A 2-Stage approach to Human Activity Recognition

Undergraduate Thesis

By

Swapnil Jayant Kumar
160100022

Under the guidance of

Professor Asim Tewari

Department of Mechanical Engineering,
Indian Institute of Technology, Bombay

18thJune, 2020

Contents

1 Introduction

Humans possess an impressive ability to recognize the actions from the visual information conveyed by the body movements and postures. Apart from recognizing complex activities even those involving groups and object interactions, we can also appreciate the subjects' intent behind them. This enables us to even predict the actions quite accurately in the short term. Developing intelligent agents that can perform recognition tasks on par with the human brain remains a highly sought-after goal. The activity recognition technologies find applications in sectors like interactive healthcare, autonomous surveillance systems, entertainment, gesture control, autonomous driving, and computer graphics.

The different modes used to collect data for activity recognition tasks can be broadly classified into two categories - vision-based and sensor-based. The vision-based approach includes recording videos of activity performed using a monocular or stereo camera. The sensor-based method includes collecting action data using sensors such as acceleration and velocity sensors, gyroscope, thermal sensors, etc. These sensors can be external or attached to the subject. A person usually collects data with the help of his eyes (stereo-vision), hence researchers are more inclined towards the vision-based approach. Also vision-based recognition in non-invasive and quite easy to deploy after development. Multimodal collection techniques are also used which involve both video and sensor data. Despite huge leaps in recent times, the task of activity recognition remains difficult, due to several major challenges. People perform the same action with different styles, which leads to considerable intra-class variations that confuse the model. Actions can also be observed from different viewpoints like the front, side, top, and back. Some human actions are quite similar with respect to joint movements. For example, walking and jogging are similar considering the joint movements, they vary on speed and intensity. The actions are usually performed in a cluttered environment instead of a controlled one, which leads to many background noises (a person walking on a busy street). Other factors like camera movements, viewpoint changes, and illuminations also add to the problem of noise. Occlusion too is a fundamental issue in which some joints are hidden by other body parts or background objects.

Classical approaches use handcrafted features (holistic and local) to represent and classify the actions. They require expertise and do not generalize well. In recent times, the problem has been addressed by deep learning-based methods. Instead of relying on a predetermined set of features, these models learn the features from the data. Due to the success of deep-networks, there have been great improvements in the state-of-the-art accuracy of recognition systems. But most of these models are resource hungry and specific to a certain application. Hence, the objective of this project is to develop a modular, versatile, and computationally efficient human activity recognition system. The rest of this report is structured in the following manner. Section 2 gives a review of the work done in the field of activity recognition. Section 3 explains the methodology and theory behind the new 2-Stage approach. Implementation details of the approach are available in section 4. Section 5 contains experimental results on the model and its possible application. Section 6 gives the summary and final comments on the project.

2 Literature Review

Traditionally, the action recognition consists of two components (i) representation and (ii) classification. The action representation is used to extract discriminative information of human action and convert it into a feature vector. Classical methods used hand-crafted features to represent actions. [2] used a silhouette based method to construct wholistic global features like Motion Energy Image (MEI) and Motion History Image (MHI) to encode the bodily motions in a single image. [17] extended [2] to 3D creating Motion History Volume (MHV), thus eliminating viewpoint dependency of the representation. [1, 19] represented the human actions using shape features extracted from the input space-time volume. [15] computed the motion information directly from videos using optical flow, by looking at the movements in consecutive frames. Many other works preferred local handcrafted features instead of global ones. Kong *et al.* provides extensive details of these in their survey [10].

With the advent of deep learning, the problem has been addressed using deep-networks, which had lead to substantial improvements in performance. Researchers used convolution operation to extract features from the video frames. [9] implemented a 2D CNN to extract features from each frame. However, many preferred 3D convolution [16, 8] due to its inherent capability to capture temporal dynamics of motion, which was lacking from 2D CNNs. [8] achieved state-of-the-art results on KTH and TRECVID dataset using 3D CNN based model. Later [16] extended [8] to other large-scale datasets. But despite these advances it has been noted in the survey by Kong *et al.* [10] that, 3D CNNs are difficult to train due to a large number of parameters and they also don't enjoy the use of pre-trained architectures like 2D CNNs. To address this issue, many researchers used multi-stream 2D CNNs for feature extraction [14], i.e. separate 2D CNNs for spatial and temporal dimensions. [6] generated an input image of the action from gyroscopic sensor data (instead of standard RGB videos) and used 2D convolutions on them to extract features. Recently, hybrid networks are found to be highly effective in performing the activity recognition task. [5] added LSTMs layers on top of the 2D CNNs to model time series of the frames features generated by CNN. [18, 13] modeled the human action using a skeleton-based spatio-temporal graph. Convolution operations on these graphs are used to extract the high-level features for final classification. [20] combined the skeletal and the CNN based approach. They used features from both skeleton modality [18, 13] and spatio-temporal regions of interest of RGB modality, for prediction.

