DEEP LEARNING FOR ACTION CLASSIFICATION USING BHARATHNATYAM DATASET

B. Gnana Priya

Assistant Professor, Department of Computer Science and Engineering, Annamalai University

Dr. M. Arulselvi

Assistant Professor, Department of Computer Science and Engineering, Annamalai University

ABSTRACT

Human activity recognition is a classification task for recognizing the action performed by human. Pose estimation forms the base of action recognition. Recently, deep learning have been deployed for pose estimation and action classification. The advantage here is that there is no need to explicitly design feature representations and detectors for parts because model and features are learned from the data. Most of the human pose algorithms make use of the benchmark datasets available. Here a new challenging dataset of a popular Bharathnatyam is taken. The dataset contains 10 different poses. We have captured images and try to classify them using deep convolutional neural network.

KEYWORDS**: **Action classification, Bharathnatyam dataset, Deep learning, Convolutional neural network.

INTRODUCTION

Human pose estimation has a wide range of application in many areas such as surveillance, assisted living, human computer interface, activity recognition, image indexing and retrieval and so on. Human pose is very important for recognizing the actions. The reason behind understanding human appearance and related aspects is to analyze humans and their interaction with surroundings which are essential and are in demand for industrial applications. Information such as posture, gesture, outlook etc plays great importance for business. Determining the human body joint locations and configuration is quizzed as human pose estimation problem.

Bharatanatyam, a pre-eminent Indian classical dance form presumably the oldest classical dance heritage of India is regarded as mother of many other Indian classical dance forms. It is a classical Indian dance form that is popular and nurtured in the Indian state of Tamil Nadu. Bharatanatyam consist of 64 principles of co-ordinated hand, foot, face and body movements which are performed to the accompaniment of dance syllables. The dance steps are based on the dancers balanced body weight distribution, firm position of lower limbs , pretty hand movements that flow around their body. The poses are very different from our normal day to day actions.

CNN are machines that combine convolutional operations using learnable filters and nonlinear activation functions for classification in hierarchical structures. The input are mapped into compact representation and are separated into classes for classification depending on the objective function. CNN extract complex and abstract features from different parts of the input by stacking and down sampling them. Three types of layers are used here: Convolutional Layer**,** Pooling Layer and Fully-Connected Layer. Convolutional layer forms the basic building block and uses kernels to detect features all over the image.

The Kernels carries out a convolution operation which is an element-wise product and sum between two matrices. Pooling layers are inserted between convolutional layers to reduce the parameters and computation in the network. It resizes the input and prevent over fitting of network.

RELATED WORKS

Human pose estimation is the most popular research area of the past decade. One straightforward solution for this problem is to use the whole image to represent a pose and treat pose recognition as a general image classification problem. Such methods have achieved promising performance [2,6,7]. Convolutional Pose Machine [9] incorporated the inference of the spatial correlations among body parts within the ConvNets. There are two widely used methods for pose detection. In the detection based approach we use heat map to find the detection score of the points[10].They do not provide the co-ordinates of joints directly. The second being Regression based approach which uses a nonlinear function that maps the input directly to the desired output [8]. Hourglass Network [16] proposed a state-of-the-art architecture for bottom-up and top-down inference with residual blocks. Tompson[17] used multiple branches of convolutional networks to fuse the features from an image pyramid, and used Markov Random Field for post-processing. Chen and Yuille [18] introduced the ConvNet to learn both the unary and the pair wise term of a tree structured graphical model. Many of the previous works usually use manually designed multi context representations, e.g., multiple bounding boxes [19] or multiple image crops [20], and hence lack of flexibility and diversity for modelling the multi-context representations.

Existing benchmarks datasets includes aspects of the human pose estimation such as sport scenes, frontal-facing people, people interacting with objects, pose estimation in group photos and pose estimation of people performing synchronized activities. Few of the challenges in predicting human pose coordinates includes the foreshortening of limbs, occlusion of limbs, rotation and orientation of the figure, and overlap of multiple subjects. Although CNN gives the highest classification scores, it requires many number of parameters to be trained, a bulge amount of memory and abundant still images for training.

