Human Pose Estimation for Yoga Using VGG-19 and COCO Dataset: Development and Implementation of a Mobile Application

Dhadkan Shrestha[1], Peshal Nepal[2], Pratik Gautam[3], Pradeep Oli[4]

[1]Texas State University, San Marcos, TX, US 78666
[2]Georgian College, Barrie, ON L4M 3X9, Canada
[3]Georgian College, Barrie, ON L4M 3X9, Canada
[4]Thapathali Engineering Campus, Kathmandu, Nepal

---***---

Abstract - *Human Pose Estimation (HPE) is a critical technology in computer vision with diverse applications ranging from healthcare to sports analysis. This project presents a method for detecting the 2D stance of multiple persons in an image using a nonparametric representation known as Part Affinity Fields (PAFs). By leveraging the first 10 layers of the VGG-19 convolutional neural network and training on the COCO dataset, our model effectively identifies and associates key points of the human body.*

The architecture employs a two-branch system that jointly learns part locations and their associations through sequential prediction. This enables the model to maintain real-time performance while achieving high accuracy, regardless of the number of persons in the image. To enhance accessibility, we developed a mobile application using Flutter and TensorFlow Lite, allowing real-time pose estimation via a mobile device's front camera. The app provides immediate feedback on physical exercises and yoga poses, making it an invaluable tool for fitness enthusiasts and healthcare professionals. Visual outputs such as heatmaps and PAFs confirm the model's capability to accurately localize and connect key points. Despite potential challenges such as data quality and hyperparameter tuning, the results indicate that our approach is both reliable and practical for real-world deployment. This project not only advances the state-of-the-art in HPE but also opens possibilities for future enhancements, including integrating 3D pose estimation and applying the technology in augmented and virtual reality applications.

Key Words: Human Pose Estimation (HPE), Convolutional Neural Network (CNN), VGG-19, Part Affinity Fields (PAFs), COCO Dataset, Real-Time Pose Detection

1. INTRODUCTION

Human Pose Estimation (HPE) is the process of identifying, tracking, predicting, and classifying the movement and orientation of the human body through input data from images or videos. It captures the coordinates of the joints, including the knees, shoulders, and head. The three primary approaches to modeling a human body are Skeleton-based, Contour-based, and Volume-based models [1]. HPE has been evolving with the advancement of artificial intelligence and has applications in human-computer interaction, augmented reality, virtual reality, training robots, and activity recognition [2]. HPE is critical in various fields such as healthcare, sports, and entertainment. In healthcare, it is used for monitoring and analyzing physical therapy exercises to ensure patients perform movements correctly, reducing the risk of injury. In sports, it aids in performance analysis, helping athletes improve their techniques. In entertainment, HPE enables the creation of more interactive and immersive experiences in video games and virtual reality.

There are several approaches to modeling a human body in pose estimation, which can be broadly categorized into three types:

- Skeleton-based Models: These models represent the human body as a collection of joints connected by bones. The coordinates of the joints are tracked over time to understand the movement and posture.
- Contour-based Models: These models focus on the outer contour of the body, capturing the silhouette to infer pose and movement.
- Volume-based Models: These models create a volumetric representation of the body, capturing the full 3D structure, which is useful for more detailed analysis.

HPE can be divided into two primary techniques:

2D Pose Estimation: This technique involves estimating key points in the joints of the human body in the 2D space for the image or video. It serves as a foundation for more advanced computer vision tasks like 3D human pose estimation, motion prediction, and human parsing.

3D Pose Estimation: This technique involves estimating the actual spatial positioning of the body in the 3D space, introducing the z-dimension. It provides a more comprehensive understanding of the body's posture and movement [3].

With the advancement of deep learning and computer vision, significant progress has been made in HPE. Convolutional Neural Networks (CNNs) have been widely used to improve the accuracy and efficiency of pose estimation. Libraries such as OpenPose, DeepCut, and AlphaPose have been developed, offering robust solutions for real-time multi-person pose estimation.

VGG-19, a convolutional network that is 19 layers deep, is known for its performance in large-scale image recognition tasks. It has been used in this project for feature extraction in human pose estimation. By utilizing the first 10 layers of VGG-19, we extract features from input images, which are then processed through various stages to acquire key points and part affinity fields.

The use of Part Affinity Fields (PAFs) allows the model to capture the spatial relationships between different body parts, enabling accurate detection of poses even in complex scenarios. By integrating these features into a mobile application, we aim to make pose estimation accessible and easy to use for a wide range of applications, from exercise monitoring to interactive gaming.