Most of the work discussed earlier, focuses on developing deep networks which are end-to-end trainable. Very deep networks are resource-hungry and hard to train. As the networks are trained end-to-end, hence implementing them on a different dataset requires retraining or leads to even deeper networks if transfer learning is used. On the other hand, the objective of this work is to develop a modular and efficient method for action recognition. A new 2-Stage approach is proposed which has a pre-processing module and a classifier module. Due to the shallow architecture of the classifier, it can easily be trained on new datasets and is computationally efficient compared to other deep-networks. The methodology and implementation of the approach are discussed in later sections.

3 Methodology

The objective of this project is to develop a deep learning model that is better or on-par with the start-of-the-art models, less computationally intensive, modular, and highly versatile (not domain-specific). The input data is provided in vision-based mode in the form of monocular RGB videos. The source code for this project is available at https://github.com/Machine-intelligence-program/Human_Activity_Recognition_2-Stage.git

Unlike the previous approaches to address the problem in a single step, a new 2-Stage approach is adopted. Therein, the problem is broken down into two stages - (i) Localization and (ii) Classification. The localization stage involves obtaining positions of crucial joints that define the pose of a person using a Human Pose Estimation algorithm. These joint positions are then used to evaluate the respective joint angles. The position and angle of the joints represent the spatial information, while their variations with time (displacements) represent the temporal information of the motion. In the second stage, the joints' data (displacements and angles) are fed to a spatio-temporal encoder. This encoder creates a feature vector by looking at these representational values. Figure 1 pictorially depicts the 2-stage process.

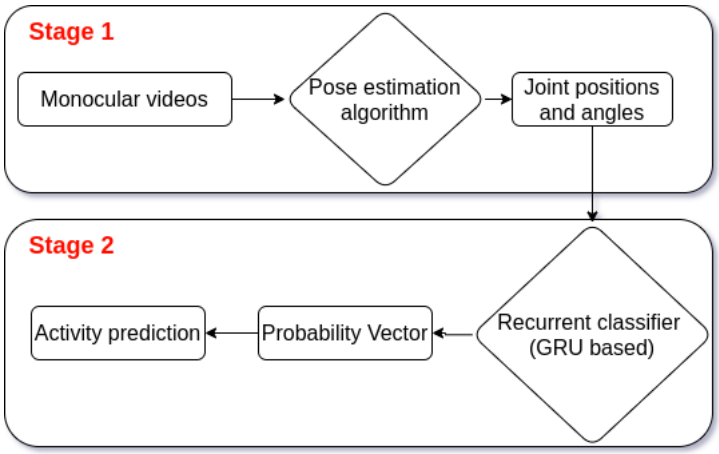


Figure 1: Flow diagram describing the 2-Stage approach.

3.1 Stage 1: Localization

In the localization stage (Stage-1), the positions of the crucial joints of the body are obtained by a human pose estimation algorithm. An OpenVino based variant of the OpenPose [3] model is used for this purpose. It is capable of evaluating joint positions for single or multiple persons. In the case of multiple persons, a pose-ID is assigned to each subject present in the image. The pose estimation model takes an image of pixel resolution 640×480 as input and fits a relevant skeleton of 18 joints on the person. The focus is on data of 14 crucial joints which affect human pose the most as marked in figure 2. The pose estimation model is modified to output the joint positions of each video frame in a CSV file.

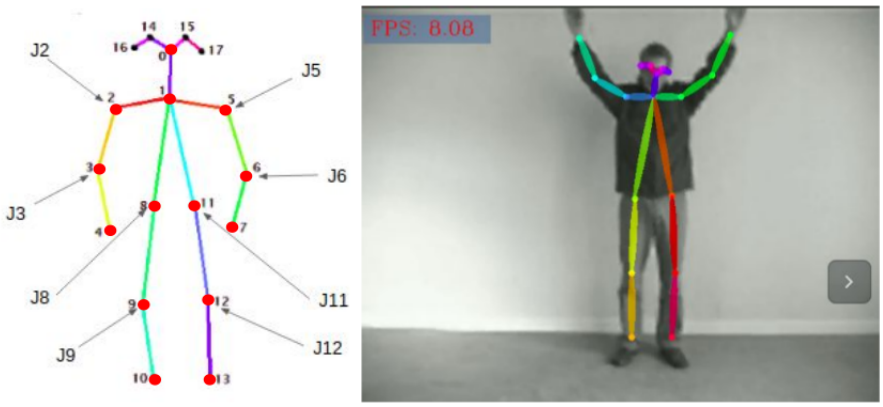


Figure 2: (a) The left side of the figure shows structure of the human skeleton being fitted. The 14 crucial joints are circled in red (background is considered to be the 18^{th} joint). Angles of the 8 pointed joint are also important. (b) The Right side of the figure shows working of the Human Pose Estimation API on a sample video frame.

3.2 Stage 2: Classification

Due to issues such as occlusion, cluttered/dark background, and camera motion joint positions are missing for some frames in the output CSV file. These missing data are imputed using interpolation. It is then used to evaluate the joint angles and change in joint positions (displacement) for each time-stamp (frame) (detailed information on data-cleaning and pre-processing are mentioned in the later sections). These joint angles and displacement are then used to classify the activity being performed, using a spatio-temporal encoder based classifier. The architecture of the classifier is shown in figure 3. ReLU is used as the non-linear activation function in this classifier.