PROPOSED WORK

We formulate the action classification problem as a multi-class classification problem that can be modelled by a convolutional neural network. The CNN takes as input a image of size 200 X 200 pixels and outputs a vector of numbers representing the probabilities of each of the activity labels corresponding to the 10 categories. There is no need to explicitly design feature representations and detectors for parts because model and features are learned from the data.

DATA SET

The data was recorded in an studio environment with constant background. The data was captured using three cameras placed around the capture space.. Twenty different persons performed the same actions and are captured. The poses are from different parts of the dance sequence performed in Bharathnatyam. Around 700 images are captured. We have chosen 650 images from the original captured image which are clear enough for processing. Each pose is captured in different angle so that our system can recognize foreshortening of limbs, occlusion of limbs, rotation and orientation of the figure. Data augmentation is done to increase the number of images needed for training. The captured images are rotated to different angles from the original angle. We got around 1050 images after augmentation. Each of our image contains only single subject and are centred in the image.

The various steps carried out are

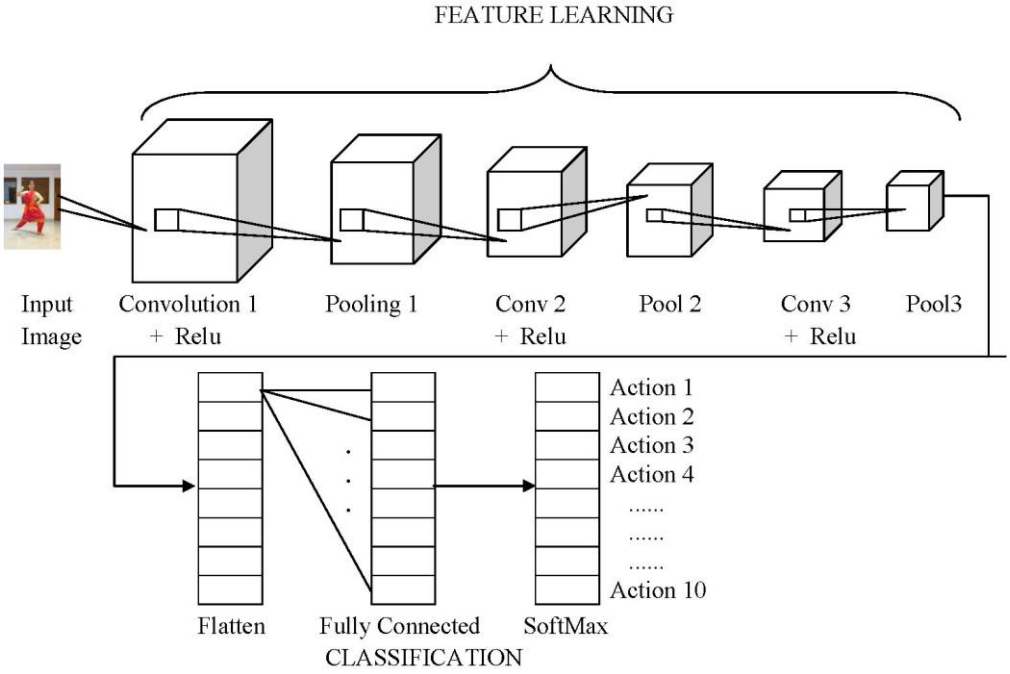
- Loading and pre-processing the Dataset.
- Modelling the Convolutional neural network in Keras.
- Training the CNN for multiclass classification.
- Evaluating the model and predicting the output class of a test image.
- Finding the accuracy and loss.

Action Category	No. Of Actions	No. Of Images
BHARATHANATYAM	1. Action 1	110
	2. Action 2	105
	3. Action 3	105
	4. Action 4	110
	5. Action 5	110
	6. Action 6	105
	7. Action 7	105
	8. Action 8	100
	9. Action 9	100
	10. Action 10	100
Total		1050

Table 1: Number of images in the Bharathanatyam Dataset

Fig 1: Sample images from the Bharathanatyam Dataset

Fig 2: Feature Extraction and Classification Flow Graph



NETWORK ARCHITECTURE

We use Keras API written in python. Keras specially designed for neural network running on top of TensorFlow. It allows us to built networks easily , extend them and add new modules in a simple manner.

