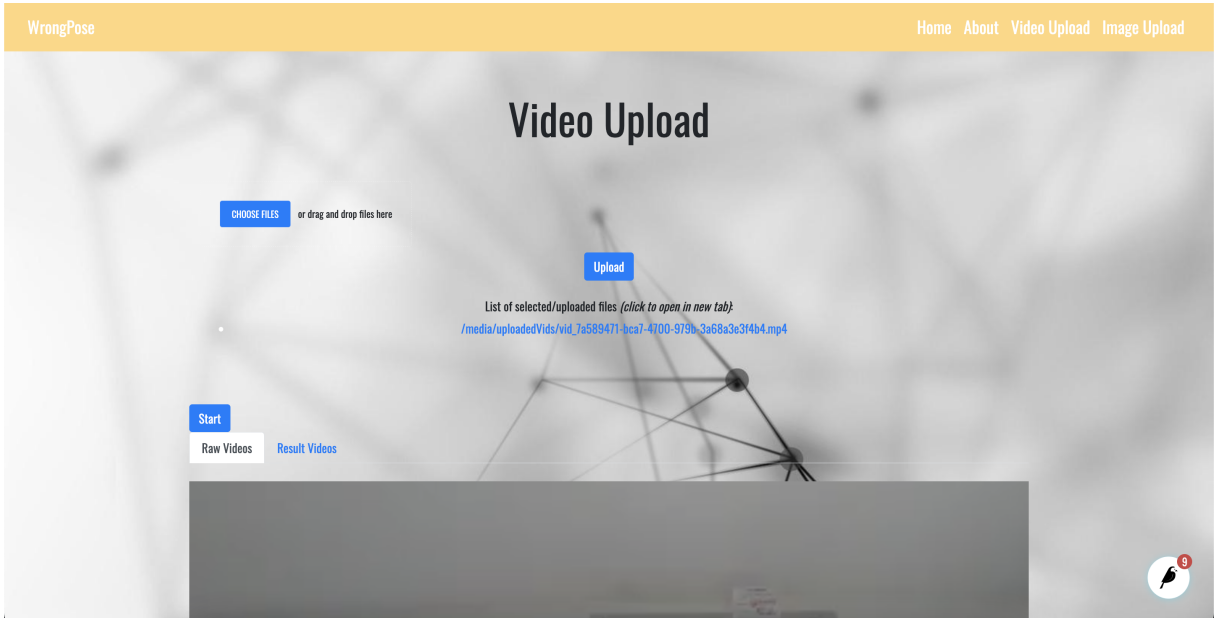


Figure 2.8: Video Upload

2.8 is the heart of our web-application: here the end users can upload the video they want to be processed through our model. Again, this is a rudimentary page which is currently under modification to add further functionalities, but the pipeline is the following:

- The user chooses the file (or multiple files) clicking on the button *"Choose files"*;
- Then, by clicking *"Upload"*, the videos get uploaded to the backbone application directory;
- The uploaded videos can be played in the box at the bottom, under the tab *"Raw Videos"*: multiple submissions will result in a carousel of media files processed sequentially;
- By clicking *"Start"*, the model receives the uploaded file(s) and carries out the inference.