2. LITERATURE REVIEW

Human Pose Estimation (HPE) has evolved significantly over the past decades with advancements in computer vision and deep learning techniques. Initially, HPE relied on simpler models and smaller datasets, which limited the accuracy and applicability of the methods.

2.1 Early Models and Approaches

- Pictorial Structures: The concept of pictorial structures was introduced by Fischler and Elschlager in the 1970s. This approach represented objects using a collection of parts and their spatial relationships [4]. Felzenszwalb and Huttenlocher later made this method practical and tractable using the distance transform trick, which significantly improved its efficiency and accuracy [5].

- Datasets: Earlier models used smaller datasets like Parse and Buffy for evaluation. However, these datasets were not suitable for training complex models due to their limited size and variability. The introduction of larger datasets, such as the Leeds Sports Pose (LSP) dataset containing 10,000 images, marked a significant milestone in the development of HPE models [6].

2.2. Advancements with Larger Datasets

- COCO Dataset: The COCO (Common Objects in Context) dataset is a large-scale object detection, segmentation, and captioning dataset that has become a standard benchmark for HPE. It provides a diverse set of images with annotated key points, making it ideal for training and evaluating HPE models [7].

- MPII Human Pose Dataset: The MPII dataset is another extensive dataset that includes around 25,000 images with annotated body joints. It covers a wide range of human activities and poses, providing a robust benchmark for HPE algorithms [8].

2.3. Key Libraries and Frameworks

Several libraries and frameworks have been developed to facilitate HPE, offering robust and efficient solutions for both single-person and multi-person pose estimation.

- **OpenPose**: Developed by Zhe Cao and his team in 2019, OpenPose is a real-time multi-person key point detection library capable of detecting 135 key points. It uses a bottom-up approach, which is efficient for handling multiple persons in an image. OpenPose is trained on COCO and MPII datasets and has become one of the most popular tools in HPE [9].

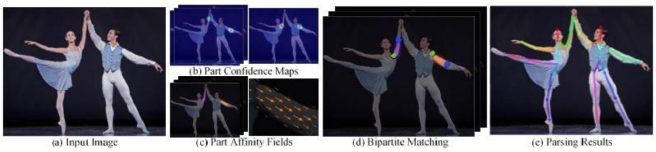

Figure-1: Pipeline of Real-time 2D Pose Estimation

- DeepCut: Presented by Leonid Pishchulin in 2016, DeepCut uses a bottom-up approach with Integral Linear Programming to model detected key points and form a skeleton representation. It addresses the challenge of multi-person pose estimation by simultaneously detecting and associating body parts [10].

- AlphaPose: Developed in 2016, AlphaPose uses a top-down approach for human pose estimation. It detects human bodies first and then localizes key points within the detected regions. AlphaPose supports various operating systems and is known for its high accuracy and robustness [11].

2.4. Convolutional Neural Networks (CNNs) in HPE

CNNs have revolutionized the field of HPE by providing powerful tools for feature extraction and pattern recognition.

- VGG-19: VGG-19 is a convolutional network that is 19 layers deep and was trained on the ImageNet database. It can classify more than 1,000 objects and is known for its performance in image recognition tasks. In this project, we use the first 10 layers of VGG-19 for feature extraction, which provides the basis for detecting key points in human poses [12].

- High-Resolution Net (HRNet): Introduced by Jingdong Wang, HRNet maintains high-resolution representations throughout the entire network. It has been used for semantic segmentation, object detection, and HPE, providing high accuracy and detailed pose estimations [13].

In this research project, we utilize the VGG-19 model for feature extraction. By using only the first 10 layers, we balance the need for detailed feature extraction with computational efficiency. The extracted features are processed through a series of CNN layers to generate Confidence Maps and Part Affinity Fields, which are used to determine the full-body pose. The final model is deployed in a mobile application, making it accessible for various use cases such as exercise monitoring.

3. METHODOLOGY

3.1 Data collection and preprocessing

The backbone of our Human Pose Estimation (HPE) model is the COCO (Common Objects in Context) dataset, which is a large-scale dataset containing over 2 million images with annotated key points. The diversity and extensive scale of this dataset make it ideal for training robust HPE models. The initial step involved filtering and annotating images using COCO's annotation files, which include detailed information about each image's size, bounding boxes, segmentation, and key point locations [14]. From this dataset, we selected approximately 65,000 images with crucial points necessary for our training purposes.

We normalized the images to ensure consistency and improve the model's performance. The normalization process involved scaling the pixel values using the formula (x/256 - 0.5) that represents the pixel values. This transformation standardized the pixel values to fall within the range of, making the data more suitable for training the neural network. Furthermore, we converted each key point into a 32 x 32 x 17 matrix, which represents the probability function for the key points [15]. These matrices were essential for generating heatmaps that the model would use to learn the spatial distribution of key points.