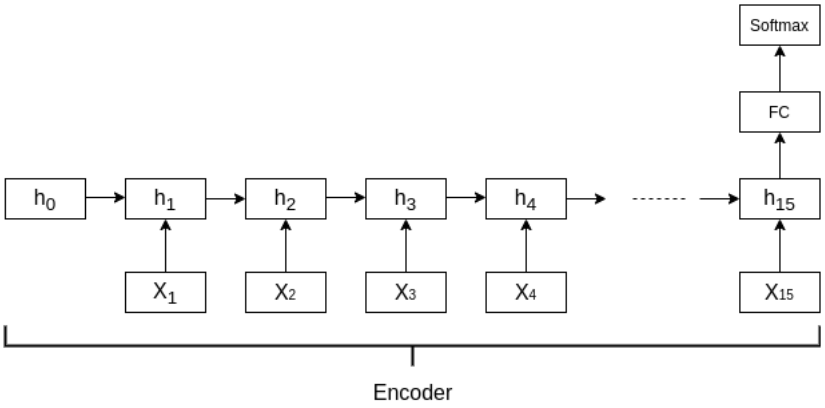


Figure 3: The architecture of the classifier used by Stage-2 of the method. The input sequence is encoded into a context vector h_{15}, which is then classified into an action using a fully-connected layer and Softmax.

$h_t|t = 0, 1, 2...$ are the hidden states at different time-steps

$x_t|t = 1, 2, 3...$ are the input vectors at different time-steps

Assuming $\mathbf{W}$ and $\mathbf{U}$ as weights of the recurrent encoder, the hidden state h_t at time t is given as.

$$h_t = \text{ReLU}(b_t + \mathbf{W}h_{t-1} + \mathbf{U}x_t)$$

The initial hidden state h_0 is taken from a normal distribution.

$$h_0 \sim \mathcal{N}(0, I)$$

The hidden state of the final time-step h_t can be considered as a context vector created by the encoder by looking at the inputs (joint angles and displacements) through the time. This context vector is then passed through a fully-connected layer and Softmax to evaluate the probability vector for different actions. Assuming $\mathbf{X}$ as weights of the fully-connected layer, this can be mathematically written as.

$$o = c + \mathbf{X}h_f$$

$$p = Softmax(o)$$

Where, h_f is the final hidden state of the encoder and also the context vector. c and o are bias and output of the fully-connected layer respectively. p is the probability vector after applying Softmax on the output o. The action which has the maximum component in p is the output label of the network.

4 Model Development

4.1 Creation of Input Sequences

As explained in section 3, the algorithm works in two stages. Stage-1 3.1 uses pose estimation to localize the crucial body joints in the action video. It saves all the 18 joint coordinates into a CSV file, of which 14 crucial joints are considered. Before proceeding to the data-processing step for creating input sequences, the missing values are imputed using interpolation. After imputation, the angles of 8 critical joints pointed in figure 2 are calculated. This is done using the vector dot product, which is shown below.

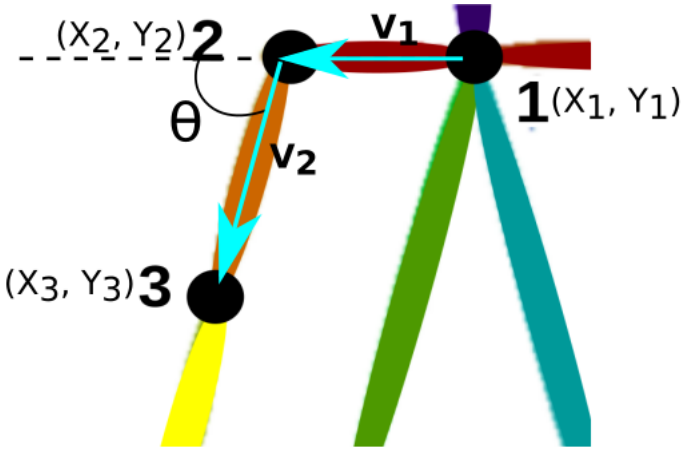


Figure 4: θ is the joint angle to be evaluated, $\vec{v_1}$ and $\vec{v_2}$ are the limb vectors.

In figure 4, the joint angle to be calcuted is marked as θ. (x_1, y_1), (x_2, y_2), and (x_3, y_3) are coordinates (in pixels) of joints 1, 2 and 3 respectively. $\vec{v_1}$ and $\vec{v_2}$ are vectors of limb l_{12} and l_{23} repectively.

$$\vec{v_1} = (x_2 - x_1)\hat{i} + (y_2 - y_1)\hat{j}$$

$$\vec{v_1} = (x_3 - x_2)\hat{i} + (y_3 - y_2)\hat{j}$$

The angle θ between the limbs is evaluated using the vector dot product as shown below.

$$\theta = \arccos\left(\frac{\vec{v_1} \cdot \vec{v_2}}{\|\vec{v_1}\|\|\vec{v_2}\|}\right)$$

$\|\cdot\|$ denotes eucledian norm and the term $\frac{\vec{v_1}\cdot\vec{v_2}}{\|\vec{v_1}\|\|\vec{v_2}\|}$ is hard clipped between 0 and 1 to prevent errors.

