An Ensemble Approach using Convolution and Recurrent Neural Networks for Human Action Recognition on the UTD-MHAD

Tan Ren Jie*, Kelvin Yeo Ngan Chong†, and Khaing Mon Kyaw‡

Institute of Systems Science, National University of Singapore
Email: e0267395@u.nus.edu*, e0267573@u.nus.edu†, e0146800@u.nus.edu‡,

***Abstract*—In this paper, we shall demonstrate an ensemble approach of performing Human Action Recognition on the University of Texas at Dallas Multimodal Human Action Dataset(UTD-MHAD), which comprises of 27 different human actions. The ensemble approach gained an accuracy of 0.821 on the validation data, which is a significant improvement to the baseline paper's accuracy of 0.672. The paper also shows the train-val performances of other models we experimented using only the Inertial and Skeleton dataset. The link to Github repository which holds to code can be found here, and the link to get the dataset can be found here.**

I. Introduction

Human Action Recognition (HAR) has been a research field that piqued interests from several computer science communities since the 1980s, due to its application in many different fields of study, such as medicine, human-computer interaction, sociology. Data in the format of videos, inertial sensors, like accelerometers and gyroscopes, depth maps and point clouds are often used to classify the different human actions [1][2].

Traditional approaches can be generally broken down into the following 3 steps:

1) Feature Extraction using Signal Processing or Computer Vision techniques to capture relevant spatio-temporal features [3][4][5]
2) Designing a pipeline to combine the extracted features
3) Training a classifier, usually a Support Vector Machine (SVM) or Random Forest (RF), using these features

This approach often requires deep domain knowledge in extracting the useful spatio-temporal features. Another approach is to use modern neural network architectures to do the feature extraction and model building automatically. Soon after 2014, there were two breakthrough papers, Single Stream and Two Stream, which ignited the research using modern approach. The main difference between them was how the model architecture combines the spatio-temporal information.

The first paper [6] uses a Single Stream Network which explores various ways to fuse temporal information from consecutive frames using 2D pre-trained convolutions. This had led to popular methods such as Long-term Recurrent Convolutional Networks (LRCN) [7], 3-D Convolutional Networks [8], and Conv3D with Attention [9].

The second paper uses a Two Stream Network [10], as shown in Fig. 1, one stream captures the spatial context (pre-trained), while the other captures the motion context. Other variants are soon developed as like the Two-Stream Network Fusion [11], Temporal Segment Networks (TSN) [12], Action Video-Level Aggregation (ActionVLAD) [13], Hidden Two-Stream Convolutional Networks [14], Two-Stream Inflated 3D ConvNet (Two-StreamI3D) [15], and Temporal 3D ConvNets (T3D) [16]

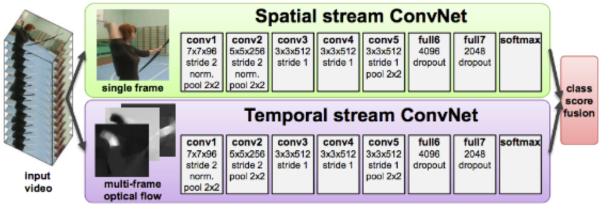

Fig. 1. Two stream architecture proposed by Simmoyan and Zisserman [10]

In this paper, we shall present an ensemble approach, using both convolutional and recurrent, single stream neural networks to recognize 27 different human actions using the UTD-MHAD dataset [17].

II. Methods and Modeling

A. Dataset

Collected via a Microsoft Kinect sensor, a wearable inertial sensor, which includes an accelerometer and a gyroscope, and video camera, the UTD-MHAD dataset contains 27 different actions performed by 8 subjects (4 females and 4 males). Each subject was asked to repeat each action 4 times. After removing three corrupted sequences, the dataset contains 861 sequences. There are 4 different types of data, the Depth dataset, Inertial dataset, Skeleton dataset, and RGB dataset. They are plotted and shown respectively in Fig. 2 - 5.

