MP Survey Completion Evidence

Thank you for completing MP feedback – Matric no: 2004368C

ENGMP <engmp@tp.edu.sg>
To TAN YUAN YANG RICHIE

Reply | Reply All | Forward

Wed 25/1/2023 8:49 am

If there are problems with how this message is displayed, click here to view it in a web browser.
Click here to download pictures. To help protect your privacy, Outlook prevented automatic download of some pictures in this message.

Dear student,

Thank you for completing the MP Student survey.

Please include the Response ID below in your MP Report, and save this email for future reference.

Response ID: 96555024

Best Regards ENG MP Admin Support

Response ID: 96555024

Thank you for completing MP feedback – Matric no: 2000578A

ENGMP <engmp@tp.edu.sg>

Wed 1/25/2023 9:09 AM

To: KHAIRUL FAUZAN B NORISHAM <2000578A@student.tp.edu.sg>

Dear student,

Thank you for completing the MP Student survey.

Please include the Response ID below in your MP Report, and save this email for future reference.

Response ID: 96555543

Best Regards ENG MP Admin Support

Response ID: 96555543

E

ENGMP <engmp@tp.edu.sg>

To: LUONG HAO WEN

Thu 1/26/2023 10:03 PM

Dear student,

Thank you for completing the MP Student survey.

Please include the Response ID below in your MP Report, and save this email for future reference.

Response ID: 96665301

Best Regards ENG MP Admin Support

Response ID: 96665301

Matric No.

2000897E

Student Name (as on matriculation card)

LUONG HAO WEN

Sleep Posture Classification by Deep Learning

Luong Hao Wen (2000897E)

Khairul Fauzan Bin Norisham (2000578A)

Tan Yuan Yang Richie (2004368C)

Supervisor: Mr Kok Peng Chow

Abstract

This project aimed to design and implement a Machine Learning model that is capable of detecting, predicting, classifying and displaying the sleeping postures of the patient. The model is able to differentiate from the three sleeping postures; Left, right and Supine. OpenCV was used to activate the camera which enables Computer Vision to detect images and perform pose classification in real time. Mediapipe is used to create landmarks which show the vital points of the body of the patient. With the objectives of the project being successfully achieved, the Pose Classification model has proven to be accurate and confident in classifying sleeping poses.

1. INTRODUCTION

1.1 Background

A bedsore is a lesion caused by unrelieved pressure that results in damage to the underlying tissue. [1] Patients with bedsores are most likely not able to easily correct their sleeping posture by themselves, causing them to feel discomfort. The best way to stop bed sores is by detecting the sleep posture so that the nurses can help the patient move.

Thus, in order to improve the unobtrusiveness and comfortableness of sleep posture monitoring, the proposed solution is to design and implement a machine learning (ML) posture detection, which does not require specialized equipment or ML expertise to achieve accurate predictions.

1.2 Objective

This project aimed to develop a machine learning model that can analyze the sleeping posture by capturing an image of the body and providing the statistic of data. The model can differentiate from the three sleeping postures; Left, Right and Supine.

The specific objectives of the project were to

- Enable the communication between the camera software and the machine learning model for capturing the images using Python programming.
- Design the machine model for OpenCV/Mediapipe/Python ScikitLearn/TensorFlow
- Integrate the user interface, the machine learning model, and camera software using Python libraries and the database MYSQL

2. PROJECT DESCRIPTION

2.1 Working of the Project

In this system, the user can connect a camera to view the sleep posture and the prediction of the image model. It is a lightweight versatile solution to detect and record sleep posture in real time from a continuous video or static image.

Figure 1 Image capture from the Image model

2.2 Image model

This model was designed and programmed using OpenCV and Mediapipe. After the user runs the model, the application will connect to the user's web camera. Next, the model will try to detect a patient in a sleeping position. Once it detects a person, the application will detect key features of the patient called landmarks. These key features include the eyes, mouth, arm, shoulder, back, and legs. Once all the key features are collected, the application will try to classify the position of the person based on the landmarks and display the prediction result on the image.

