O-Nect: Open Source Interface for Motion Capture using RGB Camera

Lenix Lobo
Pune Institute of Computer Technology
Pune, India

ABSTRACT
The use of neural networks for evaluation of Human Pose Estimation has been around for a long time in the field of entertainment , gaming , modeling using Motion Capture systems. However , these systems require expensive hardware installations. Motion capture provides several advantages over traditional animation methods. However,the cost of hardware equipment ,personnel and software makes it highly ineffective for low budget designers to obtain and process the essential data for their projects. Complex movement animations and realistic physical interactions can be easily recreated using our approach with minimal hardware investment.

In this paper, we discuss an open source library : O-Nect which can be utilized for Motion Capture using a simple RGB camera. Motion Capture based interactive applications could be potentially beneficial while designing interactive humanoid robots.

General Terms
Computer Vision and Deep Learning

Keywords
O-Nect

1. INTRODUCTION
In this project, the aim is to eliminate dedicated hardware systems and attain efficient performance using minimal tools. As the project is Open Source, developers across the world can utilize the features from our project to create their own projects.

With the recent research being done in human pose estimation, we provide a easy to develop user interface and constant feedback through the open source community.

We take a broader look at the application of this product when the open source community is able to contribute to add their features,improve performance to help improve the overall product.

2. THE APPROACH
We use the Openpose[1] implementation , add a few modifications, and use socket based approach to provide a communication module between the game engine and the pose estimation algorithm.

Our approach is capable of obtaining a 2D pose of a human from just a RGB camera. Estimating 2D pose[2] from a single

RGB camera is a challenging problem. We provide an overview of our method to tackle this challenging problem. Our network consists of two primary components. The first is a convolutional neural network (CNN) to regress 2D and 3D joint positions.

Fig below illustrates the overall pipeline of our method. The video footage frame is first analyzed by a convolutional neural network (MobileNet architecture) generating a set of feature maps F that is input to the initial stage of each branch of the network.

At first a set of confidence maps S and a set of part affinity fields L are produced.

The predictions from both the layers are then concatenated and utilized to produce new predictions. Two loss functions are applied at the end of each stage of each branch are applied to guide the network to iteratively produce confidence maps of body parts. L2 loss between estimated predictions and ground truth maps and field is determined.

The system takes, as input, a color image of size w h and produces, as output, the 2D locations of anatomical keypoints for each person in the image . A feedforward network simultaneously predicts a set of 2D confidence maps S of body part locations and a set of 2D vector fields L of part affinities, which encode the degree of association between parts . The set S = (S1, S2, ..., SJ) has J confidence maps, j 1 . . . J. The set L = (L1,L2, ...,LC) has C vector fields, one per limb1 , where Lc R(wh2) , c 1 . . . C, each image location in Lc encodes a 2D vector.

Finally, the confidence maps and the affinity fields are parsed by greedy inference to output the 2D keypoints for all people in the image. The finally obtained co-ordinates are then sent through a threaded socket interface for interaction between different game engines. Currently we use a blender game engine interface , but this may be extended later as per the developers convenience.

3. RESULTS
Fig below illustrates the result/output performance of our pipeline. We obtain real-time video input from the RGB camera, process it through the CNN architecture, process and pass the co-ordinates through a socket system, retrieve the co-ordinates in Blender and obtain real-time motion in the blender game engine.

In the original Openpose repository, the architecture is built using a VGG19 architecture for the keypoint detection problem. However, this results in slower performance in mobile/lower-end systems. To overcome this issue, we have replaced the Vgg19 network with a Mobile-Net [3]architecture, resulting much better performance.

In our approach, we use a architecture inspired by the Mobile-Net architecture, which provides us with a better,smoother performance on some affordable systems.