Apart from these 8 joint angles, change in positions (displacements) of the all the 14 joints are also considered while creating input sequences. The displacement vector of the first frame is taken as $\vec{0}$, and those of the subsequent frames are evaluated by substracting the previous frame's joint positions from the current positions. This displacement vector is then concatenated with the joint angle vector to create an input vector of length 36 for each video frame as shown in figure 5.

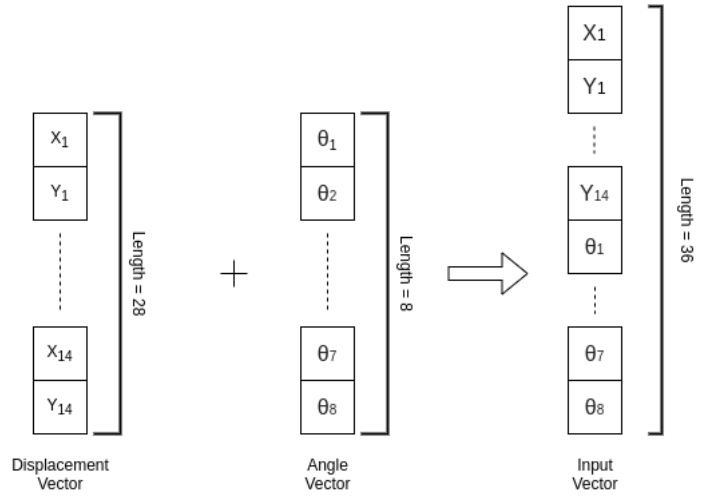


Figure 5: Concatenating displacement and angle vector to create input vector for each frame. Input vector has a size of 36, displacement vector is 28 in size (x, y components for each joint) and length of the angle vector is 8.

Input vectors of 15 consecutive frames are joined together to form an input sequence. The classifier is trained on these input sequences. To maximize the number of training examples, the sequences are created in an overlapping fashion as shown in figure 6. Hence the input sequence has a shape of $[15 \times 36]$.

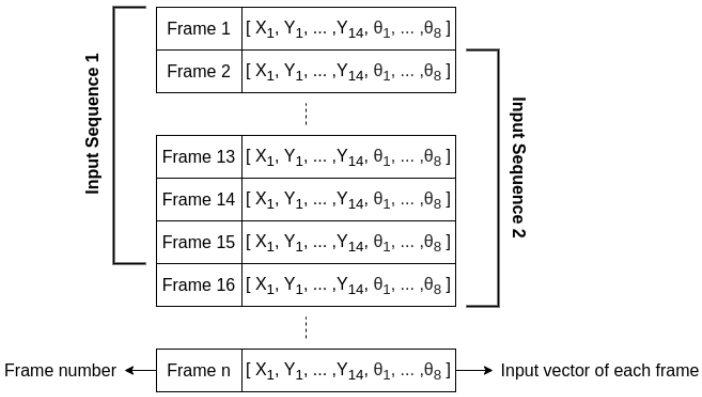


Figure 6: The figure depicts the process of making an input sequence of length 15 from a video of n frames.

4.2 Dataset

For the training and validation of the model, the KTH dataset has been used. It is one of standard and extensively used datasets in the field of human activity recognition. The dataset contains videos of six types of human actions (walking, jogging, running, boxing, handwaving, and handclapping) performed by 25 subjects in different scenarios. Variations in the background, lighting, clothing, and camera movements have been incorporated in the dataset.

4.3 Training the classifier

The architecture of the classifier network used in Stage-2 3.2 is shown in figure 3. It consists of an encoder section, which reads the input sequences and creates context vector h_{15} (final state of the encoder). This context vector is then used to classify the human action by a fully-connected layer. The sequence length of the encoder in 15 which means, it reads inputs of 15 consecutive time-stamps to create the context vector (explains the shape of input sequence). On completing the sequential data generation step as shown in figure 6, we get a total of $1,52,660$ samples. 80% of them are used to train the model, 10% are used for validation during testing and the rest is used for testing the trained model. Stochastic Gradient Descent with a batch size of 128 is used to train the classifier. The classifier is trained for 240 epochs.

The plot of training and validation error versus epochs is shown in figure 7. Due to the shallow architecture of the classifier, the training was easy. This is suggested by a nice and smooth convergence for both training and validation error in 7. The model has been tested both quantitatively and qualitatively as shown in section 5. The section also talks about the extension of the model to multi-person activity recognition and possible applications.

