

Human Pose Estimation Using Deep Neural Networks

Anurag Chandana¹, Bhupendra Ram², Mukesh Tiwari³

^{1,2,3}Assistant Professor, Department of CSE, Quantum University, Roorkee, India

Abstract- Every tracking mechanism requires object detection where object tracking is the process in which locating an object or multiple objects is done using either the static or dynamic camera. It is important and challenging to detect and track objects in real-time. The recent focus of computer vision research has been on detecting and tracking multiple objects in dynamic environments. The position of a person or object in an image or video can be inferred using pose estimation, a task in computer vision. Pose estimation is a problem that involves determining the position and orientation of a camera in relation to a particular person or object. Identifying, locating, and tracking a number of key points on a given object or person is the typical way to do this. Corners or other significant features can be significant for objects, while in humans, these key points represent major joints like an elbow or knee. Tracking these key points in images and videos is the objective of our machine learning models. CNN can be utilized to detect yoga postures and their probability.

Keywords: Deep Learning, Machine Learning, Convolutional Neural Network, Tensorflow, Keras.

I. INTRODUCTION

The task of human pose estimation is to determine the position and orientation of a person's body parts in an image. The field of motion and capture can greatly benefit from this technique. Pose estimations fall into two categories:

- An RGB image is used to estimate 2D pose (x, y) coordinates for joints.
- An RGB image is used to estimate 3D pose (x, y, z) coordinates for joints.

We have proposed a Deep Neural Network model that helps us track the yoga pose and the probability that the user is in which yoga pose among the poses provided in the dataset. This is accomplished by using a Deep Neural Network, which is a Convolutional Neural Network.

1.1 Human Pose Estimation

Human pose estimation is emerging in many object detection and computer vision fields, such as human computer interaction, action recognition, surveillance, picture understanding, threat

prediction, and so on. The concept of the human pose estimation is to detect the joints of the human body and the postures done by the user[2]. In this technique to localize the key points is done by combining the neural networks.[2]

II. ALGORITHM USED

2.1 Deep Neural Network

The deep neural network used is actually a convolutional neural network. The DNN takes an RGB image through video which captured by the real time video/stored videos/images and then by using the transformation techniques we will get to learn the denser layers and features of the images. The original image can be represented using these extracted features and selections.

2.2 Convolutional Neural Network

Convolutional neural networks are feedforward networks in which the output of the previous layers is given as input for the next layers. Convolutional layers in the CNN are used to extract features like lines and edges from an image after selecting them.

It is passed on to the next layers and undergoes different processes such as activation, maximum pooling, flattening, etc. And these processes are repeated until we go through all the dense layers of the Convolutional Neural Network.

Deep neural network counterparts called ConvNets are fully connected and use the back propagation algorithm to perform end-to-end feature learning. First, the two properties help to reduce the number of free parameters and also reduce the process of feature detection at different locations in the input. Small input translations are invariant to the learned representation in the third property.

After that the pipeline with a pixel of 64*64 starts where the input patch are local contrast normalized which is called as LCN[3].Where it emphasize geometric discontinuity and then it also improve the generalization performance[4].The LCN layer comprises of the 9*9 pixel local subtractive normalization, and a 9*9 local divisive normalization is followed after that. Three convolution and subsampling layers are used to process the input using ReLU, which refers to rectified linear units and maxpooling. Three basic steps for human pose estimation can be included in the CNN.

- Overfitting can be reduced by the sparsity of the connections in convolution. Convolution with the help of pooling can provide us with the detection of location-invariant features. Sharing parameters is part of it.
- ReLU is an activation function that introduces nonlinearity and speeds up training and computation.
- To reduce dimension and computation, reduce overfitting, and make the model more tolerant to minor distortions and variations, pooling can be utilized.