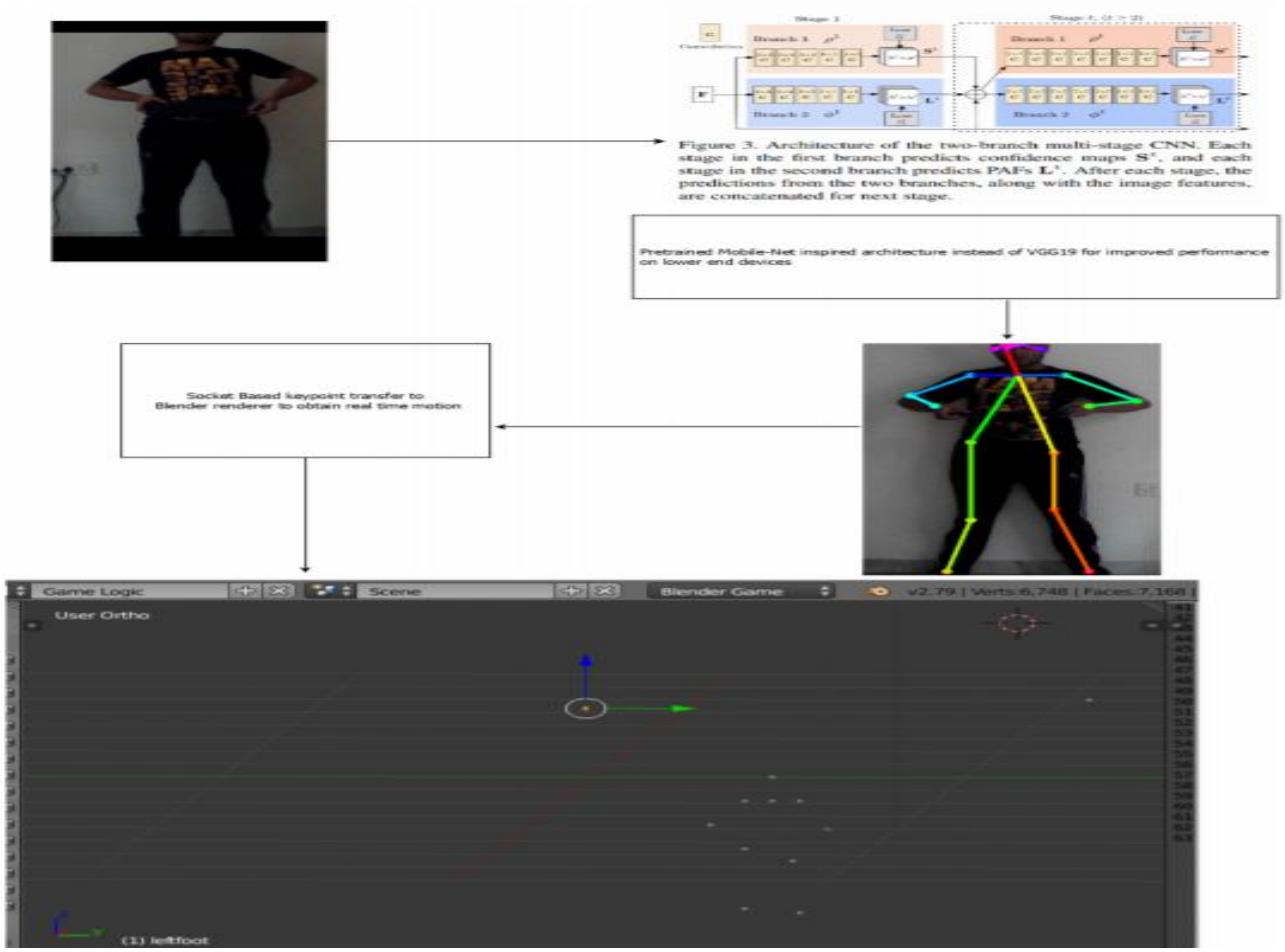
4. PERFORMANCE
To analyze the runtime performance of our method, we used a blender generated keypoint humanoid . The final

product is implemented on a laptop with a 4GB NVIDIA GeForce GTX-1050Ti GPU and a 720p Laptop camera. We used person detection and single-person CPM to generate a JSON file of keypoints which was later sent thtough a socket pipeline , retrieved in the game engine and utlilised accordingly.

We were able to achieve an average of 15 fps on our mentioned specs and are working on improving the performance with the help of the open source community . This may also be further improved using multi-threading techniques such that the performance can be achieved on lower end mobile devices as well.

5. O-Nect Figure representation

Figure below describes the picture of the O-Nect pipeline used for keypoint detection of key body points of a given human in the current input Video-Frame



6. ACKNOWLEDGMENTS

We acknowledge the efforts of the creators of the ”OpenPose: Real-time multi-person keypoint detection library for body, face, and hands estimation” to provide a highly efficient architecture for our application.

7. REFERENCES

[1] Zhe Cao and Tomas Simon and Shih-En Wei and Yaser Sheikh Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields. CVPR,2017.

[2] Shih-En Wei and Varun Ramakrishna and Takeo Kanade and Yaser Sheikh Convolutional pose machines. CVPR,2016.

[3] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, Hartwig AdamMobileNets: Efficient Convolutional Neural Networks for Mobile Vision Applications. CVPR.

[4] M. Andriluka, S. Roth, and B. SchieleMonocular 3D pose estimation and tracking by detection. CVPR,2010.

[5] V. Belagiannis and A. Zisserman.Recurrent human pose estimation. In 12th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG), 2017

[6] X. Chen and A. Yuille Articulated pose estimation by a graphical model with image dependent pairwise relations.In NIPS, 2014

[7] P.F. Felzenszwalb and D P.Huttenlocher Pictorial structures for object recognition. In IJCV, 2005

[8] [U.IqbalandJ.Gall Multi-person pose estimation with local joint-to-person associations.In ECCV Workshops, Crowd Understanding, 2016.

[9] A. Toshev and C. Szegedy. Deeppose: Human pose estimation via deep neural networks. In CVPR, 201