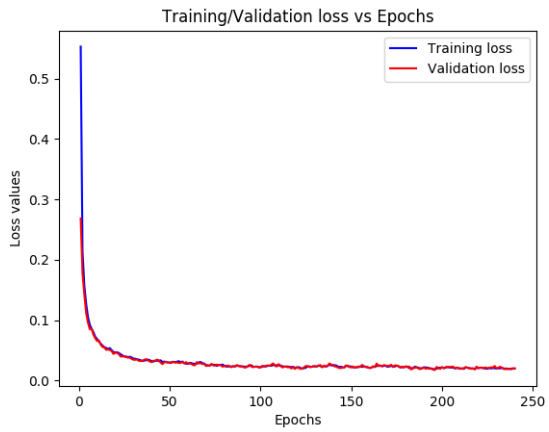


Figure 7: Training and validation loss vs. Epochs. The graph shows a good convergence for the classifier network. The training and the validation loss are almost same indicating good generalization.

5 Results and discussions

5.1 Single person activity recognition

Method	Boxing	Handclapping	Handwaving	Jogging	Running	Walking	Overall
2-Stage	99.99	99.96	99.85	94.97	92.71	91.79	**99.51**
3D-CNN [8]	90	94	97	84	79	97	90.2
Schuldt [12]	97.9	59.7	73.6	60.4	54.9	83.8	71.7
Dollar [4]	93	77	85	57	85	90	81.2
Niebles [11]	98	86	93	53	88	82	83.3
Jhuang [7]	92	98	92	85	87	96	91.7

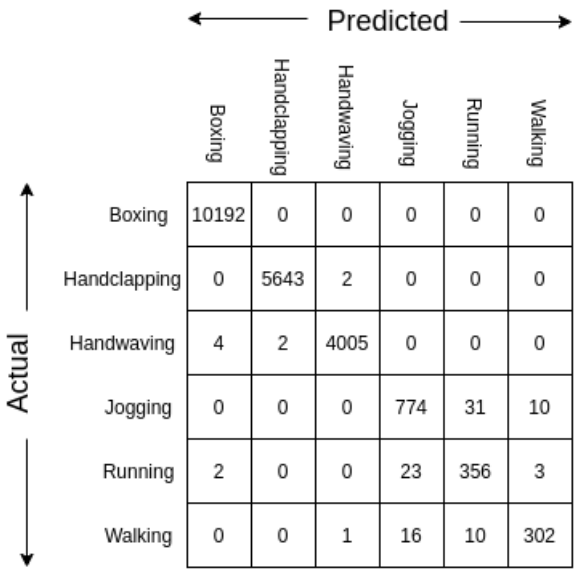
Table 1: Comparison of the experimental results of different models on KTH dataset. Results of other models are taken from S. Ji *et al.* [8]. Some of these models use different training/test split than ours.

The model was tested both quantitatively and qualitatively on the test data. Table 1 shows a quantitative comparison of the 2-Stage model's performance (in terms of accuracy) with other state-of-the-art models. The accuracy values give the percentage of correct recognition by a model for each class, the overall accuracy is weighted average on all the action. For each action, the highest accuracy value has been underlined. It is clear from table 1 that, the 2-Stage model achieves the best results in 5 activities (boxing, handwaving, handclapping, jogging, and running). For walking, the performance is on par with the best. Looking at the overall performance the 2-Stage far outperforms any other models. Also, due to the shallow architecture of the classifier, the 2-Stage model is computationally less demanding and can run on small computers if provided with processes input sequences.

Figure 8: Two frames of running are shown above, the subject goes out of range while performing the action.

A closer look at the performance of the 2-Stage model in table 1 will reveal that the accuracy of walking, jogging, and running is relatively less compared to that of boxing, handclapping, and handwaving. A possible explanation for such behavior is that videos of walking, jogging, and running have the person going out of camera range while acting. An example of this is shown in figure 8. All these frames are ignored by the algorithm which in turn, reduces the number of training samples of these actions. Due to this, the model has a somewhat biased training. This can simply be resolved by including more videos of these actions. But as the KTH dataset has the same number of videos for all the actions hence, there is a difference in the accuracies. Another explanation is given in figure 9. Looking at the confusion matrix

it is clear that the model sometimes confuses between walking, jogging, and running. This is due to the similarities in terms of the joint movements of these three activities. For example, a person running lazily can be misclassified as jogging by the model.



Actual \ Predicted	Boxing	Handclapping	Handwaving	Jogging	Running	Walking
Boxing	10192	0	0	0	0	0
Handclapping	0	5643	2	0	0	0
Handwaving	4	2	4005	0	0	0
Jogging	0	0	0	774	31	10
Running	2	0	0	23	356	3
Walking	0	0	1	16	10	302

Figure 9: Confusion matrix for the model, notice the misclassifications in walking, jogging, and running.

For the purpose of qualitative testing, the model was modified to output labels of the action on each frame. Figure 10 shows a collage of screenshots taken of the labeled output videos generated by the model. The original output videos can be found in the project repository.