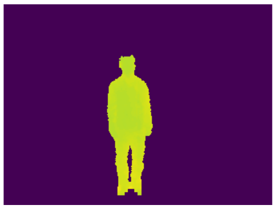

Fig. 2. Depth map dataset of a subject performing a tennis swing

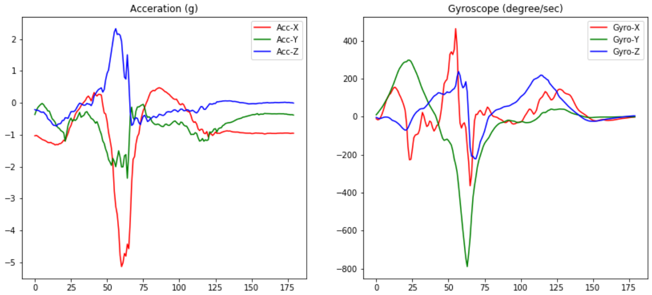

Fig. 3. Inertial dataset of a subject performing a tennis swing

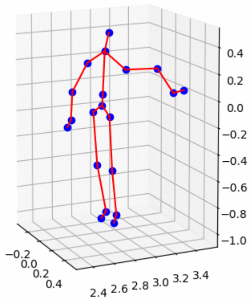

Fig. 4. Skeleton dataset of a subject performing a tennis swing

Fig. 5. Video dataset of a subject performing a tennis swing

In this paper, we would only be using the inertial and skeleton dataset for our models. We split the train-validation data in the same fashion as the original paper [17], where subjects 1, 3, 5, 7 would be used for training and subjects 2, 4, 6, 8 for validation, so that we could perform a baseline comparison.

B. Data Pre-processing

1) Inertial Data: The inertial data was re-sampled to the mean period of 180 units. This was found to allow the model to achieve convergence much quicker. Amplitude normalization was also tried but subsequently removed as it does not show improvement in the model's training. A histogram plot of the period distribution is shown in Fig. 6

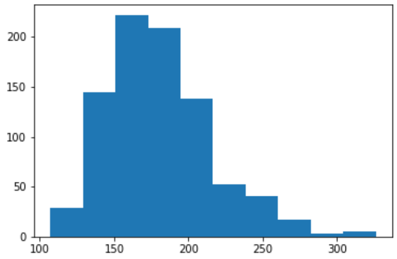

Fig. 6. Histogram plot of the period distribution of Inertial Data

Histogram plots of the inertial data are shown in Fig. 7 - 10 to show the distribution of amplitude for both maximum and minimum of the 3-axial accelerometer and gyroscope.

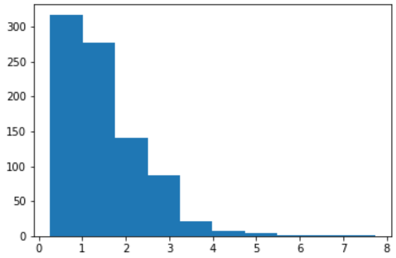

Fig. 7. Histogram plot of the period distribution of the maximum amplitude of the 3-axial accelerometer data

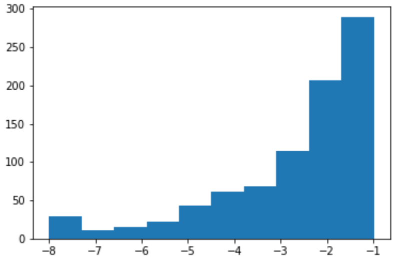

Fig. 8. Histogram plot of the period distribution of the minimum amplitude of the 3-axial accelerometer data

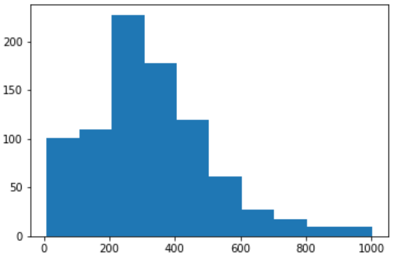

Fig. 9. Histogram plot of the period distribution of the maximum amplitude of the 3-axial gyroscope data

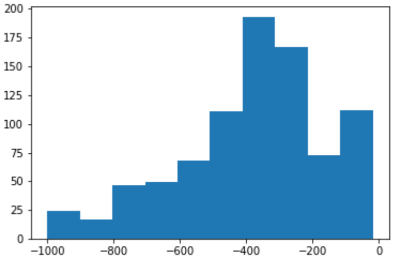

Fig. 10. Histogram plot of the period distribution of the minimum amplitude of the 3-axial gyroscope data

2) Skeleton Data: To aid with data fusion later, we re-sampled the skeleton data having 180 frames.