Figure 2 Landmarks for detection

The user also has the option to record the data and store it in the database. If the user presses the spacebar, the model will record and store the images from the camera in a database.

2.3 Pressure Sensor Model

Here there is also another option available. The user can upload pressure sensor data obtained from the pressure mat using a laptop. From the sensor data, the system will detect and classify the sleep posture.

The prediction result will then be displayed to the user via a website. The user must upload the data collected by the pressure sensor mat. The website has two functions:

1. Classify sleeping position using pressure sensor data

2. Display the prediction result from the pressure sensor model

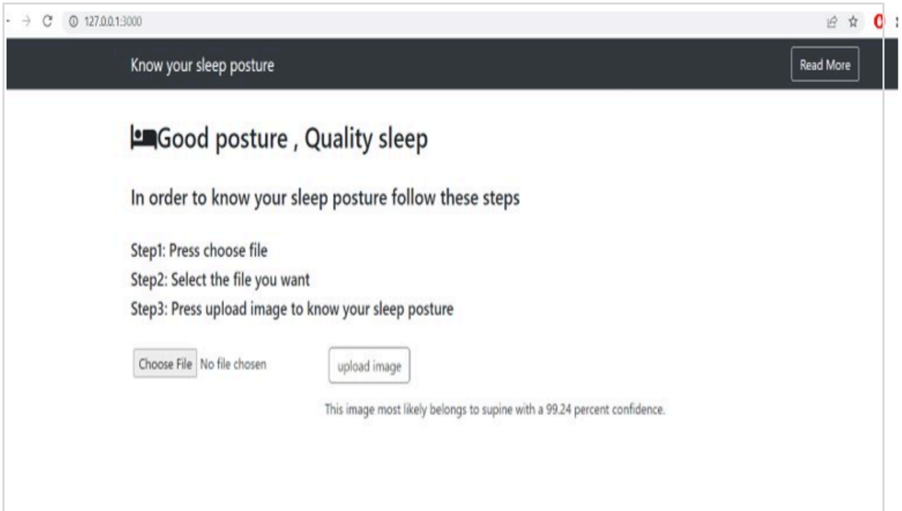


Figure 3 Landing page of website

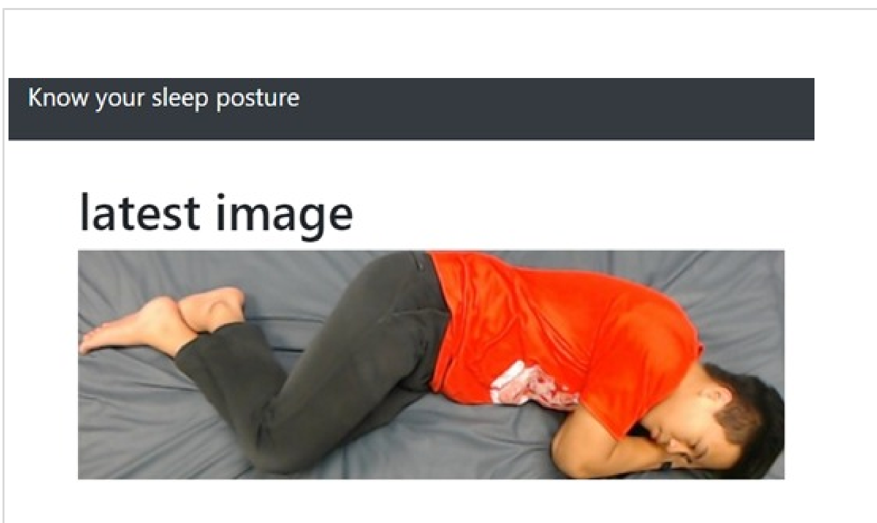


Figure 4 Live display of website

3. PROJECT DEVELOPMENT

3.1 Research on existing models

The first step was to research machine learning models used for sleep posture classification.

The scope of research was identifying models that can perform medical classification using images.

The models that were searched for were models that could take unprocessed pressure mattress reading as input and can perform conversion and classification as output. These models include 1D CNN, 2D CNN, and RNN.