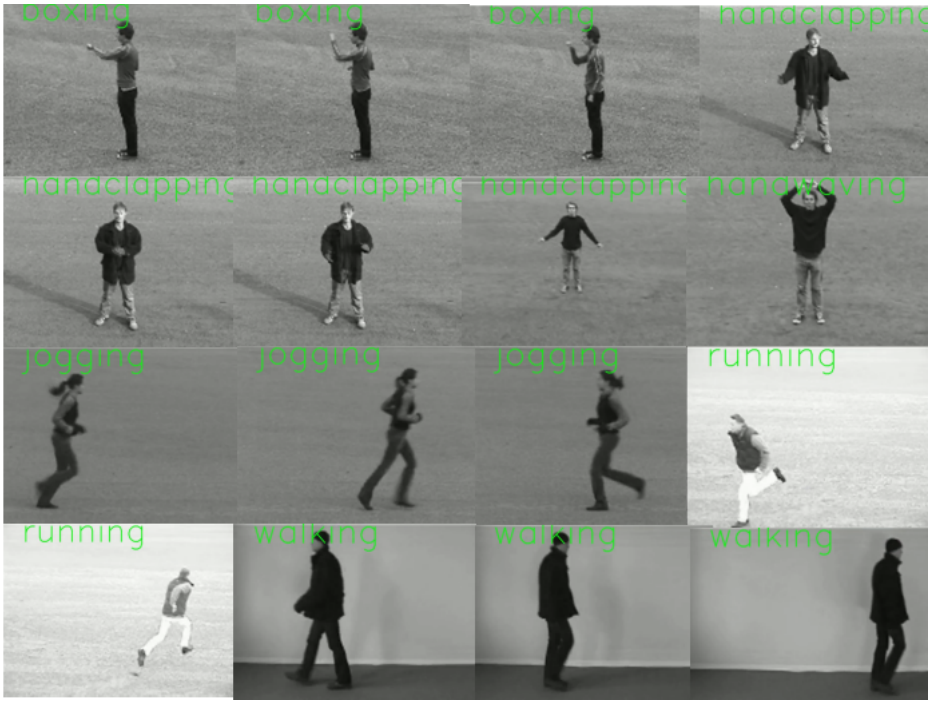


Figure 10: This figure shows a collage of the screenshots of the labeled videos obtained as output from the model.

5.2 Hyper-parameter Explorations

In this section, the effect of changing the hyper-parameters of the model on its performance is studied. Effect of two hyper-parameters namely sequence-length and activation function are explored. Various experiments are performed to observe the trends and possible reasons behind them are also discussed.

5.2.1 Effect of Sequence length

Sequence Length	Boxing	Handclapping	Handwaving	Jogging	Running	Walking	Overall
15 frames	99.99	99.96	99.85	94.97	92.71	91.79	99.51
30 frames	100	100	100	99.17	94.78	96.76	99.83

Table 2: Performance comparison of the 2-Stage model with different sequence lengths. It can be observed that the accuracy increases with increasing the sequence length but this model training takes more time.

The sequence length determines the time (number of frames) for which the classifier observes an action before classifying it. Hence inherently, a longer sequence length will result in better performance as also verified by the results in table 2. But increasing the sequence length also increases the input sequence size and classifier size. Hence, it will take a longer time to train. For the case shown in table 2, the model with a sequence length of 30 takes twice the time taken by one with a sequence length of 15. On further increasing the sequence length, the model achieves a better performance. But the computational cost will much higher compare to the gain in performance. It is also observed that 15 frames are sufficient to gather enough temporal information of the action from a 30 fps video. Figure 11 shows 15 consecutive frames of an input video. It is clear that the person almost completes a cycle of the action (boxing) in 15 frames. Including more frames will result in the inclusion of more cycles in the temporal dimension.