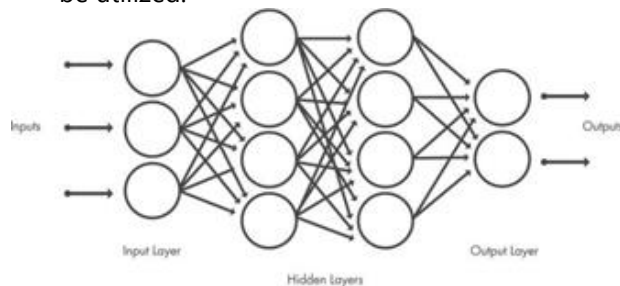


Figure 1: CNN

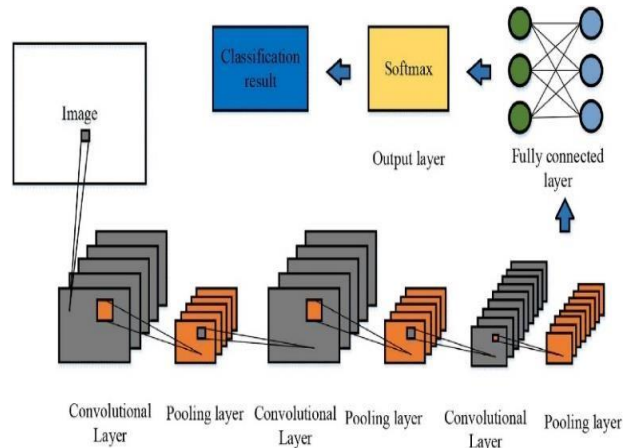


Figure 2: Convolutional Neural Network – based on the human movement recognition

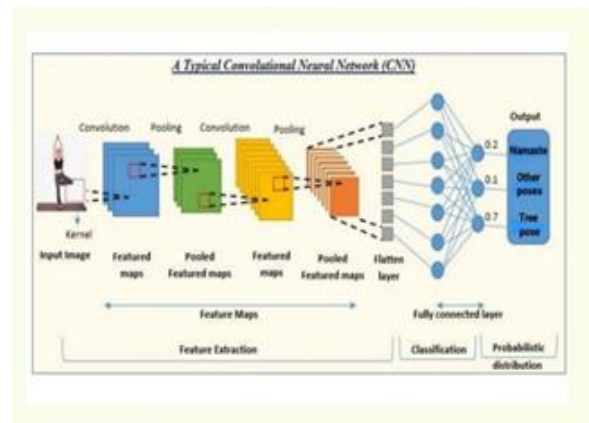


Figure 3: A typical convolutional neural network (CNN) detecting the posture with the probability that which pose it is.

2.3 Steps For Detecting Yoga Poses Using Cnn Algorithm And Showing The Probability Of The Poses

Step 1

Feature Extraction: The Feature extraction will reduce the dimensionality of an image and then the large number of pixels will be reduced and then the images can be captured easily and effectively by using the useful and the interesting parts of the image.

1: Conv2D: Then this is used for the creation of the convolutional kernel. Then the kernel is then convolved with the layer input and then the tensor of outputs are produced. The Kernel is a filter used for extracting the features form the images.

2: Activation (ReLU): The Convolutional Neural Network consist of the non-linearity layer which consist of the activation function. And then the activation map is created by using the feature map generated by the convolutional layer. Here the ReLU activation function can be used for avoid the problem of vanishing gradient. It is one of the hidden layer.It also helps to pmprove the computation performance.

3: Max pooling: Then max pooling is a process where the dimension is reduced and that is done by reducing the pixel from the previous CNN layer. Also we can say it is used for downscaling of the image.

Step 2

Classification layers: The classification layers can compute the cross-entropy loss , this is done for classification and for the weighted classification tasks with the mutually exclusive classes.

1: Flatten: The flatten is used to flatten all the multidimensional tensor input to one dimensional and also the flatten helps to change the shape of the data from a vector which is of 2d matrix to format for a dense layer to interpret.

2: Dense: The dense is where the output of previous neuron is sent to the next neuron. Based the CNN the dense is able to classify the images that which pose is done by the user.

3: Dropout: The dropout is used for overfitting of the data so that the training data does not adapt the extra or the unwanted features.

4: Activation (Softmax): Then at last we use the activation where softmax activation can be used to scale the numbers, etc. into the probability.

2.4 Uses

1. CNN can help us for feature extraction which are directly learned by the CNN and there is no need for doing any manual feature extraction.
2. The CNN are also able to produce model which provides us with a highly accurate recognition results.

3. With the help of CNN we can retrain a model for new recognition tasks which will enable us to build it on the pre-existing network.

