Machine Learning Based Framework for Human Action Detection

Srividya M S, Anala M R

Department of Computer Science and Engineering, R V College of Engineering, Bengaluru, India

Corresponding author: Srividya M S, Email: srividyams@rvce.edu.in

Understanding human actions has been an important area of computer vision based deep learning domain. Several landmark extraction frameworks like media pipe and open Pose are used to extract the landmark coordinates from the body. The proposed work leverages open-source body landmark extraction and then trains a deep learning model on custom dataset created. The proposed work classifies the human body actions into blank face, yawn, namaste, punch and kick actions. The dataset creation phase involved recording of actions corresponding to every class and flattening them into a data frame. The dataset was later trained on a machine learning pipeline with machine learning algorithms like logistic regression, ridge classifier, random forest, and gradient boosting classifier. The algorithm with best accuracy was taken for real time usage. The landmark extraction model i.e., Mediapipe was used both in creation of dataset and execution of model in real time. The deep learning model was evaluated and validated based on several evaluation metrics like accuracy, confusion matrix, confidence score and recall score. The work proposed computationally efficient way of detecting the actions performed by the subject on camera by leveraging deep learning methods and mediapipe perception model for landmark extraction.

Keywords: Deep learning, Landmark, Machine learning, mediapipe, confusion matrix.

2023. In Satyasai Jagannath Nanda & Rajendra Prasad Yadav (eds.), *Data Science and Intelligent Computing Techniques*, 849–857. Computing & Intelligent Systems, SCRS, India. https://doi.org/10.56155/978-81-955020-2-8-72

1 Introduction

Understanding human actions has been an important area of computer vision-based machine learning domain. Several advancements have been made in this field. Many real-time application avenues will be opened if orientation of person [1] and his action is identified.With the improved feature extraction and processing models [2], it is now possible to create and train state of the art machine learning models to supplement the domain computer vision. Hence, leveraging such state-of-the-art feature extraction models [3, 4] in creation of custom machine learning models proves to be very efficient approach.Over the years, many approaches to Human Pose Estimation [5,6] were tried and tested. It forms a challenging domain as extraction of useful information from a video frame is hard under various conditions and has a direct impact on its performance. Various approaches have been incorporated over the years in this domain which comprises various deep learning-based paradigms like Open Pose, Deep Cut, and RMPE [6, 7] etc. Along with this there have also been advancements in facial landmarks extraction and hand position coordinate extraction using similar perception models.Body landmark extraction frameworks like media pipe [8] and open Pose [9] are used to extract the landmark coordinates from the body. The proposed work usesopen-source body landmark extraction and then traina machine learning model on custom dataset created by the team.The proposed work uses body, facial and hand landmark extraction frameworks to come up with a custom machine learning framework that recognises the action being performed in the video source. This work takes into account body, facial and hand landmark coordinates to recognise the action.Use of state-of-the-art frameworks to extract body joint coordinates from the input for effective computer-vision based modelling of the human body. Creation of a machine learning model which would be trained on a custom dataset.Usage of this model in the real time will be on web cameras to capture and detect body language accordingly. Comparison and analysis of various machine learning model architectures with respect to performance and limitations is done.

Creation of a custom dataset and transform it to a computationally less expensive format using state of the art feature extraction models which will drive the training of the machine learning model. It is followed by the creation of a custom machine learning model which takes in the body coordinates as features and provides a multiclass output based on the custom actions performed by the subject who is based on the training classes. Fine tuning and picking the best model architecture by analysing the performance.Computer vision creates a lot of opportunities to solve many real-world problems. This project aims to use the landmark concept of state-of-the are mediapipe's holistic model to perform body language detections. Holistic model comprises three models - each for face, body skeleton and hand. These three models in sync are used to perform body language detection. This approach is better than the contemporary computer vision techniques as the background colour; orientation of the objects not interest doesn't affect the results.