Other models that did not take in unprocessed pressure mattresses were also looked at. Any model that did not need pressure mat reading could be used as a secondary way of classification.

From the research done, three models have been identified that were able to classify sleep posture.

3.2 Environment setup and packages installed

Google Collaboratory ("Colab") is used as the Integrated Development Environment (IDE) to write and execute python code for machine learning. The installed machine-learning packages are TensorFlow, Keras, and NumPy. Jupyter Lab is also used as an IDE.

3.3 Developing a 2-Dimensional Convolution Neural Network (2D CNN)

The first step of Implementing 2D CNN is to perform image processing. Since the dataset needs to be processed into an image first before it can be used for training. The second step is to store all the data processed in a directory. In this case, the data is stored in google drive.

Next, the image is split into two different folders called to train and test. 80% of the data is put into the train folder while 20% of the data is put into the test folder. After this, the images in the train folder are passed through an augmentor to generate more images. Finally, the 2D CNN Model will be defined. Once done, it will then be compiled and trained. This will result in an accuracy score being printed which is then evaluated, as well as tested with an input image to test for reliability.

3.4 Developing a Support Vector Machine (SVM) Model

The data is loaded from google drive. There are three folders in the dataset named test, testing, and predicted, with each folder containing images that will be used to train the model. Linear Support Vector Classification is used as the model. The program uses a Histogram of Oriented Gradients, also known as HOG, to extract features from the image data by focusing on the shape of the object to compute its features.

The image is then resized using CV2 and converted into an array so that the computer can understand the features of the image in numeric data. The training images are then loaded into X Train and the test images are loaded into X Test while label encoding is used for the Y Train and Y Test.

3.5 Developing an Image Model

Firstly, research was done to examine how the Mediapipe and OpenCV Pose Classification model works and how they can be implemented into our project. Once that is done, Jupyter

Notebook is required for installing the necessary dependencies and libraries, as Google Collaboratory is unable to install Mediapipe.

Next, the development of the code for the model was made. It makes use of Mediapipe to create nodes and landmark points, which will detect the body and detect and classify the poses. In addition, OpenCV will help provide visuals to show the real-time webcam feed and display the image data and results of the pose detection and classification.

Finally, after the coding was developed, initial testing was done with images that displayed the poses to ensure the accuracy of the model. Once it is established that it can do so with confidence, final testing using the webcam for real-time feed was conducted.

3.6 Data Collection + Data cleaning

First, 12 different subjects were identified. These subjects were then asked to lie down on an airbed. A camera is fixed from the top view so that a clear image could be taken of the subject. The subject is then asked to make different sleeping positions while the camera is taking images of their sleeping position.

Subjects were encouraged to move more so that more sleeping positions could be recorded.

A total of 1264 images have been collected from 12 subjects. However, the images are too big to fit inside google colab. As such, the images were scaled down as well as cropped to around the size of 284 * 100. Unimportant features such as the floor and surrounding were cropped away.

3.7 Developing a website

A website was developed using Flask as the web framework. The website was designed using Hyper Text Markup Language (HTML), JavaScript, CSS language.

Additional features were also experimented with such as getting "live" data and Amazon Web Service Deployment. These features were successfully prototyped but were difficult to implement without the physical pressure mattress to test.

3.8 Final selection

Having come to an agreement, a prototype was chosen for final implementation. The OpenCV and Mediapipe model was the best model to be used for final implementation as it was the only model that did not rely on pressure mattress data to perform accurate classification.

3.9 Additional features

Additional features were added to the OpenCV and Mediapipe model such as database connection. A MySQL database was created and structured. Most of the ideas shelved were not suitable for implementation. However, certain ideas of the shelved features could still be reused.

One of these ideas was the “live” recording. Instead of getting the “live” recording of the camera, the images and the time of recording could be collected and stored in a database. Then the database could be queried to get the most recent recording added and displayed on a website. This is to simulate a “live” recording.

