ExNET: Deep Neural Network for Exercise Pose Detection

Chapter · July 2019

DOI: 10.1007/978-981-13-9181-1_17

Sadeka Haque
Daffodil International University
12 PUBLICATIONS **155** CITATIONS

Akm Shahariar Azad Rabby
Apurba Technology
46 PUBLICATIONS **296** CITATIONS

Monira Akter Laboni
Daffodil International University
3 PUBLICATIONS **13** CITATIONS

Nafis Neehal
Islamic University of Technology
20 PUBLICATIONS **119** CITATIONS

ExNET: Deep Neural Network for Exercise Pose Detection

Sadeka Haque(✉), AKM Shahariar Azad Rabby(✉), Monira Akter Laboni,
Nafis Neehal, and Syed Akhter Hossain

Department of Computer Science and Engineering,
Daffodil International University, Dhaka, Bangladesh
{sadeka15-5210,azad15-5424,akter15-5044,nafis.cse}@diu.edu.bd
aktarhossain@daffodilvarsity.edu.bd

Abstract. Pose detection estimate human activity in images or video frames using computer vision technique. Pose detection has many applications, such as body to augmented reality, fitness, animation etc. ExNET represents a way to detect human pose from 2D human exercises image using Convolutional Neural Network. In recent time Deep Learning based systems are making it possible to detect human exercise poses from images. We refer to the model we have built for this task as ExNET: Deep Neural Network for Exercise Pose Detection. We have evaluated our proposed model on our own dataset that contains a total of 2000 images. And those images are distributed into 5 classes as well as images are divided into training and test dataset, and obtained improved performance. We have conducted various experiments with our model on the test dataset, and finally got the best accuracy of 82.68%.

Keywords: Human pose detection · Object detection · Deep learning · Exercise Pose Detection

1 Introduction

Human pose classification problem includes the recognition of different exercise poses using the image in the computer vision community. Human pose analysis is one of the interesting issues and a challenge to understand within automated images. It detect human figures in images, so that one could determine, for example, where someone's elbow shows up in an image. Pose classification has many uses, from interactive installations that react to the body to augmented reality, animation, fitness uses, and more.

Automatically detecting the presence of a human is an impossible task depending on the image because it has a variety of conditions that have to do with scale and resolution, pose etc. Most of the algorithms proposed can detect a human pose from the image that occupies a significant portion of the image sports a familiar and benign pose, and wears clothing that contrasts with the background.

K. C. Santosh and R. S. Hegadi (Eds.): RTIP2R 2018, CCIS 1035, pp. 186–193, 2019.
https://doi.org/10.1007/978-981-13-9181-1_17

While the use of neural networks has been successful in pose detection, because of the significant pose variation exhibited by the human body, and the impossibility of training for all possible variants. CNN's are appealing for human pose classification because there's no compelling reason to unequivocally configuration highlight portrayals and identifiers for parts in light of the fact that a model and highlights are found out from the information. The presence of profound learning has diminished the measure of hand-designed handling required for PC vision by performing numerous tasks, for example, max-pooling, cluster standardization, and resampling inside Convolutional Neural Networks (CNN). Convolutional neural systems (CNN) have made exceptional progress as of late on picture grouping and item confinement issues. They are fundamentally the same as conventional neural systems as far as that they are comprised of neurons with learnable loads and predispositions. Be that as it may, neural systems don't scale well to bigger picture sets. Every neuron in a layer is completely associated with every one of the neurons in the past layer, so we rapidly produce countless and wind up overfitting on the preparation set. CNN's exploit the way that the info comprises of pictures, so they oblige the design in an increasingly sensible manner which immensely diminishes the quantity of parameters.

In this task, we investigate one single designs for demonstrating human posture identify and action grouping. This model having 5 classes of different kind of human pose images they are swiss ball hamstring curl, pull up, push Up, walking and cycling showing in Fig. 1.