2 Literature Review

There have been various approaches proposed to perform pose estimation, hand tracking and face mesh detection [6]. Some of them are using 6 model architectures of CNN to classify input images without any body pose modelling framework [10].Some accomplish the results of body pose recognition, making it an entirely image processing-based approach [11]. Another work was designing a model that identifies the emotions of students from analysing their faces. It mainly does the recognitionin three steps: first is using Haar Cascades detecting theface, second is normalizing the face data and finallyusing CNN recognizing theemotion [10]. This was carried on the FER 2013 database. It has over seven types of expressions. There have been various other models that were used for the purpose like LSTM used for sign language detection [12,13,14] can model the contextual information of temporal sequence well.Architectures like OpenPose for multi-person real-timehuman pose estimation using 2D part affinity fields , etc. There were drawbacks incurred while using these models, they were too heavy models. The accuracy achieved by CNN model was not up to mark , it worked on70%accuracy rate for the FER 2013 database and wasn't very suitable for edge devices. The Resnet152 model has the

highest accuracy but is slow and heavy when it comes to deployment.It is based on the Faster R-CNN based detection algorithm [15,16]. When it comes to image processing using CNN it produces a huge computation overhead as it detects the coordinates among the whole image rather than focusing on the required part, the illumination has a certain impact on the detection. The results achieved by RNN model were better than CNN but because of recurrent behaviour in the model, the computation speedwill bereduced.This makes training theRNN model very difficult , prone to problems such as overfitting ,exploding and gradient vanishing.

Then came the era of LSTM models. Vanishing gradient problem was solved by LSTMs[17]. It was not completely resolved as it failed in some cases. The data evaluation still happens by cell-to-cell transfer of data. This is the root cause of the problem. Lot of resources and time is involved in training such models so that they are suitable enough for real time deployment. The model requires high memory as every cell has linear layers. This becomes a challenge for many systems. With enough research happening in the field of data mining, there are models other than LSTMs which are used to retain past data for a longer time. The performance of LSTMs will get affected by variation in weight initialization and so behaves similar to feed forward neural network[18]. So they are also prone to overfitting and applying dropout algorithm to tackle overfitting becomes difficult. Mediapipe is the modern approach for body landmark detection, it is an environment for iteratively improving ML applications with results reproducible across different devices and platforms. This is a lightweight and highly accurate model. With Mediapipe [3,4], the hand tracking module can be designed as a directed graph of components. Mediapipe graph used in the paper has of two subgraphs - one for hand detection and other for landmark computation. As no model is perfect there have been drawbacks with this model , the pose detection depends on the distance from the camera as well , the models propose excessive coupling which incorporates extra pre-processing steps and computation overhead, training the model can be sometimes time consuming, performance related issues on expanding the dataset,etc. Overall mediapipe has proven to be a better architecture compared to classic models, compared with running detection every frame, a pipeline with tracking function has several advantages: Provide instance-based tracking, that is, maintain the target ID across frames. No need to run the test every frame. These advantages allow us to run higher-load but more accurate detection models, while maintaining the lightweight and real-time performance of the pipeline. With the tracking function, the position of the target can be kept consistent in time, which means that the jitter of the target we observe in different frames will be smaller.There are contemporary approaches used to perform pose-detection, facial emotion analysis, hand-pose analysis [19, 20]. But none of the works combine the entire tree into a single model. Work carried out using mediapipe perception model provide an idea of using coordinate-based classification.

3 Methodology

The proposed system demands to have a mechanism where in each frame from video input has to be separated and introduced into a state-of-the-art landmark extraction model. Frame Extraction and Processing is the first step.This phase has equal importance in both dataset creation and real time deployment phase as each frame gets processed. Other important task is Dataset Generation. To be able to create a custom dataset from the video stream and encapsulate the extracted landmark coordinates into a data frame, there should be a system which automatically does the same to enhance the dataset creation process. Hence, a module is to be created which implements this functionality.Action/Gesture Classification is the final phase based on MachineLearning.Upon successful extraction of face and body landmarks through the input source, those landmarks are passed through machine learning/deep learning-based classifiers to detect the type of action/gesture performed by the subject under consideration from the given frame.To deploy the model in real time scenarios where the input is being taken from video sources like webcam, it is of paramount importance to have a GUI based interface which displays the action type along with the confidence metric to the user.The important modules of this work are shown in figure 1.