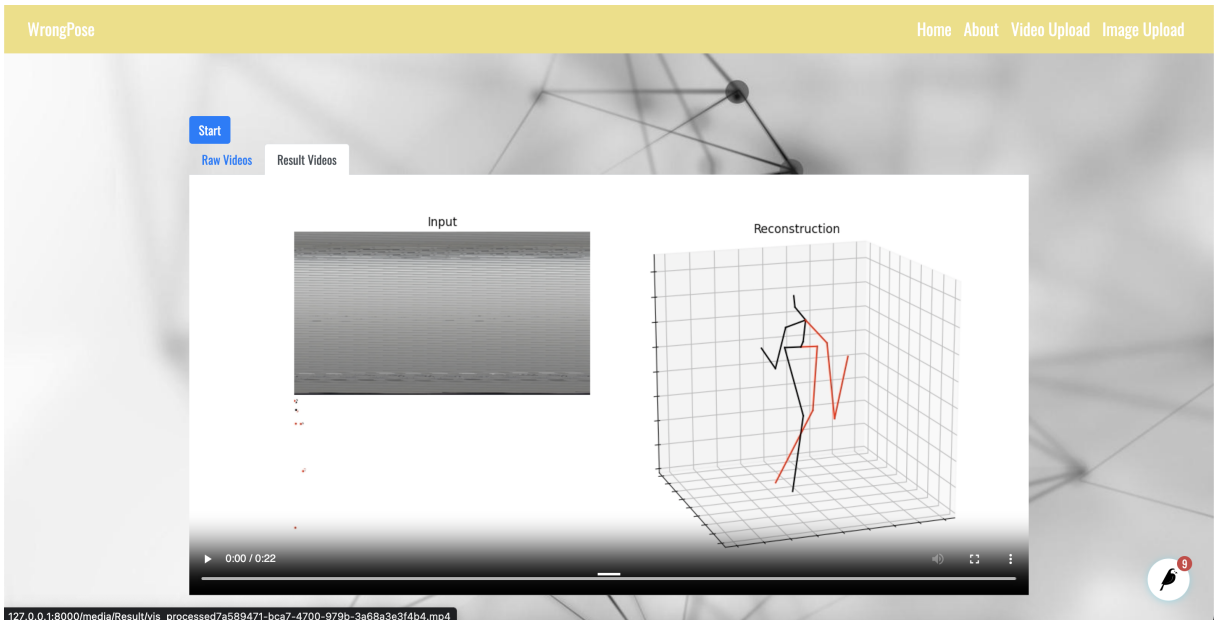


Figure 2.9: Video outputs

The tab *"Result Videos"* contains the outputs as 2.9 shows. The output is a keypointed video.

Figure 2.10: Video outputs - scores

2.10 shows a framewise score graph of the uploaded video: the frames are the relevant ones with respect to the video uploaded (the dummy video is very short, about 2s, the selected frames are the middle ones), where the pose is good if scored above 0.5 and bad if below.

We exempt from this report the discussion of the *"About Us"* page, as it only contains cards of us 3, and the *"Image Upload"* page as its content is identical to the one of *"Video Upload"*, but just meant to process image files.

3 | Results and Evaluation

We discuss the experimental results.

3.1 | Experimental Settings

The experiments were conducted on the arrays that represent the reconstructed 3D keypoints of the reference dataset. The goal was to separate the poses labelled "Good" and "Bad".

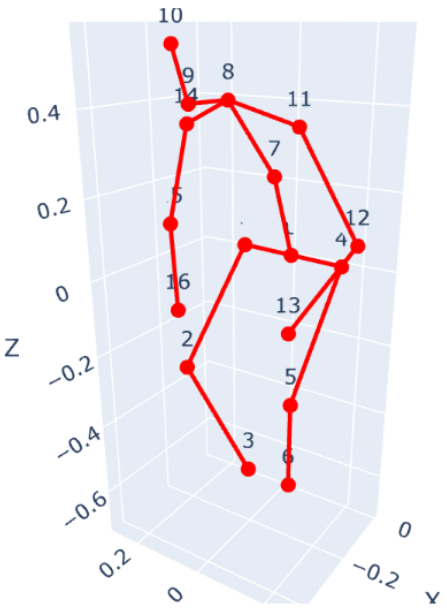


Figure 3.1: 3D reconstruction of a reference pose - label "Bad"

3.2 | Pre-Processing

When the features included the consideration of the raw coordinates of the keypoints, the frames were augmented via a rotation around the z axis by multiplying each frame by this rotation matrix:

$$R_z(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

No scaling augmentation was required because the scale was consistent between the frames. Experiments show that the standard deviation of the average length of the femur (over the whole dataset, i.e. more than 45k frames) was less than a centimetre.

For the training strategy of the last model, no data preprocessing was applied because measuring the angles already ensures the model's robustness to any kind of rotation and scaling.
However, the workflow including several intermediary arrays of processed data, it should be noted that before applying the VideoPose3D, a basic frame interpolation strategy is applied to the 2D output on frames that are missing a detected person.

3.3 | Experimental Results

From the feature engineering, several MLP models were trained and evaluated, to determine the best features for training:

Name	Number of features	Val. accuracy	Test accuracy
All features	61	100%	100%
Positions	51	100%	100%
Angles	8	99.89%	99.96%
Positions and angles	59	100%	100%
Symmetry	1	71.35%	71.64%
Distance ratios	1	53.51%	55.07%

Table 3.1: Features comparison

After having selected the 4 relevant angles, the final MLP was fully trained. We report the loss curve of the training stage.

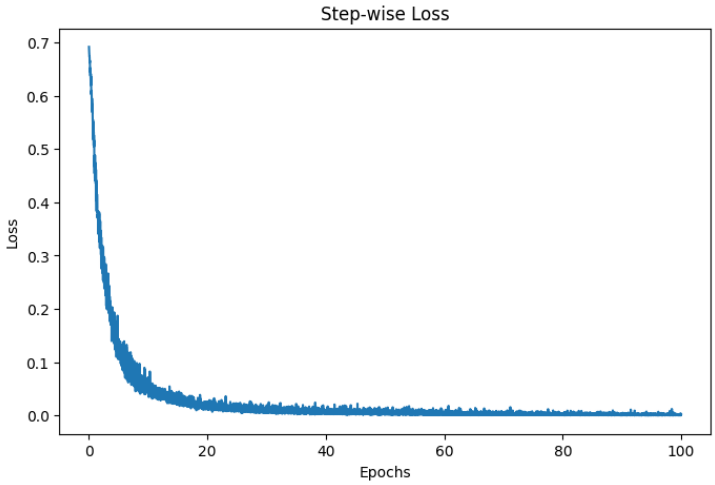


Figure 3.2: MLP training loss

3.4 | Limitations

The most important limitation is the inference time of the models, the most important one being the Detectron2 keypoint detection model, especially on CPU. Experiments show that inference on small videos (few seconds long) can take up to 15 minutes.
Other limitations concern the training data. The small variety of poses that we considered covers most of the poses encountered in a work environment, but the model could still have difficulties identifying poses that are very far from the ones in the dataset.

