Football Player Posture Detection using Logistic Regression

Prof. Neha Patil
Dept of Computer Engineering
AISSMS IOIT
Pune, India

Ms. Gargi Agrawal
Dept of Computer Engineering
AISSMS IOIT
Pune, India

Ms. Sucheta Rakshe
Dept of Computer Engineering
AISSMS IOIT
Pune, India

Mr. Anuj Parmar
Dept of Computer Engineering
AISSMS IOIT
Pune, India

Mr. Rishikesh Nagale
Dept of Computer Engineering
AISSMS IOIT
Pune, India

Abstract—Due to recent advancements in artificial intelli-gence technology, advanced digital surveillance devices, real-time sports competition recordings, and other technologies, the benefits of deep learning-based visual identification tasks have become increasingly clear. It has a high degree of flexibility and openness, which points to the direction in which intelligent image processing will develop. In order to identify and recognise the posture of a football player, a proposed model for football pose recognition utilising machine learning algorithm is described here. The system is divided into two phases: the very first involves extracting data points from the image using the Media Pipe posture estimation library, and the second involves pre-processing the data and employing classification-based machine learning techniques to train and test the data. Three football player postures are supported by the suggested method. Moreover, a desktop application with a GUI is created using the Tkinter framework. Using Media Pipe and the OpenCV library, the live video input is pre-processed, the object is found, and the pose of the human body is recognised. The stored data that was trained from real-time video of different postures is used for testing and training purposes. For both training and testing, logistic regression models are used. 90% is the accuracy rating for the system. This developed model can be used by coaches to improve the performance of their players by observing and instructing them through their posture. It can also be utilised for training the players by analysing the posture of opponent team players. It is also useful in understanding the probability of goal.

Index Terms—MediaPipe, OpenCV, Logistic Regression, Ma-chine Learning

I. Introduction

Recognizing human poses is one of the most challenging tasks in the field of computer vision. It involves the process of locating human joints in an image or video to create a skeletal representation. Because it depends on a multitude of factors, including the size and dimensions of the image, differences in lighting, temperature variations, and interactions between people and their environment, the ability to auto-matically detect a user's activity in a photograph may be a problematic issue. Machine learning has tremendously helped human posture estimation in recent years, and substantial per-formance improvements have been made. Instead of needing to manually break down the connections between structures, machine learning technologies offer an extremely convenient method of mapping the structure. This study seeks to evaluate various methods for classifying football player poses and aims to gain knowledge about estimation. Sources from conference proceedings, published articles, technical documents, and jour-nals are used in this research. Then, various cause extraction techniques are discussed alongside machine learning-based models, including logistic regression and python libraries like Media Pipe for pose estimation and OpenCV for computer vision. This model tends to create user interfaces that are mostly based on desktop programmes where the user must first register before logging in. Once the user has successfully registered and logged in, a new window is opened where the user can utilise our system. For accessibility, there is a button labelled ”Recognize” that when pressed launches the camera and analyses the player's posture.

II. Literature Survey

Ahmad Jalal, Amir N Adeem from Air University Islamabad mentioned that this research was put forth to improve human body part recognition, specifically to accurately recognise body parts by modelling interactions between body parts. Segmenting the human body from both the background and from one another is necessary. They can then determine the body components' 3-D locations and detect them. The suggested system begins with the pre-processing of a particular image as input, and the recognition of a human outline is done by combining the methods for salient region detection and foreground detection. The outline of a person is then used to apply a body parts model to identify five fundamental body key points. In addition, using the previously found fundamental body key points, our approach also finds seven additional body sub key points.[1]

Mykhaylo Andriluka1,3, Leonid Pishchulin1, Peter Gehler2, and Bernt Schiele1 collaborated in order to provide a more realistic benchmark for human posture estimation. They proposed an alternative benchmark called ”MPII Human Pose” because they believed that the previous study hinders further research on the subject. With more than 40,000 photos of individuals, the benchmark greatly outperforms the state-of-the-art in terms of visual variability and complexity. Using searches based on the characteristics of more than 800 activities, we used YouTube as a source of data to gather photos and image sequences. As a result, there is a varied collection of photos that includes not only various activities but also interior and outside scenes, various imaging situations, and recordings from both amateur and expert sources.[2]