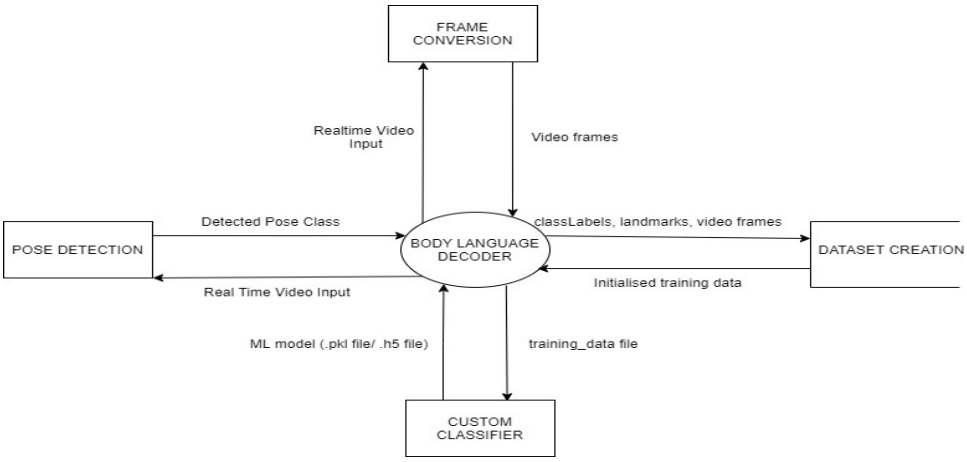

Figure 1: Block diagram of Body Pose Analyser

It is highly essential to extract every frame of real time video for both dataset creation and deployment as coordination extraction and processing takes place on every frame of a given input image. Data frame Initialization: This involves using extracted coordinates through perception models and flattening them onto data frame format which acts as a standardized way to create a dataset. Landmark Dataset Creation: Once coordinates are extracted and converted into data frame format, it is clubbed into a consolidated dataset in CSV format where number of columns signifies each coordinate value and the first column signifies the class. Initialize machine learning classifier: There are several machine learning algorithms which can be used for training, so a pipeline is created wherein all major classification algorithms are incorporated. The training was carried out using Logistic Regression, Ridge Classifier, Random Forest Classifier and Gradient Boosting Classifier.

Training the Machine Learning Pipeline: The machine learning pipeline which was initialized earlier is trained next with the dataset so created earlier. It is trained till the loss converges.Next step is Pose detection: The best algorithm is chosen from among the trained pipeline. The machine learning model so trained is leveraged to perform pose detection from real time video input. In the work, two different types of platforms were used based on the functional requirements. The following gives the details of the platform used.

a) Dataset Creation and execution platform: For creation of custom dataset based on Media pipe perception model, jupyter notebook was used which is a pythonic platform which is run locally on the system as it is highly essential to have access towebcam.

b) Machine Learning Platform: For training the machine learning pipeline for the created dataset, GoogleColab was used as it provides resources which are needed for hardware acceleration while training machine learning model on high dimensional dataset. Colab notebooks execute code on Google's cloud servers, which helps in leveraging the compute power of high-end hardware, including GPUs and TPUs, regardless of the power of the localmachine.