C. Models

For all our models, we used the AdamOptimizer [18] with a learning rate of $1e^{-4}$, β_1 of 0.9, and β_2 of 0.999. We initialize our trainable parameters using the Xavier Glorot initializer [19], and set our batch size to 3 to allow our model our model to generalize better [20]. In this paper, we shall experiment with the following models:

1) Simple LSTM
2) BiDirectional LSTM
3) Conv LSTM
4) UNet LSTM
5) Ensemble of Conv LSTM and UNet LSTM

The full architecture of the models can be found in the Appendix.

1) Simple LSTM: We start off by having a Simple LSTM model, which comprises of an LSTM unit of 512 hidden units, followed by a Dense layer, to classify all 27 actions. The LSTM [21] uses a gated approach to learn local, distributed, real-valued, and noisy pattern representations much quicker then real-time recurrent learning, back propagation through time, recurrent cascade correlation, Elman nets, and neural sequence chunking. Each recurrent unit in the LSTM was set to have a dropout rate of 0.2.

2) BiDirectional: The next model flips the copy of LSTM unit and concatenates it with original LSTM unit. This allows a form of generative deep learning, where the output layer can get information from the backwards and forward states simultaneously [22]. This is then followed by a Dense layer with a softmax activation, to classify all 27 actions.

3) Conv LSTM: The third model uses a series of 1D Convolutional and 1D Maxpooling layers to extract higher dimensional features before feeding them into 2 LSTM units to capture the temporal information. The output of the LSTM units is then flattened out and we attached a Dropout layer with a dropout rate of 0.5 before adding a Dense layer with a softmax activation to classify all 27 actions.

4) UNet LSTM: The last model is adapted from a popular architecture commonly used in image semantic segmentation tasks. The UNet [23] is an encoder-decoder convolutional neural network that is almost symmetric in the contraction and expansion path. In the contraction path, the input was is being fed through a series of convolutions and max-pooling, increasing the feature maps and decreasing the resolution of the image. This increases the "what" and decreases the "where". In the expansion path, the high dimensional features with low resolution is being up-sampled via convolutional kernels. The features maps were reduced during this operation. A novel feature of UNet is that it implements a concatenation of high dimensional features in the contraction path to the low dimensional feature maps of the expansion layers.

5) Ensemble of Conv LSTM and UNet LSTM: The Conv LSTM and UNet LSTM were found to be performing well on the validation set, so we took an average of their softmax activation to create an ensemble as shown below.

$$\sigma(z)_{j,ensemble} = \frac{\sigma(z)_{j,cLSTM} + \sigma(z)_{j,uLSTM}}{2} \quad (1)$$

where $\sigma(z)_j$ is the softmax output for an action j, given by:

$$\sigma(z)_j = \frac{\exp(z_j)}{\sum_{k=1}^{27} \exp(z_k)} \quad (2)$$

III. Computational Simulation

The code is written in Python, using Keras with Tensorflow backend, NumPy, SciPy and Matplotlib libraries. The code can be found in this Github repository, https://github.com/notha99y/Multimodal_human_actions. The models are trained on Google Colaboratory notebooks.

IV. Results

A. Inertial Data

1) Simple LSTM: The accuracy on the validation of our simple LSTM was 0.238. The train-val accuracy and loss plots of the Simple LSTM are shown in Fig. 11, 12 respectively.

It shows that the model is quickly over-fitting after epoch 2. The confusion matrix is show in Fig. 13.