Alejandro Newell, Zhiao Huang, Jia Deng presented associative embedding, a cutting-edge technique for managing convolutional neural networks to perform detection and grouping. This approach can be used to frame a variety of computer vision issues, such as multi-person posture prediction and multi-object monitoring. Instead of using multi-stage pipelines, which are often used to group detections, we present a method that trains a network to simultaneously generate detections and group assignments. This method is simple to include into any cutting-edge network architecture that generates pixel-wise predictions. We demonstrate how to use this technique for estimating multiple-person poses.[3]

Andreas Launila authored a study that looks into the issue of real-time head posture estimation in blurry football video. A technique is described for determining the head pose using video clips and information about where the football and players are. As a support vector machine processes the video while performing pattern recognition on the location data, a group of randomised ferns is contrasted with a support vector machine for evaluating the video. The two sources of information work better together than alone. It turns out that knowing where the football is is crucial knowledge. Using a combination of bottom-up and top-down characteristics, a methodology for real-time prediction of the head position has been presented.[4]

Sande, Beaugendre, and Savas introduced the idea of target recognition and monitoring based on deep learning. Finding the target object’s placement inside the image—often indicated by the target’s bounding box—is the goal of the target detection issue. To solve these issues, Sande suggested that RCNN use a decision domain strategy to identify local candidate areas that potentially have detection target in the picture. The convolutional neural network would then receive these potential areas as input, gathering their properties and tying the classifiers together. Check the feature map to determine if the proper area is a component of the target that has to be detected before doing analysis on the validation frame to alter the location of the prediction frame.[5]

S. Yokozawa, M. Takahashi, H. Mitsumine, and T. Mishina created a technique for automatically detecting heads and estimating head poses for broadcasting football videos. To make operation of the system in the field simpler, the videos are captured by a single fixed wide field of view camera. By removing player regions and mining region forms, heads are detected. A pattern recognition method based on colour histograms and histograms of oriented gradients (HOG) characteristics is used to estimate head positions. The system is designed specifically to handle face photos with low quality. Head-pose metadata contributes to the clarity and interest of sports videos. A test using datasets from two fields revealed that the framework was more resistant to changes in camera locations and video quality than a traditional approach.[6]

Vasileios Belagiannis1, Sikandar Amin2,3, Mykhaylo Andriluka3,Bernt Schiele3, Nassir Navab1, and Slobodan Ilic analysed the problem of predicting a 3D human stance from different angles. This is a more challenging work than a single human 3D position estimate because of the significantly larger state variable, partial occlusions, and across sight inconsistencies while not having a clue about the identification of the individuals in advance. We initially reduce the state variable by triangulating the associated body joints that part analyzers in sets of camera angles have recognised. This resolves the problems. We offer a unique 3D pictorial structures (3DPS) model to address the ambiguities of incorrect and mixed body parts of various individuals after triangulation as well as those resulting from false positive body part detections. Our approach generalises 3D human body structures from the state space we have reduced. The 3DPS model can be used for both single-human pose estimate and multiple-human pose estimation.[7]

Yuan Chen proposed that many applications, including video analysis, intelligence surveillance, and robot navigation, depend on multi-object tracking (MOT). Reliable MOT is hampered by occlusion, intricate motion, and similar appearance. To address these issues, we here provide a long-term multi-object tracking technique. In our approach, a tracklet-based number recognition algorithm is proposed to produce trustworthy number labels, facilitating long-term tracking in football footage. Many tests are conducted to show that our approach achieves good performance and offers trustworthy predictions for football game assessment.[8]

III. Proposed Methodology

The proposed method is planned to work on both Linux and Windows operating systems. The next step of plan is to launch independent desktop application integrated with machine learning algorithm. The system’s environment is first built, and all required libraries are installed. The following libraries are required for the system: Sklearn for machine learning algorithm instances, OpenCV for computer vision tasks, Media Pipe for pose estimation, Seaborn for data visualisation, and NumPy for numerical calculations. Data gathering is completed after the environment is created. When testing a player whose posture needs to be analysed, the data collected during training of various player postures is employed. The Media pipe receives the images after that to estimate poses. The next step is to identify the frame's key points, after which it generates their 3-D coordinates and visibility value. The new dataset is then built on top of these key points, and each key point's coordinate and visibility are added as features.