3.2. Model Architecture

The model architecture is built upon the VGG-19 convolutional neural network, specifically utilizing the first 10 layers for feature extraction [16]. VGG-19 is well-regarded for its performance in image recognition tasks due to its deep architecture and the use of small, 3 x 3 convolution filters, which effectively capture intricate details in the images. The output from these layers provided a robust set of features that served as the foundation for detecting key points and part affinity fields in the subsequent stages [16].

The extracted features were processed through a series of CNN layers, organized into stages. The first stage consisted of five convolutional layers designed to further refine the features extracted by VGG-19. The first three layers used 3 x 3 x 128 filters, the fourth layer used 3 x 3 x 512 filters, and the fifth layer employed 1 x 1 x 17 filters [17]. Each convolutional layer was followed by a ReLU activation function, introducing non-linearity and enabling the model to learn complex patterns [18].

In stages 2 through 6, the architecture branched into two separate paths: one path was responsible for generating heatmaps, while the other generated Part Affinity Fields (PAFs). These branches contained layers with 7 7 128 kernels, and the final layers had 1 x 1 x 128 and 1 x 1 x 34 kernels. The heatmaps represented the probability of key points in a two-dimensional space, while the PAFs depicted the location and orientation of limbs, forming pairs in the image domain [19].

3.3 Model Training

The training and validation of the model were carried out using a split from the COCO dataset, where 50,000 images (90%) were used for training and 5,000 images (10%) for validation. This division ensured that the model had ample data to learn from while also providing a separate set of images to evaluate its performance. The training process spanned multiple epochs, during which the model's parameters were optimized to minimize the loss functions for both branches (heatmaps and PAFs).

The loss function for the heatmaps was defined as:

$$L_{\text{heatmap}} = \sum_{j} \sum_{p} \| S_j(p) - S_j^*(p) \|_2^2$$

Where Sj(p) represents the predicted heatmap for the point j at position p, S*j(p) represents the ground truth heatmap [20]. Similarly, the loss function of PAFs was defined as:

$$L_{\text{PAF}} = \sum_{c} \sum_{p} \| L_c(p) - L_c^*(p) \|_2^2$$

Where Lc(p) represents the predicted PAF for Limb c at position p, and L*c(p) represents ground truth PAF [20].

3.4. Data flow

The data flow for the Human Pose Estimation project starts with collecting the COCO dataset, which is partitioned into training, validation, and testing subsets. Data IO processes ensure proper loading and saving of data during preprocessing, training, and evaluation. Samples are drawn for training and validation to refine the model.

Model selection involves choosing the architecture and hyperparameters, initializing the VGG-19 network for feature extraction, and defining loss functions for heatmaps and Part Affinity Fields (PAFs). The Adam optimizer is used to adjust model parameters, minimizing the loss functions.