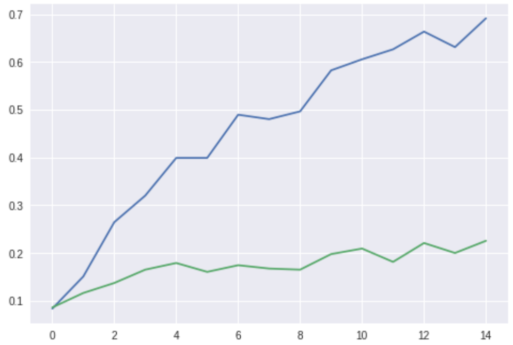

Fig. 11. Train (blue)-val (green) accuracy plot of the Simple LSTM model

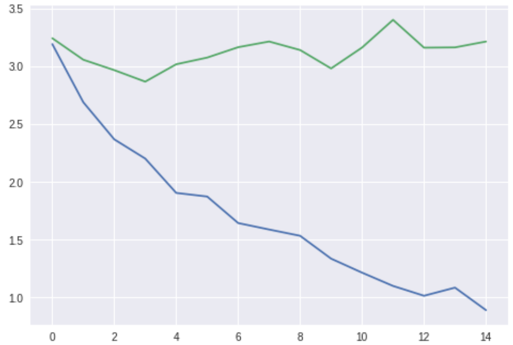

Fig. 12. Train (blue)-val (green) loss plot of the Simple LSTM model

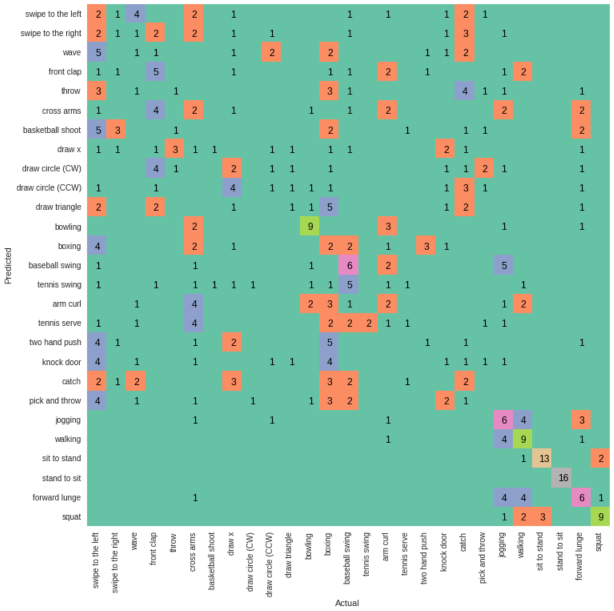

Fig. 13. Confusion matrix of the Simple LSTM model

2) Bi-Directional LSTM: Similarly, the train-val accuracy, loss plots are shown in in Fig.14, 15 respectively, with the confusion matrix in Fig. 16 with a validation accuracy of 0.465.

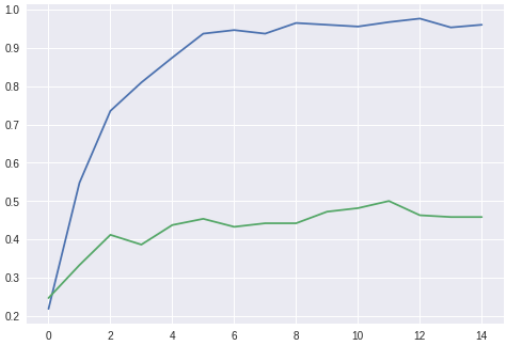

Fig. 14. Train (blue)-val (green) accuracy plot of the Bi-Directional LSTM model

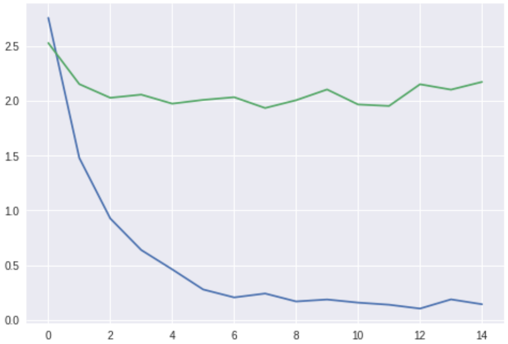

Fig. 15. Train (blue)-val (green) loss plot of the Bi-Directional LSTM model

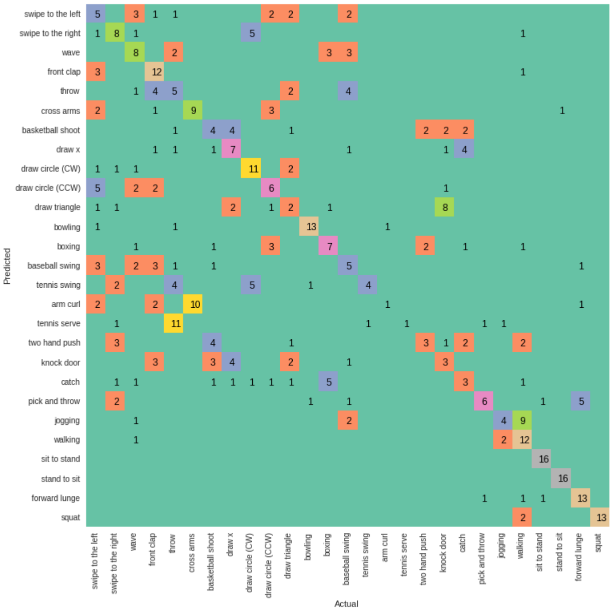

Fig. 16. Confusion matrix of the Bi-Directional LSTM model

3) Conv LSTM: The Conv LSTM was the first major breakthrough among our models, achieving a validation accuracy of 0.700. The train-val accuracy, loss plots are shown in Fig. 17, 18 with the confusion matrix in Fig. 19.