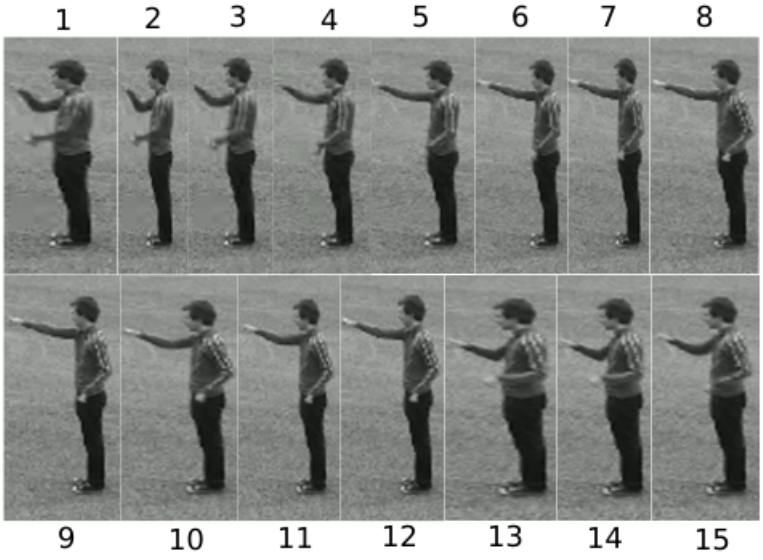


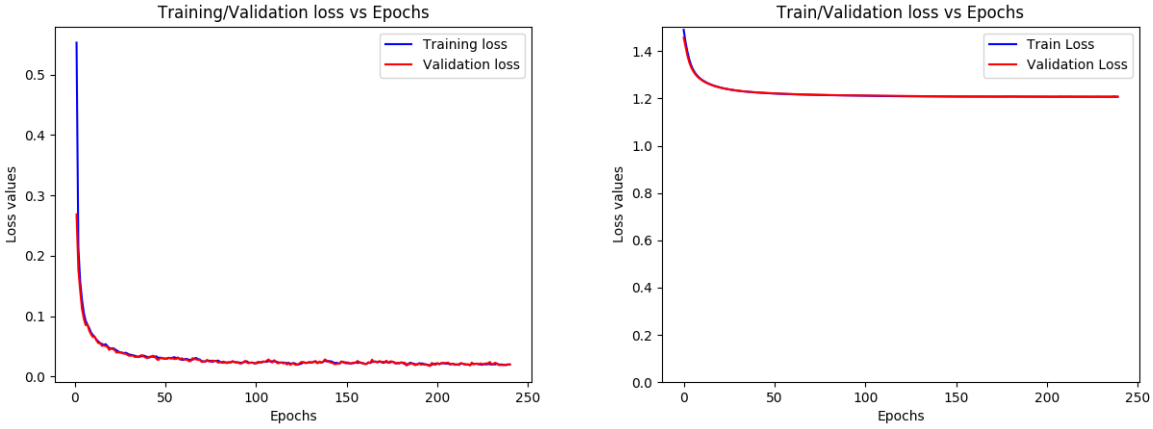
Figure 11: Consecutive frames of "Boxing" input video at 30fps. It is clear that in the time-period of 15 frames, the person completes one cycle (punching) of the action.

5.2.2 Effect of activation function

Activation function	Boxing	Handclapping	Handwaving	Jogging	Running	Walking	Overall
Relu	99.99	99.96	99.85	94.97	92.71	91.79	99.51
Leaky Relu	99.98	100	100	95.80	88.38	94.95	99.55
Sigmoid	100	2.34	10.42	0.00	0.00	0.00	50.45

Table 3: Performance comparison of the 2-Stage model with different activation functions. The model is unable to learn due to the vanishing gradient resulting from the shape of the Sigmoid activation function.

Table 3 shows performance of the model using different activation functions. The model's performance severely drops on using the Sigmoid activation. Figure 12 also confirms it, as training loss for sigmoid activation is higher indicating poor learning of the model.



(a) Relu activation function

(b) Sigmoid activation function

Figure 12: From 12a and 12b it is clear that the model with sigmoid activation has higher loss and poor learning.

One possible explanation is vanishing gradient due to the shape of sigmoid activation as shown in figure 13. The regions marked with red indicate the region of zero gradients. If the input lies in this region then the gradient is almost zero, preventing parameter updates while training. A similar result is seen on using tanh activation. On the other hand, Relu has no such issue, due its to constant gradient.

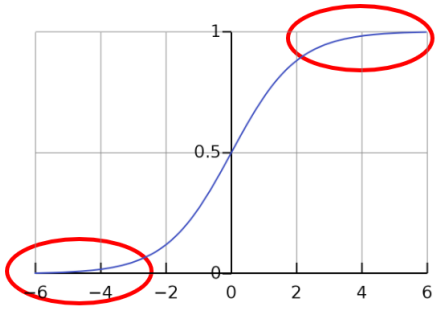


Figure 13: Sigmoid activation curve. For values greater than 4 and less than -4 the gradients are almost zero.

5.3 Multi-person activity recognition

So far, the focus was only on the activity recognition of a single person. But with little modifications in the sequence generation step 4.1, the model can be used for activity recognition for multiple subjects. Figure 14 and 15 shows how that output looks like. The models capacity is not just limited to 3 persons, it can be generalized for any number of subjects. But the accuracy decreases on increasing the number of subjects. This suggests the versatility of the model as it was trained on a single subject dataset but can work on multi-person recognition tasks with decent accuracy.

Figure 14: Model output for activity recognition involving 2 subjects. Person ID starts from the left.

Figure 15: Model output for activity recognition involving 3 subjects. Person ID starts from the left.

5.4 Possible Application - Payment system

This section describes one possible application of Human Activity Recognition, which was the original motivation for this project. Some initial work has already been done towards this application. Details of the work done and the advantages of using the 2-Stage model are discussed. In many sectors of manufacturing, there are assembly line workers who do their assigned set of tasks. They are paid based on the hours worked and mostly the payments are in cash. Hence, there is a tendency for exploitation by employers in paying them. This is one of the unthought-of areas where human activity recognition can be used. For example, this 2-Stage model can be used as an autonomous monitoring system. It will continuously monitor the work of a worker throughout the day and tell the amount to be paid based on the rates. The model needs to know the positions of the workers on the line to assign them IDs. Rudimentary work on this particular application has been done as explained below.