We use the Sequential model for building our network. The desired layer can be added one by one in the Sequential model. The Dense layer(fully connected layer) is used to build a feedforward network in which all the neurons from one layer are connected to the neurons in the previous layer. ReLU activation function is required to give non-linearity to the model. This will make the network to learn non linear decision boundaries. As our problem is a multiclass classification problem we use the softMax layer as the final layer.

In order to configure the network we use SGD(Stochastic Gradient Descent) optimizer. The loss type used here is categorical cross entropy which is used for multiclass classification. The accuracy and loss are the metrics we are tracking during the training process. The training of the network is done using the fit() function in Keras. The number of epochs for training is 300. We use dropout layer to prevent overfitting as the parameters of the network getting biased towards training data. The dropout layer will randomly turn off few neurons during our training, so that the dependency of training set may get reduced.

The training of the network is done using the fit() function in Keras. The number of epochs for training is 300. We use dropout layer to prevent overfitting as the parameters of the network getting biased towards training data. The dropout layer will randomly turn off few neurons during our training, so that the dependency of training set may get reduced. The Flow graph of the network is shown in Fig(2). A sample of bharathanatyam images present in our dataset is given in Fig(1).

Collecting large amount of data requires more effort and time. Training a large network is very expensive and needs us to train it recursively changing various parameters. So, we apply the popular Transfer learning strategy wherein we use the weights of pre-trained network

trained on a large dataset applied to a different dataset. We also use Fine tuning to train the last few layers of the pre-trained network on the new Bharathanatyam dataset to adjust the weight.

EVALUATION OF CLASSIFICATION

We need to structure our training and validation datasets for that we are going to create a directory structure. Images of each class are put in separate sub-directory in the training and validation directories. We are going to use the VGG16 model with weights pre-trained on ImageNet. To save the bottleneck features from VGG16 model we include a function save_bottlebeck_features() .We take 800 images for training and 250 images for validation. Initially we obtain a accuracy of 50% since our dataset is trained for the first time. On fine tuning various parameters we finally got a accuracy of 65%. Since we have only small number of images it is not sufficient to train our network. The future work will be to improve the size of our dataset and images need to be augmented to increase the size.

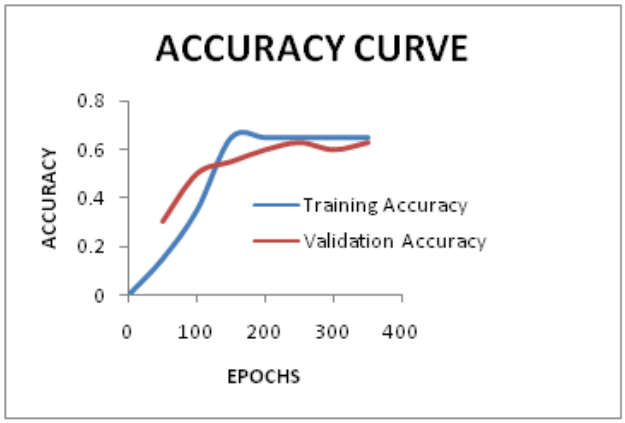


Fig 3: Training and Validation accuracy for 300 epochs

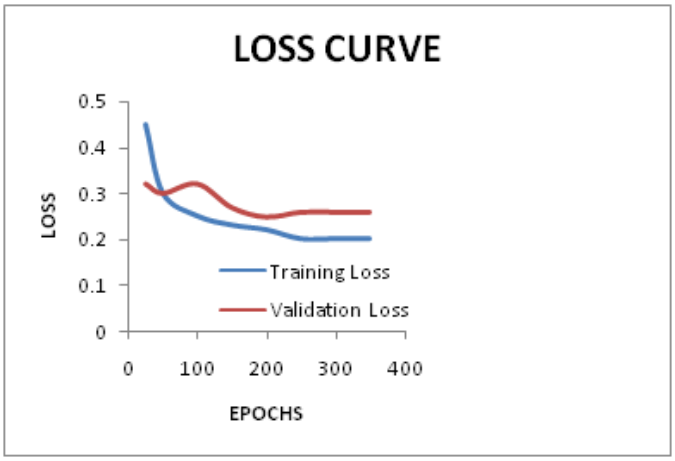


Fig 4: Training and Validation loss for 300 epochs

CONCLUSION

Convolutional neural networks are the most preferred architecture for machine learning and computer vision task due to their simplicity and reduced number of parameters. They are successfully applied in various image classification tasks. In this work action classification of

few Bharathanatyam moves from the original data captured are carried out based on deep learning algorithm using Keras running in top of Tensorflow. The proposed work classifies the pose with an accuracy of 65% .In future, this work will be extended for different martial arts, sports and dancing datasets. The aim is to build a multi-view dataset for various poses. Also, the number of poses can be increased and the network can be trained for classifying many diverse poses.