4 | Discussion and Conclusions

4.1 | Project summary

This project leverage a combination of 2 pretrained CNN models (Detectron2 and VideoPose 3D) to extract 3D keypoints from videos. Then, a *correctness score* is inferred from the 3D keypoints, to assess if a given pose is "Good" or "Bad". Our model reach a nearly perfect classification score of 99.96% on our reference dataset with a selection of 4 relevant angles.

4.2 | Future improvement

The goal of future improvements would be to implement a live version of our algorithms, even if this means decreasing the spatial precision of the 3D keypoints. The main issue that prevents our current implementation from running live is the inference time of the keypoint detection model.
We looked at other models that could speed up this process. **YOLOv7** and **Meidapipe's Blazepose** pose detection models are both faster than Detctron2, but less accurate. Blazepose is even optimised to run on a CPU at 30fps and includes a tracking functionality that allows the model to reduce jitter by introducing frame consistency, which can be applied in our case because we work with videos.
However, the VideoPose3D was specifically trained to work with the Detectron predictions, so using other models could reduce the overall spatial accuracy enough to miscalculate the 3D keypoints.
It should nonetheless be noted that the VideoPose3D model includes a parameter called *causal* that makes the model use only causal convolution instead of symmetric convolutions, meaning that the model would infer keypoints only using data from the past, making it usable in real time applications. This also implies a slightly increased error.
To allow the model to cover more types of poses, we would add diversity to the dataset. By adding more poses to the reference dataset, we would be able to cover more situations. We also assume that this could lead to poses that are harder to assess and that would need more inputs than the 4 angles in our current implementation. To mitigate this issue, we suggest adding a fourth model, that would first classify the task, to determine what the last model of the pipeline would be, given that we would have trained a variety of models that are specialised on specific poses, and also select the relevant features to feed to this model.

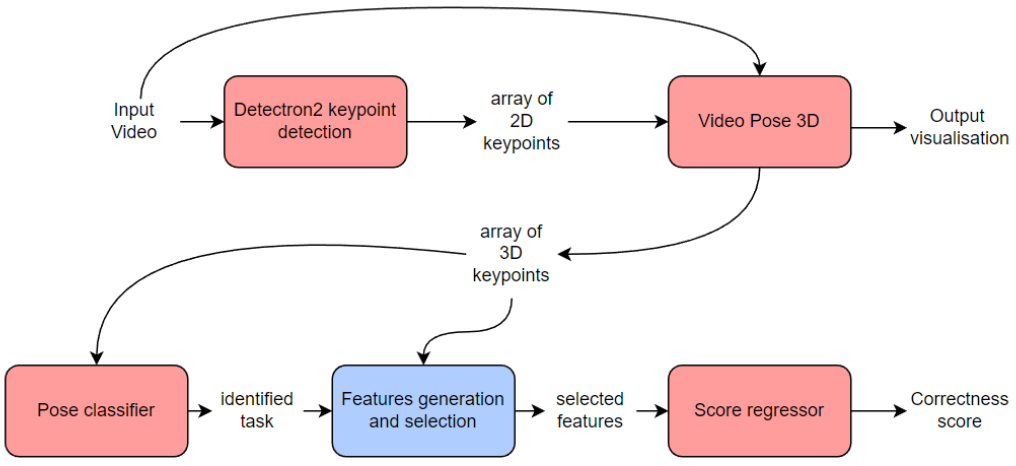


Figure 4.1: Suggested improved workflow

5 | References

[1] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster r-cnn: Towards real-time object detection with region proposal networks, 2016.

[2] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks, 2016.

A | Appendices - Contribution

A.1 | Individual Contribution

- **Jonathan Lys:** Script and GitHub repository combining Detectron2, VideoPose 3D and the MLP, keypoints dataset generation
- **Javier Sabin Ulloa:** Video dataset generation, feature engineering, MLP design, training and evaluation
- **Joan Velja:** Website design and implementation, integration of the models to the website

A.2 | Individual Contribution Split

Team member name	Percentage
Jonathan Lys	33.3%
Javier Sabin Ulloa	33.3%
Joan Velja	33.3%

All the team members have discussed and agreed on the above individual contribution to the project.

Jonathan Lys : Jonathan Lys
Javier Sabin Ulloa : Javier Sabin Ulloa
Joan Velja : Joan Velja