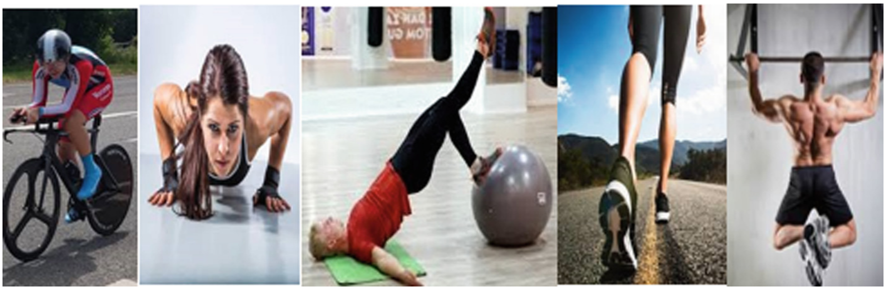

Fig. 1. Different kind of human poses

2 Literature Review

CNN's have been utilized for characterization undertakings or classification, yet they are progressively being connected towards location issues. Sermanet et al. propose an incorporated way to deal with article discovery, acknowledgment, and confinement with a solitary CNN [1]. At the abnormal state, human exercises can frequently be precisely portrayed regarding body posture, movement, and communication with scene objects [2]. However, because of the testing idea of this issue, most current movement acknowledgment models depend on all encompassing portrayals that extricate appearance and movement highlights from the video from which the pictures are pulled. Recently, Toshev et al. [3] demonstrated that applying profound CNN's to act estimation like a relapse issue has the benefit of thinking about posture in a basic however all encompassing style. This methodology accomplished (0.61) and was less complex than past strategies dependent on unequivocally planned element portrayals and graphical models. One of the initial phases toward this path has been the utilization of organized expectation over semantic division, e.g. by combining image-based DenseCRF [4] inference with CNN's for semantic segmentation [5], training both systems jointly [6], or more recently learning CNN-based pairwise terms in structured prediction modules [7]. All of these works involve coupling decisions so as to reach some consistency in the labeling of global structures, typically coming in the form of smoothness constraints.

3 Proposed Methodology

3.1 Dataset

The dataset we used for this model is collected from the publicly available sources. This dataset contains 2000 images of human exercise poses divided into five class labeled as pull up, push up, walking, swiss ball hamstring curl, cycling. Each class contains 400 images. Initially, the images had different dimension and different color and background. Figure 2 showing some example of human exercise pose.

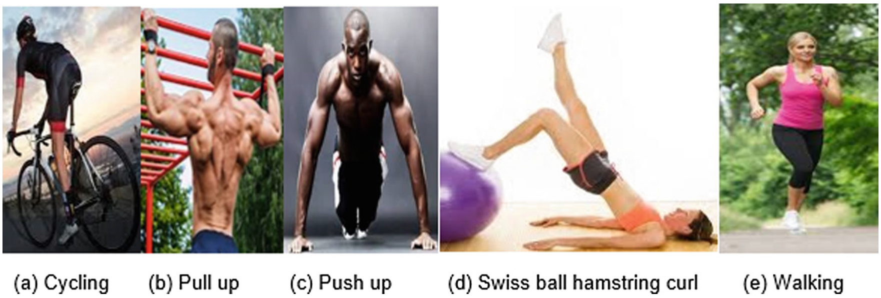

Fig. 2. Exercise pose dataset

3.2 Dataset Preparation

The pose dataset contains images with various size and shape. While CNN required the fix input size. Also, the image has lots of unnecessary information. So, at first, we manually cropped all the images to reduce all the unnecessary information. Later resized all images into 32 × 32 pixel to get highest accuracy within lowest computation cost.

Deep learning system performs better in the event that it discovers more information. Consequently, information increase creates more data artificially. To augment the data, we choose 6 methods. First, we randomly 40° rotate the images, then width and height shift the images. Later normalize the images, zoom and shear the images with a horizontal flip.

When the data augmentation done, the Leeds dataset consists of about 2000 images and then it was split 80% training and 20% validation images.

3.3 Architecture of the Model

Our proposed model ExNet is a multilayer convolutional network begins with the input size of 64 × 64 grayscale image connected with a convolutional layer which has 32 filters, 5 × 5 kernel size, and swish (1) activation [10]. The next layer is also a convolutional layer with 32 filter size and 3 × 3 kernel. A max pool layer added followed by this two-layer with a pool size of 2. Also, we added a 25% dropout [9] layer to avoid overfitting. We also added batch normalization [8] to avoid overfitting after a dropout and a convolutional layer.

$$\sigma(z) = (1 + \exp(-z)) - 1 \tag{1}$$

Later we added three convolutional layers one after another with the same parameters of a 3 × 3 kernel, 64 filters with swish activation. The output is connected with a max pool layer with 25% dropout. After that we conquered the layer and connected it to a fully connected dense layer with 512 hidden units with 50% dropout. The final output layer has 5 output node. Final output unites usage the softmax (2) activation to predice the output of the model.

$$\sigma(z)_j = \frac{\mathrm{e}^{z_j}}{\sum_{k=1}^{k} \mathrm{e}^{\mathrm{z}_\mathrm{k}}} for j = 1, ...k \tag{2}$$

Figure 3 Showing the proposed model architecture.