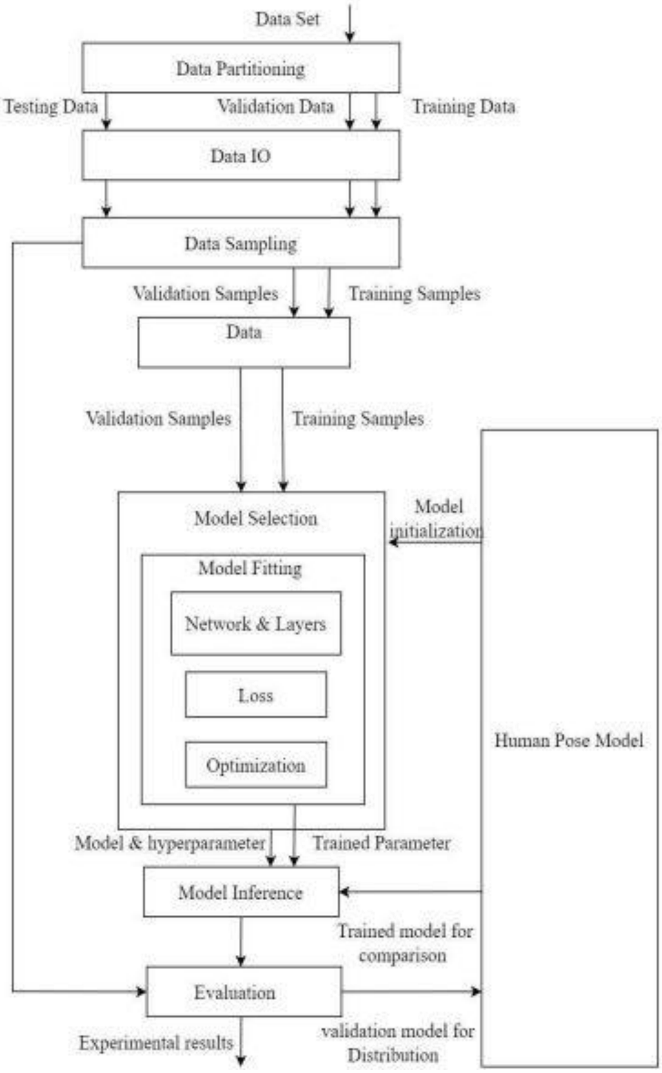

Figure-2 DataFlow Diagram

The model undergoes fitting, with layers configured for extracting features and generating heatmaps and PAFs. Hyperparameters are fine-tuned to enhance performance. Model inference applies the trained model to validation samples, followed by evaluation using metrics like accuracy, precision, recall, F1 score, and Mean Squared Error (MSE).

Experimental results provide insights into model performance, with comparisons to benchmark its effectiveness. The validated model is then prepared for deployment in a mobile application, allowing real-time pose estimation using a mobiledevice's front camera. This comprehensive data flow ensures the model is accurately trained and capable of effective humanpose estimation.

3.5. Deployment in Mobile Application

To make the HPE model accessible and user-friendly, we deployed it as a mobile application using Google's Flutter framework. Flutter is an open-source framework that allows developers to create natively compiled applications for mobile, web, and desktop from a single codebase. This choice ensured that our application could run efficiently on various devices.

We used TensorFlow Lite to deploy the trained model on mobile devices. TensorFlow Lite is a lightweight version of TensorFlow designed specifically for mobile and embedded devices, providing efficient performance and low latency. The application was equipped with features for real-time pose estimation, using the front camera of a mobile device to capture and analyze poses.

The application's user interface was designed to be intuitive, allowing users to select different exercises or yoga poses and receive instant feedback on their performance. The feedback mechanism used visual indicators, such as green for correct poses and red for incorrect poses, to help users adjust their posture in real time.

The model's performance was evaluated using standard metrics such as precision, recall, F1-score, and Mean Squared Error (MSE). Precision-measured the accuracy of key point detection, recall assessed the model's ability to detect all relevant key points, and the F1-score provided a balanced measure of accuracy by combining precision and recall. MSE evaluated the difference between the predicted and actual key point locations, quantitatively measuring the model's accuracy.

4. RESULT AND ANALYSIS

The Human Pose Estimation project achieved significant results through the implemented methodology. This section details the outcomes, including accuracy and loss analysis, heatmaps, part affinity fields, mobile application output, and error analysis.

4.1. Accuracy and Train Loss

The training process involved monitoring the accuracy and loss metrics to evaluate the model's performance over successive epochs. The metrics provided insights into the model's learning progress and helped identify any potential issues.

- Epoch vs Training Accuracy: Training accuracy was tracked over each epoch to measure the model's ability to correctly predict key points during the training phase. The training accuracy showed a steady improvement, indicating that the model was learning effectively from the training data.

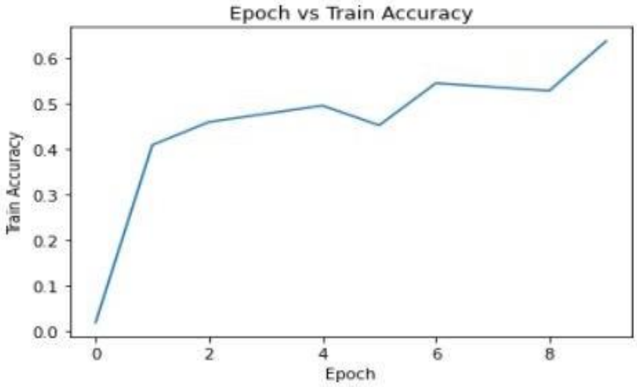

Figure-3. Epoch vs Training Accuracy

- *Epoch vs Validation Accuracy:* Validation accuracy was tracked to measure the model's performance on unseen data. This metric was crucial for assessing the model's generalization ability. The validation accuracy also showed a consistent improvement, demonstrating that the model was not overfitting and could generalize well to new data.