3.10 Integrated Testing

The testing was done by positioning a camera to capture a person lying on the floor/bed in a sleeping posture. It is mounted on a tripod to stabilize the camera. A USB cable is used to connect to the laptop which runs the image model. The person will change sleeping posture every 2 minutes. The model will then predict the sleep posture.

Once it does the prediction, the prediction will be verified with the actual sleeping posture and the image model prediction result.

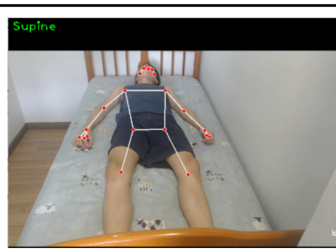
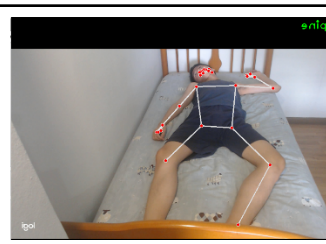
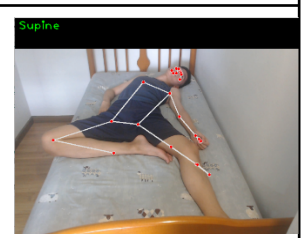
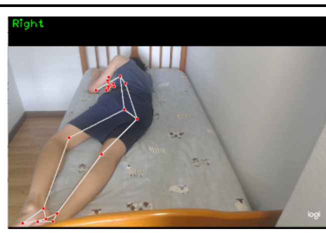
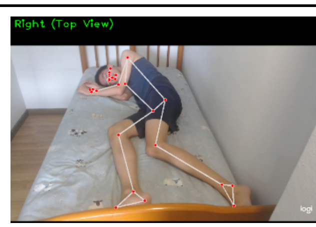
The hardware requirement for testing is a Logitech C922 Pro HD Stream Webcam and an ASUS laptop windows 11, I5 processor, 16 GB RAM, and 500GB Hard Disk.

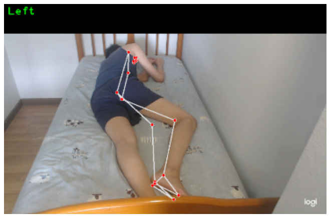
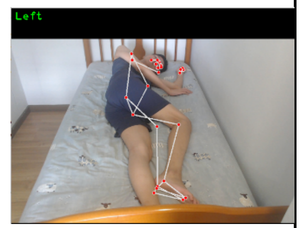
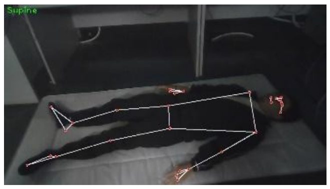
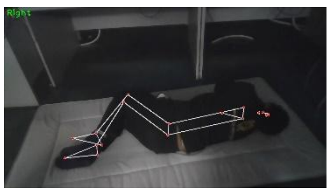
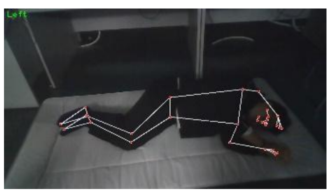
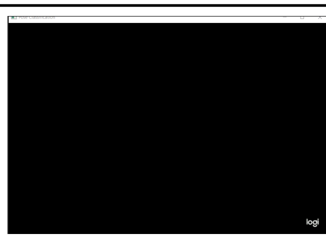
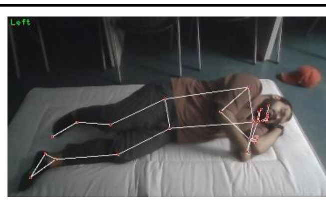
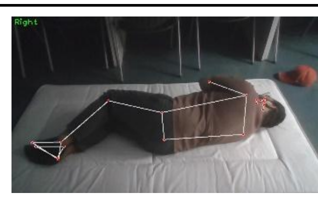
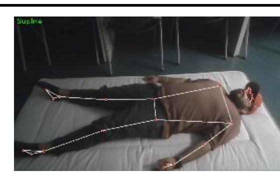
3.11 Testing Record

The test cases were recorded with the results shown in a table.