The data is then cleaned up using data pre-processing techniques to make it appropriate for machine learning models. The dataset is normalised for pre-processing of the data. The process of normalisation involves setting each value in the dataset to a number between 0 and 1. The primary basis for doing this is that some machine learning algorithms need normalised data to function well and produce accurate findings. Next, using the dataset's existing features, feature engineering is used to create new features.

Key points are converted into vectors in order to achieve this. Joint angles are determined using these vectors, which represent 3-dimensional body parts. The classification-based machine learning algorithms are then given all the processed data in order to train them. First, testing data must be gathered or produced. This information must be brand-new and cannot include any duplicate information from the training phase. Then, to make testing data comparable to training data, identical pre-processing and feature engineering are applied to the testing data. This is because machine learning models are capable of comprehending training data. The output for the test data is then obtained, and it is supplied to the trained model so that it may be evaluated on the basis of its predictions.

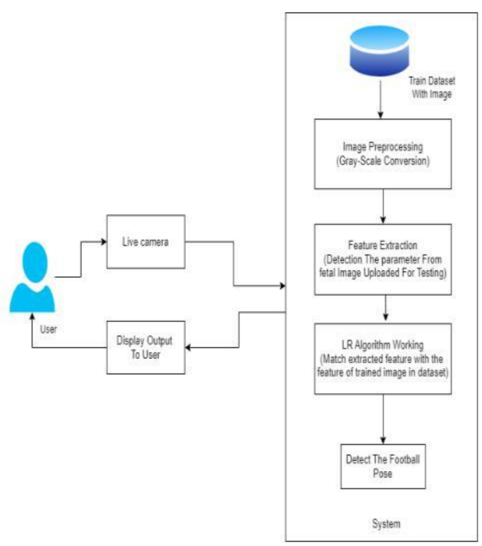

Fig. 1. System Architecture of the proposed system

Machines must have at least 8GB of RAM in order to run the model. Anaconda navigator, which comprises of Spyder and command prompt, is the software required for putting the suggested system into practise. It is compatible with Windows, Linux, and Mac OS. The DBSqlite server is utilised to provide database connectivity.

Dataset collection and Pre-processing is done using following steps:

1) Pre-Processing: For data preprocessing, we used the Python Imaging Library (PIL). It offers crucial features including an extensive file format, effective internal representation, the ability to create thumbnails, change the format of image files, and apply filters to pictures. Use the pip install pillow command to add the PIL library to your machine.
2) Feature Extraction:The OpenCV library and Media Pipe are used to extract features. At the time of feature extraction, the below mentioned steps are followed:
   a) The image samples of targeted postures are collected and then pose predictions are run on them.
   b) The collected pose landmarks must next be trans-formed into a representation that is appropriate for the classifier and assembled into a training set. They are used to transform key points into vector format.
   c) The classification is then carried out appropriately.
3) Modelling: Every posture position's structure is based on the x, y, and z coordinates of the joint points. The data is classified and the pose is found using a logistic regression model. The X (input variable) represents feature data, while the Y (output variable) represents the targeted value when the coordinates x, y, and z are provided to the model. Data is used for testing 30% of the time and training 70% of the time. Almost 90% accuracy is provided by the suggested model.

IV. Algorithms

1) Logistic Regression (LR)

   A strong and recognized technique for supervised classification is logistic regression (LR). It can only represent a dichotomous variable, which frequently specifies whether an event will occur or not, and can be thought of as an extension of conventional regression. The likelihood that a new instance belongs to a particular class can be determined with the use of LR. Given that the result represents a probability, it falls between 0 and 1. As a result, a threshold must be set to distinguish between two classes in order to use the LR as a binary classifier. For instance, if an input instance's probability value is greater than 0.50, it will be assigned to "class A"; otherwise, "class B." A categorical variable with more than two values can be modeled using the LR model in a more general way. This generalized version of LR is known as the multinomial logistic regression.