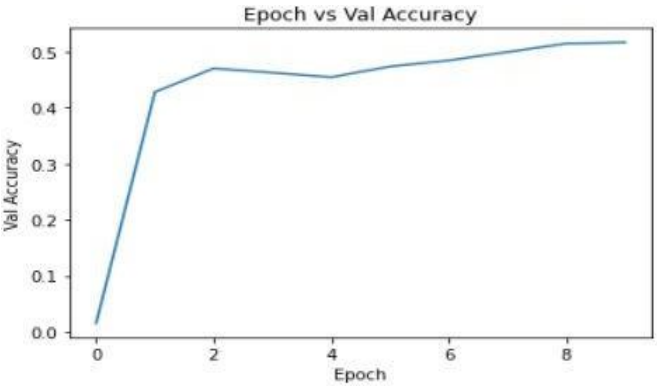

Figure-4: Epoch vs Validation Accuracy

- *Epoch vs Training and Validation Accuracy:* A combined plot of training and validation accuracy provided a comprehensive view of the model's performance. Both metrics showed a similar trend, further confirming that the model was learning effectively without overfitting.

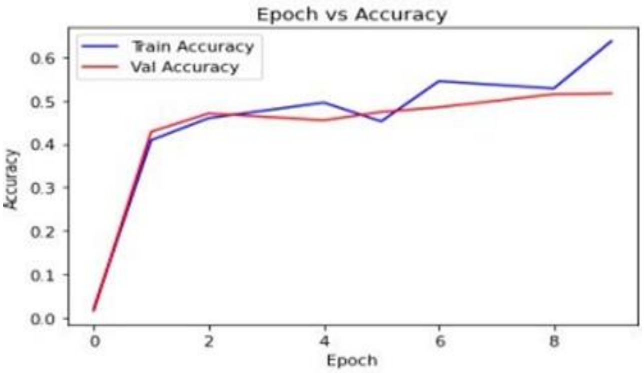

Figure-5: Epoch vs Training and Validation Accuracy

Epoch vs Training Loss: Training loss was monitored to evaluate the model's error in predicting key points during the training phase. The loss showed a decreasing trend, indicating that the model's predictions were becoming more accurate over time.

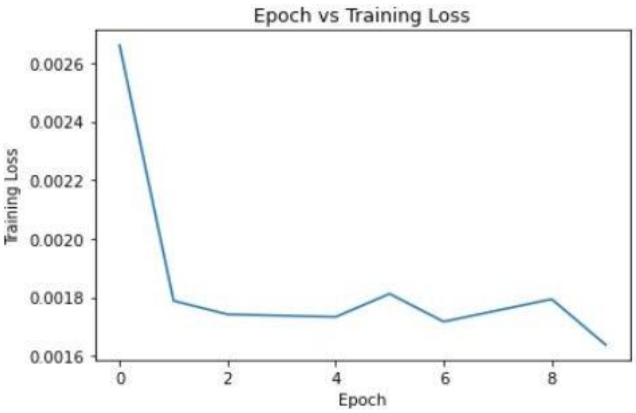

Figure-6: Epoch vs Training Loss

4.2 Output

The project successfully visualized human poses by detecting 17 key points and joining them to form a skeleton-like structure. The key points included the nose, eyes, ears, shoulders, elbows, wrists, hips, knees, and ankles.

1. Heatmaps: Heatmaps were generated to visualize the probability of key point locations in a two-dimensional space. The heatmaps provided a clear representation of where the model predicted each key point to be. Each key point was represented by a separate heatmap, showing color variations to indicate the probability of the key point's occurrence [21].

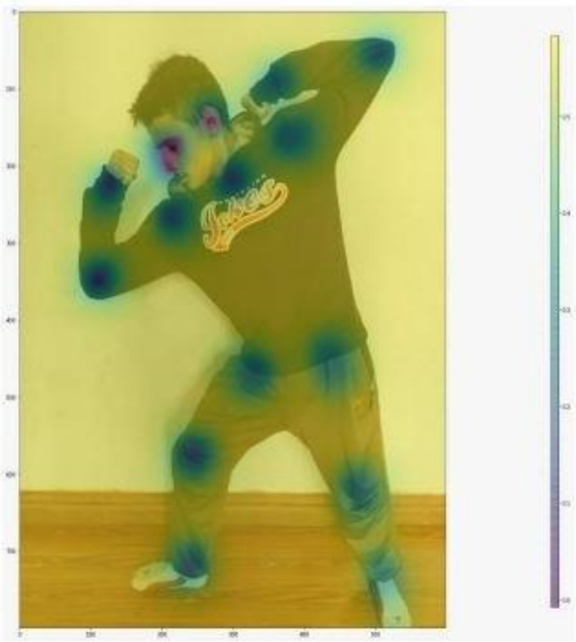

Figure-7: Heatmap of all keypoints

Part Affinity Fields (PAFs) were used to depict the location and orientation of limbs by forming pairs of key points. The PAFs were represented as 2D vector fields, providing direction vectors between key points that needed to be connected. The model generated a 32 32 34 matrix for PAFs, which was then processed using a greedy algorithm to identify the closest key points and connect them, forming a complete human skeleton.