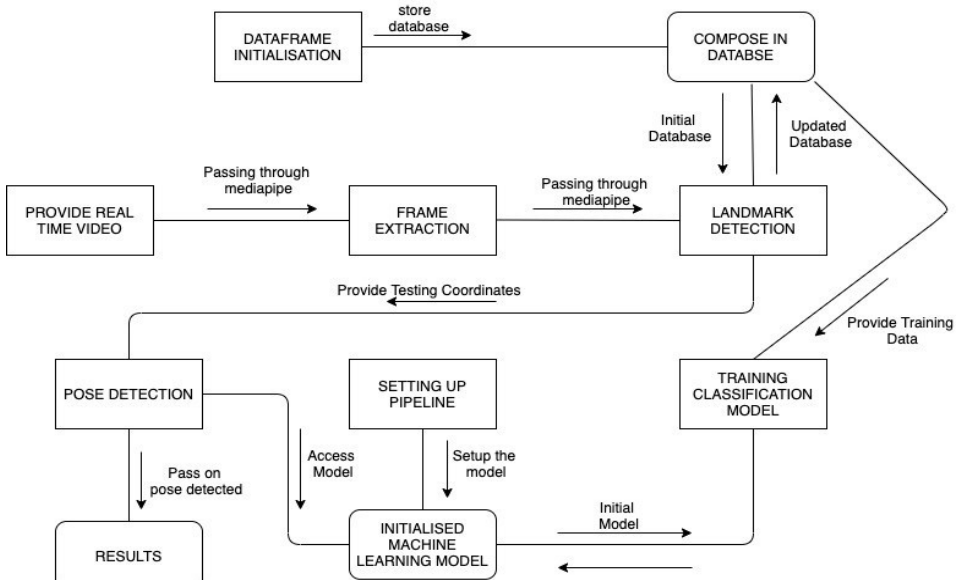

Figure 2: Architecture of Body Pose Analyser

The proposed pipeline for the project consists of Google Research's Mediapipe followed by custom trained ML classifier. This along with the custom training data used to train the ML model ensures fair results.Although the results of ML can have significant real-world impact, it is transparent in its behaviour. Transparency refers to the ability to observe the process that leads to decision making in the model.Also,efforts have been taken to address privacy issues when using personal data to train and test ML model.Test strategy is to systematically test and analyse the outcome of ML systems have been adopted.It is important for the ML based project to look at the reliability of individual ML predictions, focusing on reliability estimation. Other requirements for ML, such as sustainability or maintainability, have not been given significant attention. From a broader perspective, there are efforts to take into consideration specific quality trade-offs, e.g., privacy vs. processing time.

4 Experimental Dataset

The dataset used for training and validation was created by the team members. It is a CSV(comma separated values) file which consists of landmark coordinates for each of the landmarks used by the perception model. To be precise it is 468 for face, 33 for body skeleton and 21 each for left and right hand. Each landmark corresponds to a 4-tuple definition (x,y,z,v) - x, y, z are coordinate values and v stands for visibility. Visibility is a value between 0 and 1 which is based on to what extent the landmark is visible. Media pipe holistic model was used to create the dataset for the project.

The following steps were carried out to create the dataset:

1 Specify the class for which the data has to be created
2 Render the media pipe model to detect the landmarks

3 Extract landmarks from the holistic model, flatten the numpy array to a list and write the list to the CSV file
4 Class name is appended to the beginning of the row in the CSV file.

The above steps are carried out for each class that the model needs to be trained on and till the desired number of records for the class is achieved. The desired number refers to the count that won't cause over fitting or under fitting (bias or variance) in the multilabel classification model.

5 Accuracy and Performance

In order to measure the performance of machine learning based algorithms is by finding the algorithm accuracy using precision and recall. It gives a measure of correctness of the output with respect to the reality. The proposed work satisfies the accuracy and performance requirements when tested upon webcam supported by commercial computing resources. Creation of custom dataset covering all possible real-time edge cases ensures good mean average precision (mAP) and recall.

The confusion matrix, in this work will be a 6x6 matrix, as the number indicates the number of classes predicted. For this work 6 classes are - blank face, yawn, namaste, punch, kick, standstillare considered. In confusion matrix C, C[I, J]indicates the number of testcases which actually belonged to the group I but was predicted by the model to be in group J.

The below are few metrics with reference to confusion matrix:

- Accuracy: the ratio of the total number of correct predictions to the total dataset.
- Positive Predictive Value or Precision: the ratio of positive cases that were correctly identified to that of all correctly identified cases.
- Negative Predictive Value: the ratio of negative cases that were correctly identified to that of all correctly identified cases.
- Sensitivity or Recall: the ratio of positive cases that were correctly identified to that of all positives cases in the dataset.

Formula for Accuracy score:

```
Accuracy = Number of correct predictions / Total number of predictions
```

```
Recall Score = True Positives / (True Positives + False Negatives)
```

The result is a value between 0.0 for no recall and 1.0 for full or perfect recall.

The F1 score can be interpreted as a weighted average of the precision and recall. F1 score reaches its best value at 1 and worst score at 0. The relative contribution of precision and recall to the F1 score are equal.

```
F1 =2 * (Precision * Recall) / (Precision + Recall)
```

Confidence score is calculated as

```
Confidence score = Ture Positive / (True Positive + False Positive)
```

Performance of the multilabel classification model was analysed on the real-time webcam feed.

The testing was done using four different multi label classification models:Logistic Regression, Ridge Classifier, Random Forest Classifier and Gradient Boosting Classifier. The scores of these four ML classifiers are given in table 1.

Table 1: Performance scores of ML Classifier

Scores	Logistic Regression	Ridge Classifier	Random Forest Classifier	Gradient Boosting Classifier
Accuracy Score	0.97319	0.94190	0.93158	0.99425
Recall Score	0.92437	0.96890	0.99129	0.99502
F1 Score	0.99156	0.97343	0.97291	0.98573

Almost same Accuracy score as well as Recallscore means that the model is "balanced", that is, its ability to correctly classify positive samples is same as its ability to correctly classify negative samples.Confidence score will be displayed on the real-time frame along with the class_name using opencv-python. Some sample outputsare shown in figure 3.