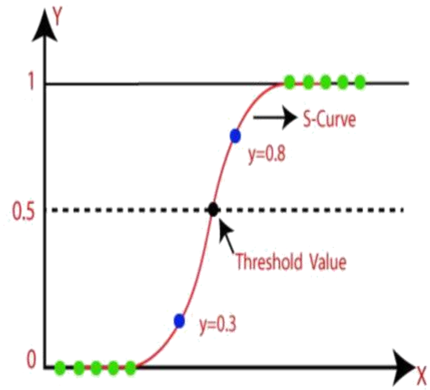

Fig. 2. LR as a binary classifier

2) Random Forest (RF)

A classification model called random forest is made up of multiple decision tree classification models, and each individual model has the power to cast one vote for the most practical and effective classification model. The fundamental idea is to use a lot of decision trees during training, and then during decision time, average out all of their outputs. The fact that RF can avoid decision trees' overfitting problem is its greatest advantage over those systems.

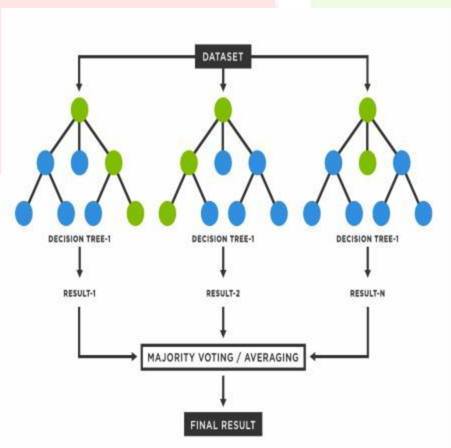

Fig. 3. Random Forest Classifier

V. Results

With the help of logistic regression, ridge classifier, random forest classifier, and gradient boosting classifier, the system was able to achieve 90% accuracy for the first three and 89% for the last. As a final prediction model, logistic regression was employed for making the system usable.

ALGORITHM	PRECISION	RECALL	F1-SCORE	SUPPORT
Logistic Regression	1.0	1.0	1.0	651
Ridge Classifier	0.99	0.99	0.99	651
Random Forest Classifier	1.0	1.0	1.0	651
Gradient Boosting Classifier	1.0	1.0	1.0	651

Fig. 4. Comparison of Algorithms

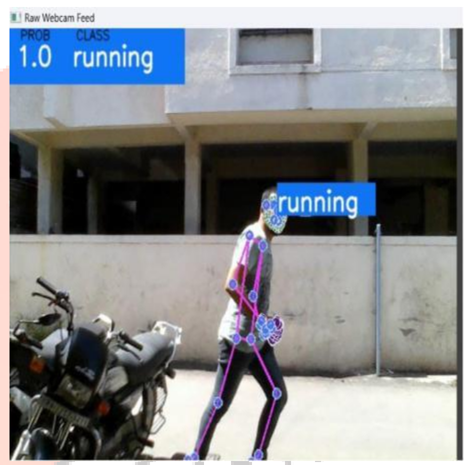

Fig. 5. Action identified as Running

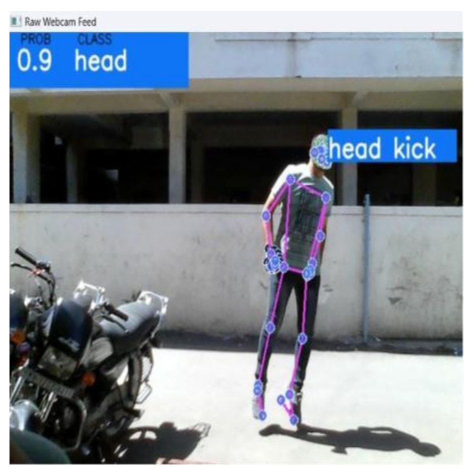

Fig. 6. Action identified as Head Kick

Fig. 7. Action identified as Leg Kick

VI. CONCLUSION

A method that can more correctly determine football players' posture in real time is suggested in order to serve the sports community of football players. Using machine learning, the suggested system can recognise various postures. A live desktop camera feed is transmitted to the server while the football player is on the field. The server uses numerous interconnected modules to forecast and create output of the pose and accuracy. The user is presented with a guidance of the predicted pose in a real-time application. To find patterns between key points, the time distributed LR layer is used. By minimising the error caused by incorrect key point detection and using LR to store the records of previous frames and polling for denoising, the results further increase the system's robustness.