2.5. Tensorflow And Keras

TensorFlow is an open-source software library. TensorFlow was originally developed by the researchers and engineers working on the Google Brain Team within Google's Machine Intelligence research organization for the purposes of conducting the machine learning and the deep neural networks research, but the system is general enough to be applicable in a wide variety of other domains as well!

Keras is an open-source software library that provides the Python interface for an artificial neural networks. Keras also acts as an interface for the TensorFlow library. Keras allows users to productize the deep models on smartphones (iOS and Android), on the web, or on the Java Virtual Machine.[3] It also allows the use of distributed training of the deep-learning models on clusters of Graphics processing units (GPU) and tensor processing units (TPU).

III. THE STEPS AND IMPLMENTATION OF HUMAN POSE EXTIMATION USING DNN(CNN)

1. User can register on the web application and if already registered the user can login using his/her credentials.
2. Users gets two options for detecting the pose:
 - By uploading the recorded video
 - By turning on the real time camera
3. After capturing the user's movements the CNN algorithm is applied.
4. By comparing the captured poses with the poses in the datasets and after going through the CNN algorithm output is displayed
5. On the screen where poses are captured the name of the pose detected is been displayed.
6. With that the probability of which pose is been detected is also been displayed.
7. After the use the user can directly close the web application.

2. Pose estimation can also be used in the animation field.
3. It can also be used Activity Recognition
4. In motion capture and augmented reality
5. For training robots where the robots can follow the trajectories of a human skeleton where the human is performing some actions.
6. It can be used in some fields where it can be used for the motion tracking for consoles.
7. And also it can be used to track human postures and detect the poses and can be used for AI based yoga or AI based gym Trainers.

Future Scope

Pose estimation is a type of computer technology that uses vision techniques to detect the location of a person or an object. This can be achieved by studying certain key-points, as well as a combination of poses of a person or an object. In humans, these key-points are the various joints on their bodies that include the wrists, elbows, and knees, etc. Because objects are innate, these key-points include the corners and also the other important features.

The main aim of adopting pose estimation is to track the above key-points in videos or in the photos.

As much as pose estimation is challenging to the not-so-tech-savvy, it is an aspect of the computer technology that is slowly sipping into every sector of the economy. Programmers and developers are also increasingly considering the implementation of the pose estimation into their programs.

Furthermore, many businesses are also looking to explore possibilities with the pose estimation and there are also reasons why.

IX. CONCLUSIONS

In this project we have built a model with the help of Deep Neural Network under which we have used Convolutional Neural Network. Here the model detects the yoga poses and displays the probability and name of the yoga pose which is detected.

Acknowledgement

To complete any type of project work is teamwork. It involves all the technical/nontechnical expertise from various sources. The contribution from the experts in the form of knows-how and other technical support is of its vital importance. We are indebted to our inspiring guide Prof. Pranoti Kale.

We have great pleasure in offering big thanks to our honorable Principal Prof. Dr. S. R. Patil. Last but not least, we would like to thank all the direct and indirect help provided by the staff and our entire class for successful completion of this project. We will be also thankful for allowing us to publish the paper.

REFERENCES

1. Naimat Ullah Khan and Wanggen Wan from School of Communication and Information Engineering. Institute of Smart City. Shanghai University, Shanghai, China, A Review of Human Pose Estimation from Single Image, Conference Paper · July 2018
2. Vyas, Parth, San Jose State University SJSU ScholarWorks, "POSE ESTIMATION AND ACTION RECOGNITION IN SPORTS AND FITNESS" (2019).
3. K. Jarrett, K. Kavukcuoglu, M. Ranzato, and Y. LeCun. What is the best multi-stage architecture for object recognition? In Computer Vision, 2009 IEEE 12th International Conference on, pages 2146–2153, Sept 2009.
4. N. Pinto, D. D. Cox, and J. J. DiCarlo. Why is real-world visual object recognition hard? PLoS computational biology, 4(1):e27, 2008.
5. X. Glorot, A. Bordes, and Y. Bengio. Deep sparse rectifier networks. In Proceedings of the 14th International Conference on Artificial Intelligence and Statistics. JMLR W&CP Volume, volume 15, pages 315–323, 2011.