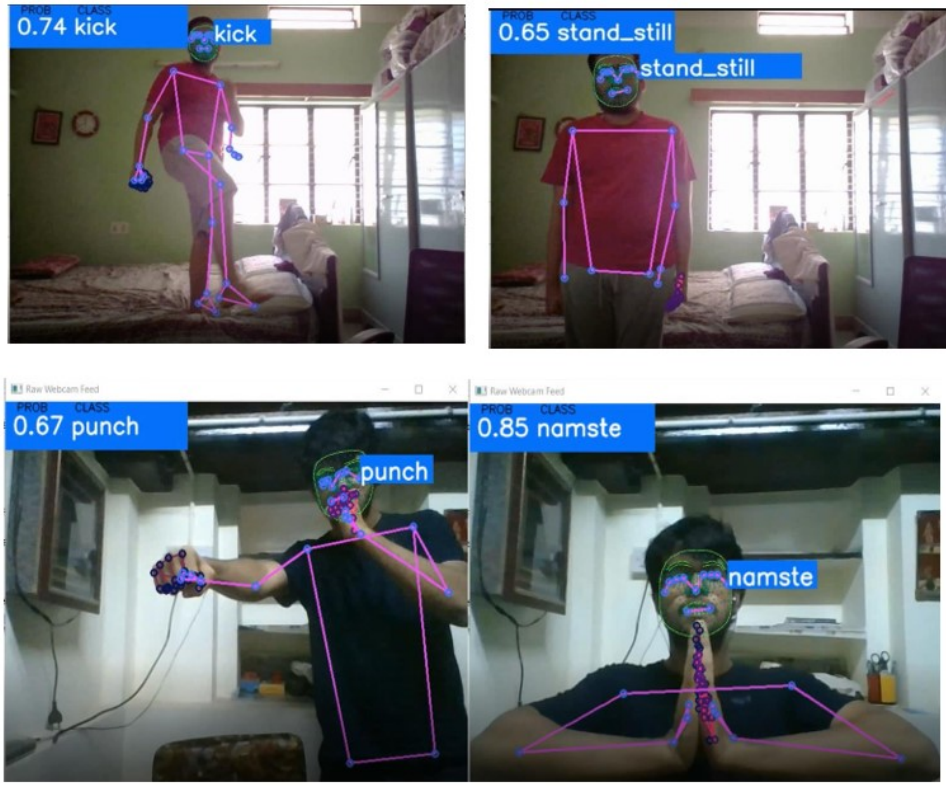

Figure 3: Sample Output

6 Conclusion and Future Enhancements

This work proposed an efficient and computationally effective way of detecting the actions performed by the subject on camera by leveraging the mediapipe perception model for landmark extraction. The model performs better than the conventional CNN models as classification happens based on the information extracted from the state-of-the-art mediapipe model. Also, mediapipe is better compared to other landmark extraction frameworks like Open-Pose as MediaPipe is able to achieve its speed thanks to the use of GPU acceleration and multi-threading.

6.1 Limitations of the Project

This work provided a means of creating an efficient action detection system leveraging the state-of-the-art media pipe by Google. But there are a few limitations to it as well. The system so created does not perform well in scenarios where a partial number of coordinates are captured in the video source. So, the less important features don't help the machine learning model in processing and giving accurate results.

6.2 Future Enhancements

Though the project performed really well in real time, it could get even better in terms of type of action by finding correlation between face and body pose coordinates. Also, time series models can be used to have a look back window by usage of LSTM type architectures to consider a set of action sequences to map to a particular class. There are several challenges to encounter in the creation of the same in terms of performance of the trained model vs. hardware requirements. Also, it is highly important to have a dataset to accomplish such a feat.

References

[1] L. Xie and X. Guo, "Object Detection and Analysis of Human Body Postures Based on TensorFlow," *2019 IEEE International Conference on Smart Internet of Things (SmartIoT)*, 2019, pp. 397-401, doi: 10.1109/SmartIoT.2019.00070.

[2] Lasri, A. R. Solh and M. E. Belkacemi, "Facial Emotion Recognition of Students using Convolutional Neural Network," *2019 Third International Conference on Intelligent Computing in Data Sciences (ICDS)*, 2019, pp. 1-6, doi: 10.1109/ICDS47004.2019.8942386.