REFERENCES

[1] https://en.wikipedia.org/wiki/Bharatanatyam

[2] E. Insafutdinov, L. Pishchulin, B. Andres, M. Andriluka, and B. Schiele. Deeper Cut: A Deeper, Stronger, and Faster Multi-Person Pose Estimation Model. In European Conference on Computer Vision(ECCV), May 2016.

[3]I. Lifshitz, E. Fetaya, and S. Ullman. Human Pose stimationUsingDeepConsensusVoting,pages246–260.Springer International Publishing, Cham, 2016

[4] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele. 2d human pose estimation: New benchmark and state of the art analysis. In IEEE Conference on Computer Vision and Pattern Recognition(CVPR),June 2014

[5]U. Rafi, I. Kostrikov, J. Gall, and B. Leibe. An efficient convolutional network for human pose estimation. In BMVC, volume 1, page 2, 2016.]

[6] G.Ning ,Z.Zhang ,and Z.He. Knowledge-guided deep fractal neural networks for human pose estimation. IEEE Transactions on Multimedia, PP(99):1–1, 2017.

[7] T. Pfister, K. Simonyan, J. Charles, and A. Zisserman. Deep convolutional neural networks for efficient pose estimation in gesture videos. In Asian Conference on Computer Vision (ACCV), 2014.

[8] A. Toshev and C. Szegedy. Deep Pose: Human Pose Estimation via Deep Neural Networks. In Compute Vision and Pattern Recognition(CVPR), pages 1653–1660, 2014.

[9] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh. Convolutional pose machines. In IEEE Conference on Computer Vision and Pattern Recognition(CVPR), 2016.

[10] A. Bulat and G. Tzimiropoulos. Human pose estimation via Convolutional Part Heat map Regression. In European Conference on Computer Vision (ECCV), pages 717–732, 2016.

[11] C.Cao, Y.Zhang, C.Zhang, and H.Lu. Body joint guided 3d Deep convolutional descriptors for action recognition. CoRR, abs/1704.07160, 2017.

[12] Y. Chen, C. Shen, X. Wei, L. Liu, and J. Yang. Adversarial pose-net: A structure -aware convolutional network for human pose estimation. CoRR, abs/1705.00389, 2017.

[13] G. Cheron, I. Laptev, and C. Schmid. P-CNN: Pose-based CNN Features for Action Recognition. In ICCV, 2015.

[14] C.-H. Chen and D. Ramanan. 3d human pose estimation = 2d pose estimation + matching. In The IEEE Conference on Computer Vision and Pattern Recognition (CVPR), July 2017.

[15] S. Herath, M. Harandi, and F. Porikli. Going deeper into action recognition: A survey. Image and Vision Computing, 60(Supplement C):4 – 21, 2017. Regularization Techniques for High-Dimensional Data Analysis.

[16] A. Newell, K. Yang, and J. Deng. Stacked hourglass networks for human pose estimation. In Proc. Eur. Conf. Comp. Vis., pages 483–499, 2016.

[17] J. Tompson, R. Goroshin, A. Jain, Y. LeCun, and C. Bregler. Efficient object localization using convolutional networks. In Proc. IEEE Conf. Comp. Vis. Patt. Recogn., pages 648–656, 2015.

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

[19] V. Ramakrishna, D. Munoz, M. Hebert, J. A. Bagnell, and Y. Sheikh. Pose machines: Articulated pose estimation via inference machines. In ECCV. 2014.

[20] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In NIPS, 2012.

[21] Karen Simonyan, Andrew Zisserman ,Very Deep Convolutional Networks for Large-Scale Image Recognition In Computer Vision and Pattern Recognition (2014)