S/N	Test case	Result	Remark	Reference
1	The model should be able to classify the position and store the images it collects in a database	Pass	The image was record with the timestamp in the database	Refer to Table 1
2	The website was able to display the “live” image	Pass	The most recent image was displayed in the web browser correctly.	Refer to Table 1
3	The model was able to perform classification under partial darkness	Fail	The accuracy of model predictions is affected.	Refer to Table 1
4	The model was able to perform classification under full darkness	Fail	The model failed to detect the person and no prediction result	Refer to Table 1
5	The model was able to perform classification at a different angle of the camera	Pass	The model was able to perform predictions at side view and top view	Refer to Table 1
6	The model was able to identify foreign objects	Fail	The model failed to identfy the human	Refer to Table 1

Figure 5 Tests checklist

<table>
<tr><th>Test Case</th><th>Description</th><th colspan="3">Reference</th></tr>
<tr><td>1</td><td>Record from database</td><td>
<table>
<tr><td>Edit</td><td>Copy</td><td>Delete</td><td>31</td><td>[BLOB - 64.0 KiB]</td><td>TheImage_31.jpg</td><td>2023-01-27 12:08:43</td><td>Left</td></tr>
<tr><td>Edit</td><td>Copy</td><td>Delete</td><td>32</td><td>[BLOB - 64.0 KiB]</td><td>TheImage_32.jpg</td><td>2023-01-27 12:08:46</td><td>Left</td></tr>
<tr><td>Edit</td><td>Copy</td><td>Delete</td><td>33</td><td>[BLOB - 64.0 KiB]</td><td>TheImage_33.jpg</td><td>2023-01-27 12:08:53</td><td>Left</td></tr>
<tr><td>Edit</td><td>Copy</td><td>Delete</td><td>34</td><td>[BLOB - 64.0 KiB]</td><td>TheImage_34.jpg</td><td>2023-01-27 12:08:57</td><td>Left</td></tr>
<tr><td>Edit</td><td>Copy</td><td>Delete</td><td>35</td><td>[BLOB - 64.0 KiB]</td><td>TheImage_35.jpg</td><td>2023-01-27 12:09:09</td><td>Right</td></tr>
<tr><td>Edit</td><td>Copy</td><td>Delete</td><td>36</td><td>[BLOB - 64.0 KiB]</td><td>TheImage_36.jpg</td><td>2023-01-27 12:09:16</td><td>Right</td></tr>
<tr><td>Edit</td><td>Copy</td><td>Delete</td><td>37</td><td>[BLOB - 64.0 KiB]</td><td>TheImage_37.jpg</td><td>2023-01-27 12:09:18</td><td>Right</td></tr>
<tr><td>Edit</td><td>Copy</td><td>Delete</td><td>38</td><td>[BLOB - 64.0 KiB]</td><td>TheImage_38.jpg</td><td>2023-01-27 12:09:19</td><td>Right</td></tr>
<tr><td>Edit</td><td>Copy</td><td>Delete</td><td>39</td><td>[BLOB - 64.0 KiB]</td><td>TheImage_39.jpg</td><td>2023-01-27 12:09:20</td><td>Right</td></tr>
<tr><td>Edit</td><td>Copy</td><td>Delete</td><td>40</td><td>[BLOB - 64.0 KiB]</td><td>TheImage_40.jpg</td><td>2023-01-27 12:09:24</td><td>Right</td></tr>
<tr><td>Edit</td><td>Copy</td><td>Delete</td><td>41</td><td>[BLOB - 64.0 KiB]</td><td>TheImage_41.jpg</td><td>2023-01-27 12:09:42</td><td>Unknown Pose</td></tr>
<tr><td>Edit</td><td>Copy</td><td>Delete</td><td>42</td><td>[BLOB - 64.0 KiB]</td><td>TheImage_42.jpg</td><td>2023-01-27 12:10:22</td><td>Unknown Pose</td></tr>
<tr><td>Edit</td><td>Copy</td><td>Delete</td><td>43</td><td>[BLOB - 64.0 KiB]</td><td>TheImage_43.jpg</td><td>2023-01-27 12:10:33</td><td>Right</td></tr>
<tr><td>Edit</td><td>Copy</td><td>Delete</td><td>44</td><td>[BLOB - 64.0 KiB]</td><td>TheImage_44.jpg</td><td>2023-01-27 12:10:48</td><td>Right</td></tr>
<tr><td>Edit</td><td>Copy</td><td>Delete</td><td>45</td><td>[BLOB - 64.0 KiB]</td><td>TheImage_45.jpg</td><td>2023-01-27 12:10:49</td><td>Right</td></tr>
<tr><td>Edit</td><td>Copy</td><td>Delete</td><td>46</td><td>[BLOB - 64.0 KiB]</td><td>TheImage_46.jpg</td><td>2023-01-27 12:10:50</td><td>Unknown Pose</td></tr>
<tr><td>Edit</td><td>Copy</td><td>Delete</td><td>47</td><td>[BLOB - 64.0 KiB]</td><td>TheImage_47.jpg</td><td>2023-01-27 12:11:43</td><td>Supine</td></tr>
<tr><td>Edit</td><td>Copy</td><td>Delete</td><td>48</td><td>[BLOB - 64.0 KiB]</td><td>TheImage_48.jpg</td><td>2023-01-27 12:11:45</td><td>Supine</td></tr>
<tr><td>Edit</td><td>Copy</td><td>Delete</td><td>49</td><td>[BLOB - 64.0 KiB]</td><td>TheImage_49.jpg</td><td>2023-01-27 12:11:47</td><td>Supine</td></tr>
<tr><td>Edit</td><td>Copy</td><td>Delete</td><td>50</td><td>[BLOB - 64.0 KiB]</td><td>TheImage_50.jpg</td><td>2023-01-27 12:11:48</td><td>Supine</td></tr>
</table>
Console</td><td>NA</td><td>NA</td></tr>
<tr><td rowspan="2">2</td><td>Supine</td><td>
</td><td></td><td>
</td></tr>
<tr><td>Right</td><td>
</td><td>
</td><td>
</td></tr>
</table>