REFERENCES

[1] Y. Amit and D. Geman. Shape Quantization and Recognition with Randomized Trees. Neural Computation, 9(7):1545–1588, 1997.

[2] P.K. Atrey and M.S. Kankanhalli. Probability Fusion for Correlated Multimedia Streams. In ACM International Conference on Multimedia, pages 408–411, 2004

[3] S.O. Ba and J.M. Odobez. A probabilistic framework for joint head tracking and pose estimation. In International Conference on Pattern Recognition, volume 4 pages 264–267, 2004.

[4] B. Benfold and I. Reid. Colour invariant head pose classification in low resolution video. In British Machine Vision Conference, 2008.

[5] S.T. Birchfield and S. Rangarajan. Spatiograms Versus Histograms for RegionBased Tracking. In IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005

[6] A. Bosch, A. Zisserman, and X. Munoz. Image Classification using Ran-dom Forests and Ferns. In IEEE International Conference on Computer Vision, pages 1–8, 2007.

[7] L. Breiman. Bagging Predictors. Machine learning, 24(2):123–140, 1996

[8] M. Briers, A. Doucet, and S. Maskell. Smoothing algorithms for state-space models. Technical report, Cambridge University Engineering Department, CUED/F-INFENG/TR.498, 2004

[9] L.M. Brown and Y.L. Tian. Comparative Study of Coarse Head Pose Estimation. In IEEE Workshop on Motion and Video Computing, pages 125–130, 2002.

[10] C.C. Chang and C.J. Lin. LIBSVM: a library for support vector machines, 2001.

[11] W. Rawat and Z. Wang. (2017). Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review.Neural Computation. 29. 1-98. 10.1162/NECOa00990

[12] "Joint Training of a Convolutional Network and a Graphical Model for Human Pose Estimation," by J. Tompson, A. Jain, et al. 2014 Eprint Archive:1799–1807.

[13] X. Chu, W. Ouyang, H. Li and X. Wang, "Structured Feature Learn-ing for Pose Estimation," 2016 IEEE Conference on Computer Vi-sion and Pattern Recognition (CVPR), 2016, pp. 4715-4723, doi: 10.1109/CVPR.2016.510.

[14] F. Bogo, A. Kanazawa, C. Lassner, P. Gehler, J. Romero, and M. Black. Keep it smpl: Automatic estimation of 3d human pose and shape from a single image. In ECCV, 2016.

[15] G. Hua, M. Yang, and Y. Wu. Learning to Estimate Human Pose with Data Driven Belief Propagation. In IEEE Computer Society Conference on Computer Vision and Pattern Recognition, volume 2, page 747, 2005.

[16] Pauzi A.S.B. et al. (2021) Movement Estimation Using Mediapipe BlazePose. In: Badioze Zaman H. et al. (eds) Advances in Visual Informatics. IVIC 2021. Lecture Notes in Computer Science, vol 13051. Springer, Cham.

[17] Syed Abdul Gaffar Shakhadri — May 18, 2021, Pose Estimation using OpenCV, Analytics Vidhya

[18] Cramer, Jan Salomon. "The origins of logistic regression." (2002)

[19] Bazarevsky, Valentin, et al. "BlazePose: On-device Real-time Body Pose tracking." arXiv preprint arXiv:2006.10204 (2020)

[20] End-to-End Learning for Joint Detection and Grouping with Associative Embedding 2017 Alejandro Newell, Jia Deng, and Zhiao Huang

[21] E. Bulbul, A. Cetin and I. A. Dogru, "Human Activity Recognition Using Smartphones," 2018 2nd International Symposium on Multidisciplinary Studies and Innovative Technolo-gies (ISMSIT), Ankara, 2018, pp. 1-6, doi: 10.1109/ISMSIT.2018.8567275

[22] Yang Wang, Hao Jiang, M. S. Drew, ZeNian Li and G. Mori, "Unsuper-vised Discovery of Action Classes," 2006 IEEE Computer Society Con-ference on Computer Vision and Pattern Recognition (CVPR'06), New York, NY, USA, 2006, pp. 1654-1661, doi: 10.1109/CVPR.2006.321.