Assume that, there are m subjects performing a set of n different activities in their respective positions. The model can be used either in the multi-person recognition mode or single person recognition for each subject. Localization step (Stage-1) by the human pose estimation algorithm can be done in real-time, and joints position data can be saved in a CSV file for later use. At the end of the day, the classification algorithm (Stage-2) can work on the data and create an "Activit-Matrix" ($\mathbf{A}$) such that.

$$\mathbf{A} = [a_{ij}]_{(m \times n)}$$

where a_{ij} is the time for which subject i performs activity j

$$\mathbf{R} = [r_k]_{n \times 1}$$

where $\mathbf{R}$ is the "Rate Matrix" and r_k is the price for activity k in Rs/unit-time.

$$\mathbf{P} = [p_l]_{m \times 1}$$

where $\mathbf{P}$ is the "Payment Matrix" and p_l is the payment to be made to person l.

$$\mathbf{P} = \mathbf{AR}$$

Figure 16 shows output of the model modified as payment system. Amount is highlighted by blue box.

Figure 16: The terminal output of the Autonomous Payment System implementation of our model.

6 Conclusions

In this project, a new 2-Stage approach for human action recognition was proposed and developed. The model was tested on the KTH dataset and was found to be highly accurate despite having a shallow classifier. It can be easily modified and trained for different datasets due to its modular structure. It was demonstrated that the model though trained on a single action dataset can also be used for the multi-person recognition task (versatility). One possible application of the activity recognition (2-Stage model) and some rudimentary work done in that direction was also discussed.

It is worth noting that, the performance of the model depends heavily on the output of Stage-1. The OpenVino pose estimation does a tremendous job of giving good quality joint localizations. Hence, the use of a superior pose estimation algorithm will increase the model's performance. Also, the Stage-1 of the model (which uses pose estimation) is quite intensive w.r.t. computational requirement. Development of a relatively efficient system of extracting joint coordinates from videos can lead to full-scale deployment of this model on small-to-medium computers. The 2-Stage model suffers from viewpoint changes because it operates in 2D (RGB images and videos). This can be resolved in the future by extending it to the 3D domain (RGBD). Although work needs to be done in finding a way to extract joint positions in 3D as 3D pose estimation is difficult and less worked on. Once the joint positions are available, they can be processed to evaluate the respective joint displacements and angles (can be represented by Quaternions). The work done on payment system 5.4 is quite rudimentary at this stage. It will be exciting to see further developments and large scale deployments of this system.

References

[1] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri. "actions as space-time shapes". In *Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1*, volume 2, pages 1395–1402 Vol. 2, 2005.

[2] A. F. Bobick and J. W. Davis. The recognition of human movement using temporal templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(3):257–267, 2001.

[3] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. "openpose: Realtime multi-person 2d pose estimation using part affinity fields", 2018.

[4] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie. "behavior recognition via sparse spatio-temporal features". In *2005 IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance*, pages 65–72, 2005.

[5] Jeff Donahue, Lisa Anne Hendricks, Marcus Rohrbach, Subhashini Venugopalan, Sergio Guadarrama, Kate Saenko, and Trevor Darrell. "long-term recurrent convolutional networks for visual recognition and description", 2014.

[6] S. Ha and S. Choi. "convolutional neural networks for human activity recognition using multiple accelerometer and gyroscope sensors". In *2016 International Joint Conference on Neural Networks (IJCNN)*, pages 381–388, 2016.

[7] Hueihan Jhuang, Thomas Serre, Lior Wolf, and Tomaso A. Poggio. "a biologically inspired system for action recognition". *2007 IEEE 11th International Conference on Computer Vision*, pages 1–8, 2007.

[8] S. Ji, W. Xu, M. Yang, and K. Yu. "3d convolutional neural networks for human action recognition". *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(1):221–231, 2013.

[9] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. "large-scale video classification with convolutional neural networks". In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1725–1732, 2014.

[10] Yu Kong and Yun Fu. "human action recognition and prediction: A survey", 2018.

[11] Juan Carlos Niebles, Hongcheng Wang, and Fei Fei Li. "unsupervised learning of human action categories using spatial-temporal words". volume 79, pages 1249–1258, 09 2006.

[12] C. Schuldt, I. Laptev, and B. Caputo. "recognizing human actions: a local svm approach". In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, volume 3, pages 32–36 Vol.3, 2004.

[13] Chenyang Si, Wentao Chen, Wei Wang, Liang Wang, and Tieniu Tan. "an attention enhanced graph convolutional lstm network for skeleton-based action recognition", 2019.

[14] Karen Simonyan and Andrew Zisserman. "two-stream convolutional networks for action recognition in videos", 2014.

[15] D. Sun, S. Roth, and M. J. Black. "secrets of optical flow estimation and their principles". In *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2432–2439, 2010.

[16] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. "learning spatiotemporal features with 3d convolutional networks", 2014.

[17] Daniel Weinland, Remi Ronfard, and Edmond Boyer. "free viewpoint action recognition using motion history volumes". *Comput. Vis. Image Underst.*, 104, 01 2006.

[18] Sijie Yan, Yuanjun Xiong, and Dahua Lin. "spatial temporal graph convolutional networks for skeleton-based action recognition", 2018.

[19] Alper Yilmaz and Mubarak Shah. "actions sketch: A novel action representation". volume 1, pages 984 – 989 vol. 1, 07 2005.

[20] Bruce X. B. Yu, Yan Liu, and Keith C. C. Chan. "skeleton focused human activity recognition in rgb video", 2020.