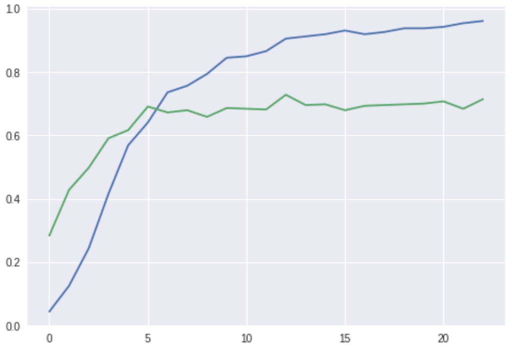

Fig. 17. Train (blue)-val (green) accuracy plot of the Conv LSTM model

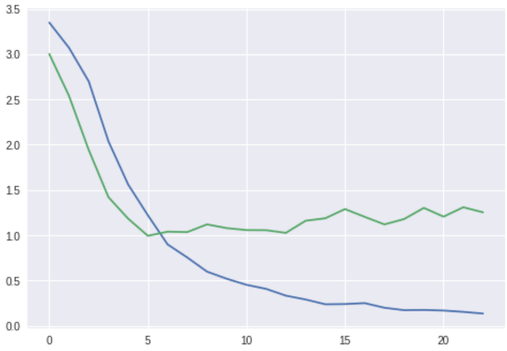

Fig. 18. Train (blue)-val (green) loss plot of the Conv LSTM model

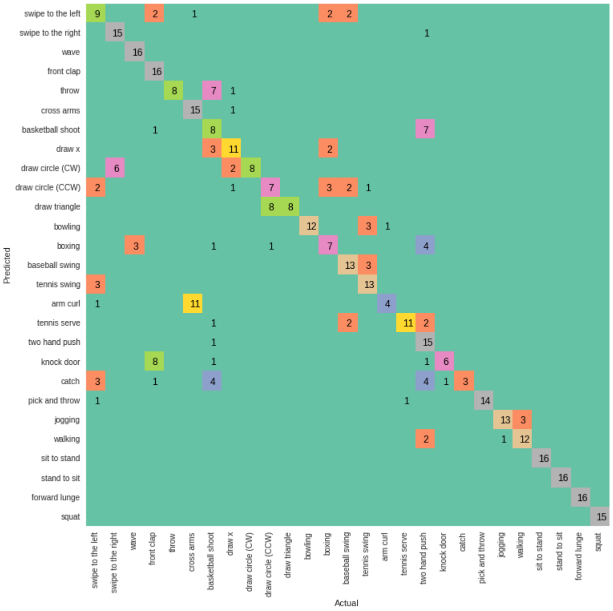

Fig. 19. Confusion matrix of the Conv LSTM model

4) UNet LSTM: Lastly, our UNet LSTM got the highest accuracy of 0.712. The relevant plots are shown below.