	Left			
3	Supine			NA
	Right			NA
	Left			NA
4	full darkness		NA	NA
5	Left view			

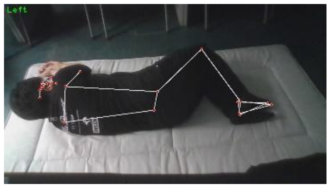
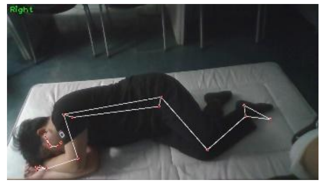
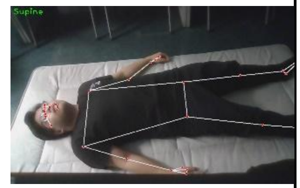
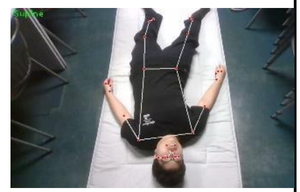
	Right view			
	Top view			
6	Foreign object		NA	NA

Figure 6 Tests photo

4. RESULTS

4.1 Findings

The model was functional but had some limitations that were not considered. The OpenCV and Mediapipe model was able to classify sleeping positions most of the time. However, the classification could not predict accurately under suboptimal conditions.

The camera we were given was not high quality and was not sensitive to light. This will cause the OpenCV and Mediapipe model to give a wrong prediction as it could not detect the patient's landmark. A camera with better detection of light will be preferred when using OpenCV and Mediapipe models.

4.2 Limitations

There were some conditions where the classification was given the wrong prediction result. For example, the lighting in the room affected the results of the classification and the model had difficulties identifying objects that look like a human. Next, the database will populate very fast when doing a “live” prediction which may result in a database crash.