Figure-8: PAF of the body part

1) *Joining Key Points and Final Output:* The key points detected by the model were represented by different colors. These key points were then joined to form the final output, a skeleton-like structure that accurately represented the human pose.

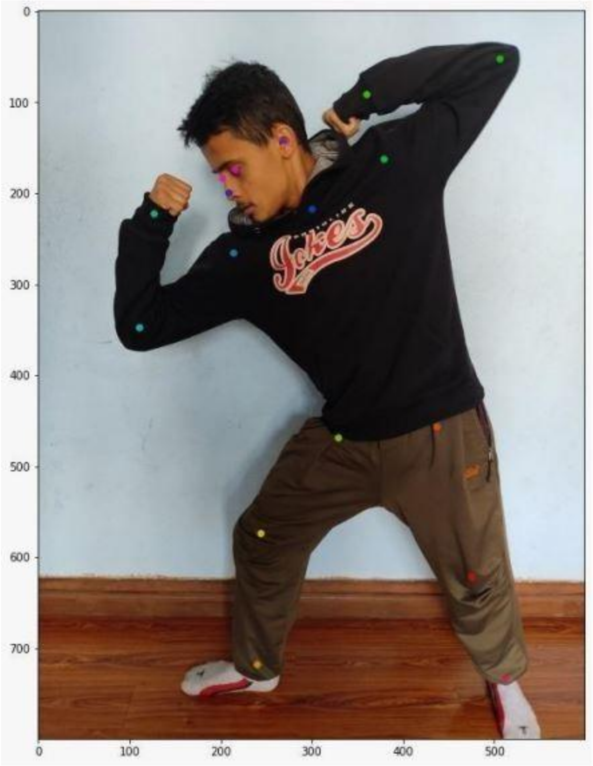

Figure-9: key points of body represented by different color

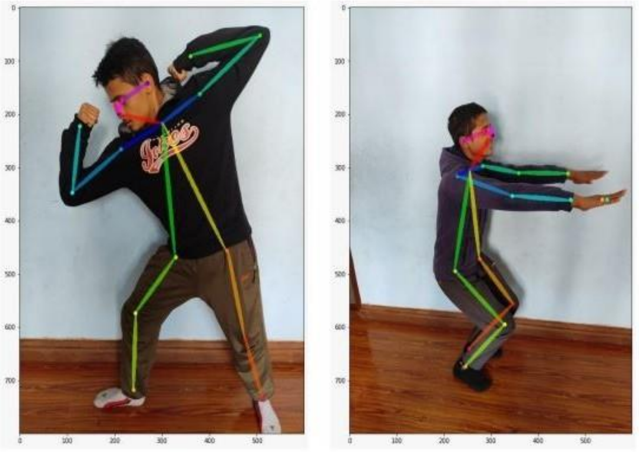

Figure-10: keypoint joining with the final output

4.3 Output from Mobile Applications

The model was deployed in a mobile application to provide real-time pose estimation using the front camera of a mobile device. The application could detect exercises such as squats and arm raises, providing visual feedback to the user on their performance.