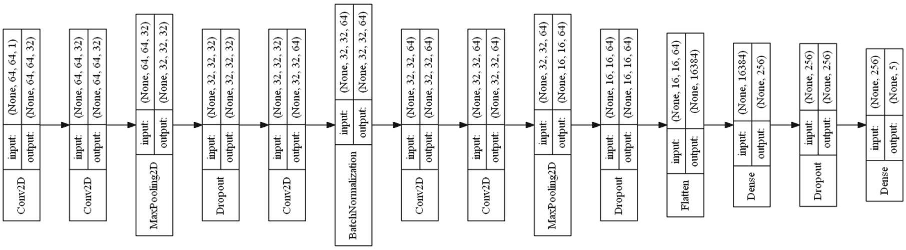
Conv2D input: (None, 64, 64, 1) output: (None, 64, 64, 32)
Conv2D input: (None, 64, 64, 32) output: (None, 64, 64, 32)
MaxPooling2D input: (None, 64, 64, 32) output: (None, 32, 32, 32)
Dropout input: (None, 32, 32, 32) output: (None, 32, 32, 32)
Conv2D input: (None, 32, 32, 32) output: (None, 32, 32, 64)
BatchNormalization input: (None, 32, 32, 64) output: (None, 32, 32, 64)
Conv2D input: (None, 32, 32, 64) output: (None, 32, 32, 64)
Conv2D input: (None, 32, 32, 64) output: (None, 32, 32, 64)
MaxPooling2D input: (None, 32, 32, 64) output: (None, 16, 16, 64)
Dropout input: (None, 16, 16, 64) output: (None, 16, 16, 64)
Flatten input: (None, 16, 16, 64) output: (None, 16384)
Dense input: (None, 16384) output: (None, 256)
Dropout input: (None, 256) output: (None, 256)
Dense input: (None, 256) output: (None, 5)

Fig. 3. Model architecture

3.4 Optimizer and Learning Rate

Optimizer plays an important role in the deep neural network. Optimizer algorithm helps us to minimize or maximize the error function of the model which depends on the model hyper-parameter which is used to compute the class label base on the model input. The hyper-parameter of a model is important to efficiently and effectively train the model and produce the accurate result. For ExNET we used the Adam optimizer. Adam stands for Adaptive Moment Estimation which computes adaptive learning rates for hyper-parameter. The Adam optimization algorithm is easy when to implement and it is computationally efficient. It requires little memory which is why it is used in most of the computer vision and NLP applications. Our given model ExNET used ADAM (3) Optimizer [11] with a learning rate of 0.001.

$$\theta_{t+1} = \theta_t - \frac{\eta}{\sqrt[1]{\widehat{v}} + \epsilon}\widehat{m} \tag{3}$$

When we use a neural network for performing classification and prediction tasks we have to calculate the error rate to find the model performance. One of the recent studies show that the cross entropy function gives better performance than classification error and mean square error [12]. Proposed method used categorical cross entropy (4) as loss function.

$$L_i = -\sum_j t_{i,j} \log(p_{i,j}) \tag{4}$$

To make the optimizer converge faster and closer to the global, we used an automatic Learning Rate reduction method [13]. Through the minimum loss, learning rate is the step size of walks. When the learning rate is low, it takes much time to get the global minima. On the other hand when the learning rate is high, the training can not get converged or even diverged. To get the fast computation time we set a maximum learning rate that can be automatically decreased with the monitoring of validation accuracy.

4 Training the Model

The ExNET is trained on human exercise pose dataset where 80% used as a training set and other 20% in validation set. After 50 epochs model performed accurately on classifying human exercise pose. The automatic learning rate reduction formula helps the Adam optimizer to converge faster by monitoring the validation accuracy and reduced the learning rate from 0.001 to 0.000001.

5 Evaluating the Model

The ExNET is trained on human exercise pose images and give a promising result on train and validation set.

5.1 Train Validation Split

To examine how the model performs, train and validation set has been created. The training data set is used to train the model with known output and validation data used to check how the model performs during training time and help to measure its performance and tune the hyper meters.

The human exercise pose dataset has a total of 2000 images in five class, while each class contains 400 images each. To create the train set we use 1600 images (80%) and 400 images (20%) on the validation set.

5.2 Model Performance

After 50 epochs proposed ExNET got 82.68% accuracy on classifying 2D human exercise pose from the dataset.

Analyzing the result and confusion matrix, we found that the proposed model performs so well in classifying the human exercise pose.