5. CONCLUSION

Regarding the Image Model’s reliability, the person’s face must be present to detect sleep posture. The model works best when the person is visible in the frame, but it also detects a partial body.

The model built using OpenCV and Mediapipe is satisfactory in the classification of sleeping postures.

If more time was given, more research work could have been done to tune the model detecting the presence of a human face. The data insertion process will be improved to ensure the reliability of the database. Furthermore, the user interface display for the visualization and analysis of the statistical data using the Flask web framework can also be further developed.

In addition, there were further setbacks to the project. The researcher relayed the message that the pressure sensor mattress could not be used to collect data as it was still in the process of being prototyped. The spare pressure mat she provided for data collection was also found faulty. The aim of the project had to shift from classifying sensor data to image data. The CNN model and SVM model are then modified to use an image instead of pressure sensor data. This change in objective had a great impact on time management and disrupted the project workflow.

Acknowledgments

The authors wish to thank Mr Kok Peng Chow (Major Project Supervisor) for his patience and help, the researcher, Ms Rugminiamma Aryalekshmi, for helping us give feedback and suggestions to progress with the project, and people who have helped volunteer to be test subjects for the data collection.

References

[1] Mayo Clinic, “Bedsores (pressure ulcers),” Apr. 19, 2022. Accessed on: Oct. 17, 2022. [Online]. Available: https://www.mayoclinic.org/diseases-conditions/bed-sores/symptoms-causes/syc-20355893

Other Sources

PhysioNet, “A pressure map dataset for in-bed posture classification,” Sep. 28, 2017. Accessed on: Oct. 20, 2022. [Online]. Available: https://physionet.org/content/pmd/1.0.0/

L. Bainbridge, “How does a pressure Mattress Work?,” Beaucare Medical Ltd, Jun. 21, 2018. Accessed on: Oct. 20, 2022. [Online]. Available: https://www.beaucare.com/insights/pressure-mattress-work#:~:text=What%20is%20a%20Pressure%20Mattress,mounting%20stress%20on%20pressure%20points.

Vishal, "Insert / Retrieve file and images as a Blob in MySQL using Python," PYnative, Mar. 09, 2021. Accessed on: Dec. 20, 2022. [Online]. Available: https://pynative.com/python-mysql-blob-insert-retrieve-file-image-as-a-blob-in-mysql/

Bleed AI, "Real-Time 3D Pose Detection & Pose Classification | Mediapipe | OpenCV| Python," Youtube. Aug. 3, 2021. Accessed on: Dec. 01, 2022. [Video File]. Available: https://www.youtube.com/watch?v=aySurynUNAw

Nicholas Renotte, "Tensorflow Multi-Person Pose Estimation with Python // Machine Learning Tutorial," Youtube. Aug. 30, 2021. Accessed on: Dec. 01, 2022. [Video File]. Available: https://www.youtube.com/watch?v=KC7nJtBHBqg

Nicholas Renotte, "Accessing USB Devices and Webcams with OpenCV and Python," Nov. 4, 2021. Accessed on: Nov. 29, 2022. [Video File]. Available: https://www.youtube.com/watch?v=FygLqV15TxQ&list=LL&index=257

CodeAsItIs, "Python MySQL Tutorial | How To Store Images, Documents (BLOB Data) In MySQL Database Using Python," Nov. 29, 2020. Accessed on: Jan. 10, 2023. [Video File]. Available: https://www.youtube.com/watch?v=Y6rHqQV2rRI&list=LL&index=19

Coding Shiksha, “Python OpenCV Tutorial To Capture Images From Webcam Full Project For Beginners,” Jun. 30, 2020. Accessed on: Nov. 30, 2022. [Video File]. Available: https://www.youtube.com/watch?v=IhRfqiC29Ds&list=LL&index=175

Augmented Cognition Laboratory, “SLP Dataset for Multimodal In-Bed Pose Estimation,” Jun. 27, 2019. Accessed on: Dec. 20, 2022. [Online]. Available: https://web.northeastern.edu/ostadabbas/2019/06/27/multimodal-in-bed-pose-estimation/

Appendices