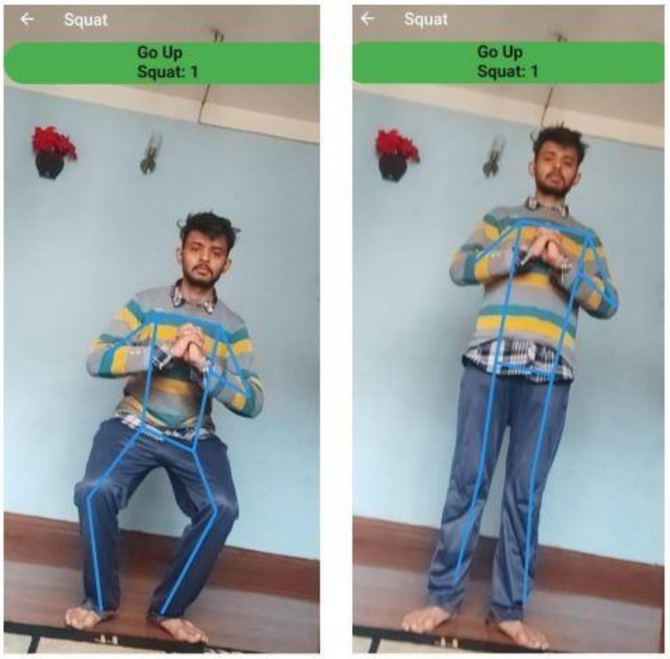

Figure-11: Output of Squat in App

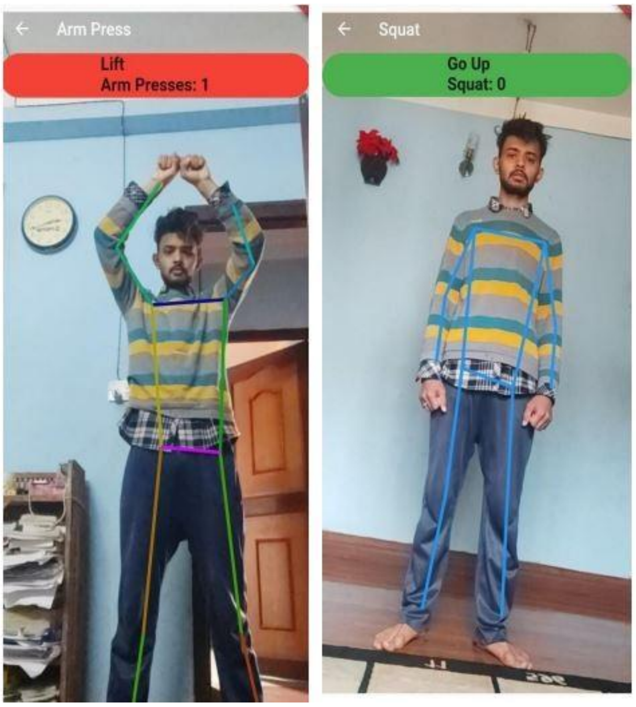

Figure-12: Output of lift in App

4.4. UI of Mobile App

The user interface of the mobile application was designed to be intuitive, allowing users to select exercises and receive instant feedback. The app displayed the detected key points and skeleton overlay on the camera feed, helping users adjust their posture in real time.

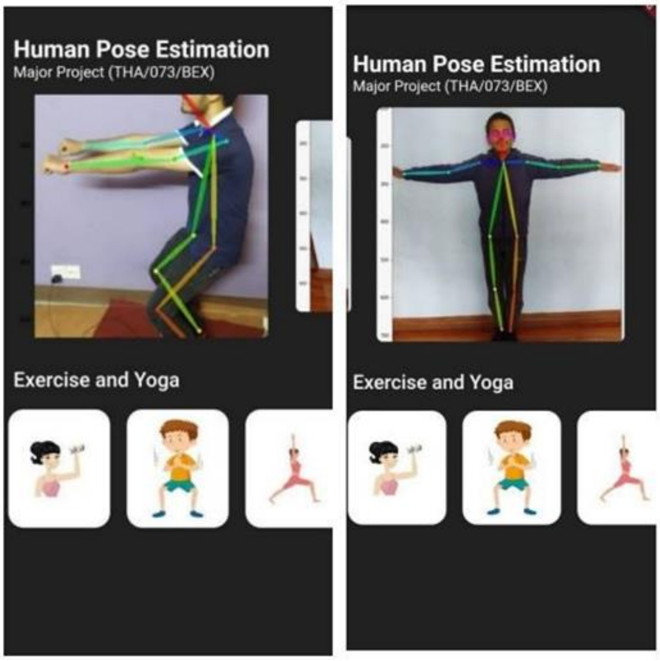

Figure-13: UI of yoga app

5.Conclusion

The Human Pose Estimation project successfully demonstrated the ability to detect and visualize human poses using deep learning techniques. The model achieved high accuracy in predicting key points and forming a skeleton-like structure, making it suitable for various applications such as exercise monitoring and interactive gaming. Despite the challenges and potential sources of error, the project provided reliableand consistent results, proving its feasibility for real-world deployment [22].

The project demonstrated significant potential for future enhancements, including:

Integrating face recognition and detection for defense applications.

Enhancing the system to predict user movements is usefulin defense and gaming.

Applying pose estimation in CGI for movies and video games.

Using 3D cameras for capturing three-dimensional human poses, provides better visualization and accuracy.

REFERENCES

[1] N. Barla, "V7Labs," [Online]. Available: https://www.v7labs.com/blog/human-pose-estimation-guide.

[2] P. Ganesh, "Towards Data Science," 15 March 2019. [Online]. Available: https://towardsdatascience.com/human-pose-estimation-simplified-6cfd88542ab3.