[3] Lugaresi, Camillo & Tang, Jiuqiang& Nash, Hadon & McClanahan, Chris &Uboweja, Esha & Hays, Michael & Zhang, Fan & Chang, Chuo-Ling & Yong, Ming & Lee, Juhyun& Chang, Wan-Teh& Hua, Wei & Georg, Manfred & Grundmann, Matthias. (2019). MediaPipe: A Framework for Building Perception Pipelines. arXiv:1906.08172

[4] Fan Zhang, Valentin Bazarevsky, Andrey Vakunov, Andrei Tkachenka, George Sung, Chuo-Ling Chang, Matthias GrundmannMediaPipe Hands: On-device Real-time Hand TrackingarXiv preprint arXiv:2006.10214

[5] Ce Zheng□ ,Wenhan Wu□ , Taojiannan Yang, Sijie Zhu, Chen Chen, Member, IEEE, Ruixu Liu, Ju Shen, Senior Member, IEEE, Nasser Kehtarnavaz Fellow, IEEE and Mubarak Shah, Fellow, IEEE Deep Learning-Based Human Pose Estimation: A Survey arXiv:2012.13392v3

[6] Geetha Natarajan, E S Samundeeswari, A Review on Human Activity Recognition System INTERNATIONAL JOURNAL OF COMPUTER SCIENCES AND ENGINEERING Vol.-6(Issue-12)

[7] Arpita Haldera,AkshitTayade, Real-time Vernacular Sign Language Recognition using MediaPipe and Machine Learning International Journal of Research Publication and Reviews ISSN 2582-7421

[8] Antonio Domenech L ASL Recognition with MediaPipe and Recurrent Neural Networks Bachelor-Thesis

[9] Ishan Behoora and Conrad S. Tucker Machine learning classification of design team members' body language patterns for real time emotional state detection destUd 0142-694X

[10] RenatBashirovAnastasiaIanina Karim Iskakov Yevgeniy KononenkoValeriyaStrizhkova Victor Lempitsky Alexander VakhitovReal-time RGBD-based Extended Body Pose Estimation In Proc. IEEE 2021
[11] Braden Bagby, David Gray, Riley Hughes, Zachary Langford, and Robert Stonner Simplifying Sign Language Detection for Smart Home Devices using Google MediaPipe
[12] MR. Abid, EM. Petriu, E. Amjadian, "Dynamic sign language recognition for smart home interactive application using stochastic linear formal grammar," in IEEE Transactions on Instrumentation and Measurement, Sep. 2015, pp. 596-605
[13] W. Liu, Y. Fan, Z. Li, Z. Zhang, "Rgbd video based human hand trajectory tracking and gesture recognition system," in Mathematical Problems in Engineering, Jan. 2015
[14] R. Sharma, R. Khapra, N. Dahiya, "Sign Language Gesture Recognition.," in Sign, June 2020, pp.14-19
[15] Murakami K, Taguchi H. 1991. Gesture recognition using recurrent neural networks. In: Proceedings of the ACM SIGCHI conference on Human factors in computing systems, pp 237–242. https ://dl.acm.org/doi/pdf/10.1145/10884 4.10890 0
[16] Rekha J, Bhattacharya J, Majumder S. 2011. Hand gesture recognition for sign language: a new hybrid approach. In: International Conference on Image Processing, Computer Vision, and Pattern Recognition (IPCV), pp 80–86
[17] Das, P., Ahmed, T., & Ali, M. F. 2020, June. Static Hand Gesture Recognition for American Sign Language using Deep Convolutional Neural Network. In 2020 IEEE Region 10 Symposium (TENSYMP) (pp. 1762-1765). IEEE.
[18] T. Liu, W. Zhou, and H. Li. "Sign language recognition with long short term memory". In: 2016 IEEE International Conference on Image Processing (ICIP). 2016, pp. 2871–2875.
[19] Z. Cao, G. Hidalgo Martinez, T. Simon, S. Wei, and Y. A. Sheikh. OpenPose: Realtime multi-person 2d pose estimation using part affinity fields. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2019.
[20] Horkoff, Jennifer. (2019). Non-Functional Requirements for Machine Learning: Challenges and New Directions. 386-391. 10.1109/RE.2019.00050.