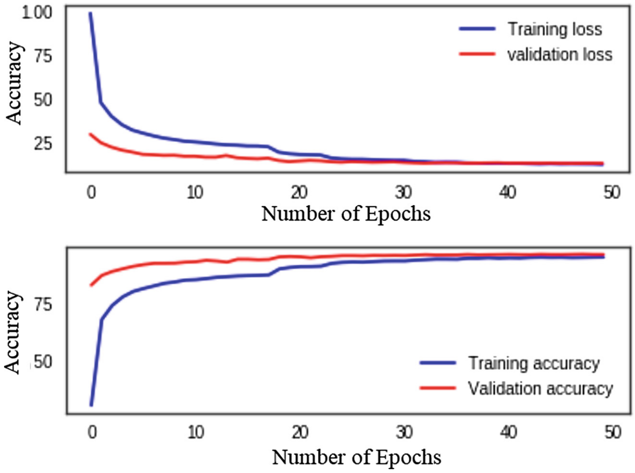

Fig. 4. Accuracy and loss for ExNet

Many exercise images were looking so similar for exercise equipment, lighting condition, and similar human pose. For those case model also did pretty good. Figure 4 showing the loss and accuracy of ExNET and Fig. 5 shows the confusion matrix for the proposed ExNET.

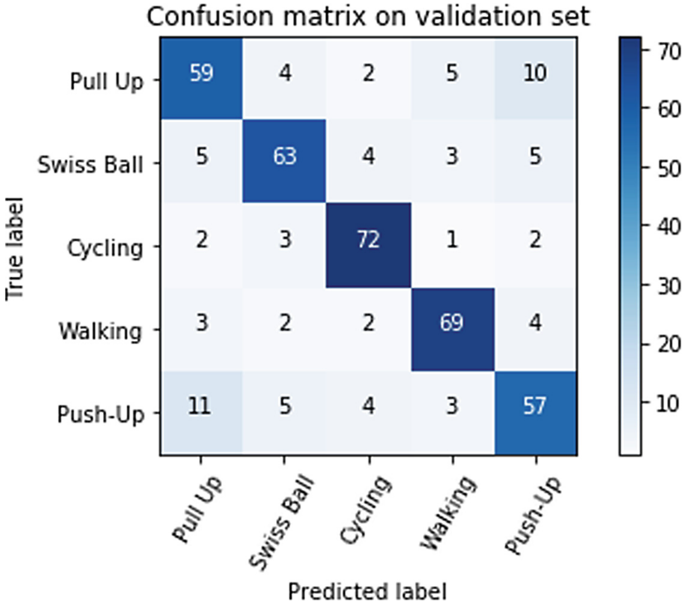

Fig. 5. Confusion matrix for ExNet

6 Conclusion and Future Work

This proposed model presenting a better performance of classification of human poses exercise. The dataset for both train and validation set for lesser epochs and less computation time compared to the other CNN model and achieve 82% accuracy on our dataset. As a result, we are able to achieve state-of-art results on several challenging exercise pose datasets.

Sometimes the proposed model confused to understand some pose due to overfitting. Future work increasing the dataset size will help to perform the model better. To diminish the gap among training and validation execution further tweak the hyper parameters of our model.

References

1. Sermanet, P., Eigen, D., Zhang, X., Mathieu, M., Fergus, R., LeCun, Y.: OverFeat: integrated recognition, localization, and detection using convolutional networks. arXiv preprint arXiv:1312.6229
2. Pishchulin, L., Andriluka, M., Gehler, P., Schiele, B.: Poselet conditioned pictorial structures. In: 2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 588–595, Berlin, Heidelberg, June 2012, 2013
3. Toshev, A., Szegedy, C.: DeepPose: human pose estimation via deep neural networks, pp. 1653–1660 (2014)
4. Krähenbühl, P., Koltun, V.: Parameter learning and convergent inference for dense random fields. In: ICML (2013)

5. Chen, L., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A.L.: Semantic image segmentation with deep convolutional nets and fully connected CRFs. In: ICLR (2015)
6. Zheng, S., et al.: Conditional random fields as recurrent neural networks. In: ICCV (2015)
7. Chandra, S., Kokkinos, I.: Fast, exact and multi-scale inference for semantic image segmentation with deep Gaussian CRFs. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9911, pp. 402–418. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46478-7_25
8. Ioffe, S., Szegedy, C.: Batch normalization: accelerating deep network training by reducing internal covariate shift. arXiv:1502.03167 [cs]
9. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. **15**, 1929–1958 (2014)
10. Neural and Evolutionary Computing (cs.NE): Computer Vision and Pattern Recognition (cs.CV); Learning (cs.LG) arXiv:1710.05941 [cs.NE]
11. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv:1412.6980 [cs.LG], December 2014
12. Janocha, K., Czarnecki, W.M.: On loss functions for deep neural networks in classification. ArXiv, abs/1702.05659 (2017)
13. Schaul, T., Zhang, S., LeCun, Y.: No more pesky learning rates. arXiv preprint arXiv:1206.1106 (2012)