[3] M. A. Fischler and R. A. Elschlager, "The Representation and Matching of Pictorial Structures," IEEE Trans. Comput., vol. C-22, no. 1, pp. 67-92, 1973, doi: 10.1109/T-C.1973.223602.

[4] P. F. Felzenszwalb and D. P. Huttenlocher, "Pictorial structures for object recognition," Int. J. Computer. Vis., vol. 61, no. 1, pp. 55-79, 2005.

[5] L. Johnson and C. Everingham, "Clustered Pose and Nonlinear Appearance Models for Human Pose Estimation," in Proc. British Mach. Vis. Conf., 2010, pp. 1-11, doi: 10.5244/C.24.12.

[6] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," IEEE Transl. J. Magn. Jpn., vol. 2, pp. 740-741, Aug. 1987 [Digests 9th Annual Conf. Magnetics Japan, p. 301, 1982].

[7] T.-Y. Lin et al., "Microsoft COCO: Common Objects in Context," in European Conf. Computer. Vis. (ECCV), 2014, pp. 740-755, doi: 10.1007/978-3-319-10602-1_48.

[8] M. Andriluka et al., "2D Human Pose Estimation: New Benchmark and State of the Art Analysis," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2014, pp. 3686-3693, doi: 10.1109/CVPR.2014.471.

[9] Z. Cao, G. Hidalgo, T. Simon, S.-E. Wei, and Y. Sheikh, "OpenPose: Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields," IEEE Trans. Pattern Anal. Mach. Intell., vol. 43, no. 1, pp. 172-186, 2019, doi: 10.1109/TPAMI.2019.2929257.

[10] L. Pishchulin et al., "DeepCut: Joint Subset Partition and Labeling for Multi-Person Pose Estimation," in Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 2016, pp. 4929-4937, doi: 10.1109/CVPR.2016.533.

[11] H. Fang et al., "RMPE: Regional Multi-person Pose Estimation," in Proc. IEEE Int. Conf. Computer. Vis. (ICCV), 2017, pp. 2334-2343, doi: 10.1109/ICCV.2017.254.

[12] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," arXiv preprint arXiv:1409.1556, 2014.

[13] J. Wang et al., "Deep High-Resolution Representation Learning for Visual Recognition," IEEE Trans. Pattern Anal. Mach. Intell., vol. 43, no. 10, pp. 3349-3364, 2020, doi: 10.1109/TPAMI.2020.2983686.

[14] T.-Y. Lin, M. Maire, S. Belongie, L. Bourdev, R. Girshick, J. Hays, P. Perona, D. Ramanan, C. L. Zitnick, and P. Dollar, "Microsoft COCO: Common Objects in Context," in European Conf. Comput. Vis. (ECCV), Zurich, Switzerland, 2014, pp. 740-755.

[15] K. Simonyan and A. Zisserman, "Very Deep Convolutional Networks for Large-Scale Image Recognition," in Int. Conf. Learn. Representations (ICLR), San Diego, USA, 2015.

[16] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, "Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields," in Proc. IEEE Conf. Computer. Vis. Pattern Recognit. (CVPR), Salt Lake City, USA, 2017, pp. 7291-7299.

[17] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in Int. Conf. Learn. Representations (ICLR), San Diego, USA, 2015.

[18] X. Peng and K. Saenko, "Synthetic to Real Adaptation with Generative Correlation Alignment Networks," in Proc. IEEE Winter Conf. Appl. Computer. Vis. (WACV), Lake Tahoe, USA, 2018, pp. 1982-1991.

[19] A. Newell, K. Yang, and J. Deng, "Stacked Hourglass Networks for Human Pose Estimation," in European Conf. Computer. Vis. (ECCV), Amsterdam, Netherlands, 2016, pp. 483-499.

[20] S. Johnson and M. Everingham, "Learning Effective Human Pose Estimation from Inaccurate Annotation," in Proc. IEEE Conf. Comput. Vis. Pattern Recognition. (CVPR), Boston, USA, 2011, pp. 1465-1472.

[21] D. Shrestha and D. Valles, "Evolving Autonomous Navigation: A NEAT Approach for Firefighting Rover Operations in Dynamic Environments," in *2024 IEEE International Conference on Electro Information Technology (eIT)*, Eau Claire, WI, USA, 2024, pp. 247-255, doi: 10.1109/eIT60633.2024.10609942.

[22] D. Shrestha, "Comparative Analysis of Machine Learning Algorithms for Heart Disease Prediction Using the Cleveland Heart Disease Dataset," *Preprints*, vol. 2024, no. 2024071333, 2024. https://doi.org/10.20944/preprints202407.1333.v1