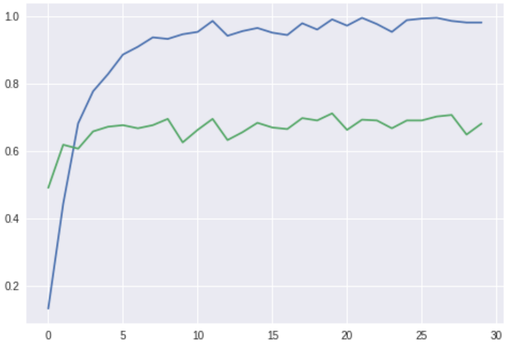

Fig. 20. Train (blue)-val (green) accuracy plot of the UNet LSTM model

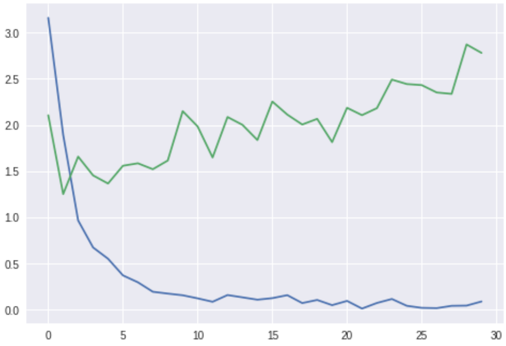

Fig. 21. Train (blue)-val (green) loss plot of the UNet LSTM model

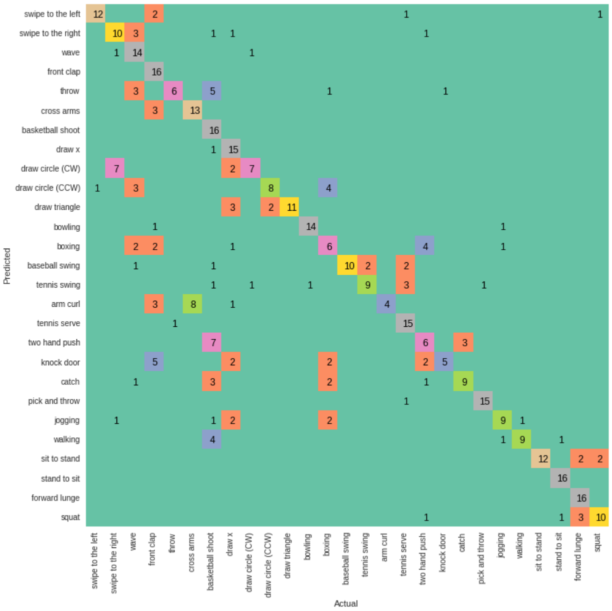

Fig. 22. Confusion matrix of the UNet LSTM model

5) Ensemble of Conv LSTM and UNet LSTM: The confusion matrix of the ensemble is shown in Fig. 23 with a total accuracy of 0.765.

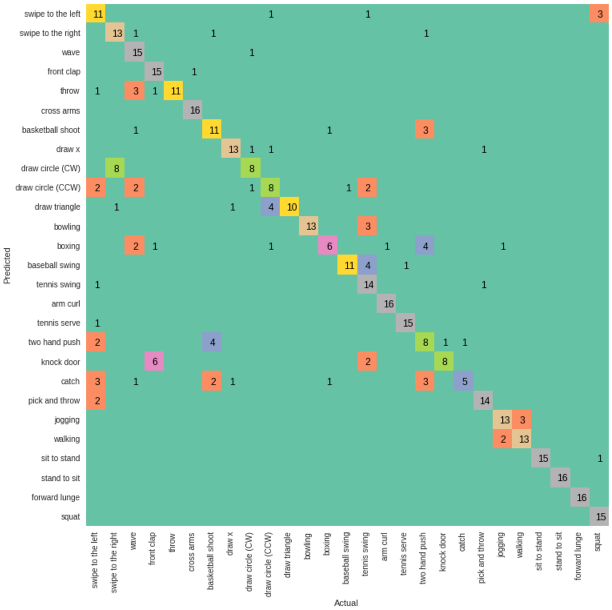

Fig. 23. Confusion matrix of the Ensemble of Conv LSTM and UNet LSTM

B. Inertial + Skeleton Data

1) Conv LSTM: We combined the Skeleton data along the features axis with the Inertial data and trained the Conv LSTM to achieve an accuracy of 0.784. The train-val accuracy, loss plots together with the confusion matrix is shown below.

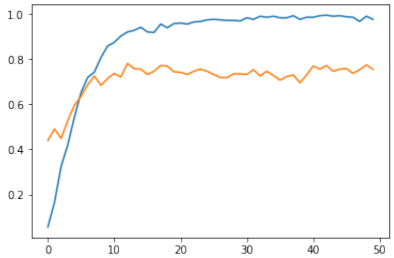

Fig. 24. Train (blue)-val (green) accuracy plot of the Conv LSTM model on both Inerital and Skeleton Data

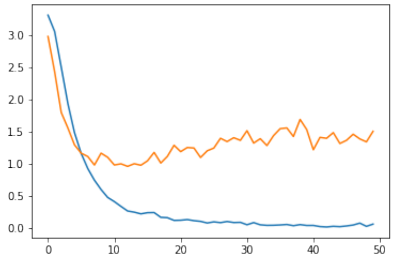

Fig. 25. Train (blue)-val (green) loss plot of the Conv LSTM model on both Inerital and Skeleton Data

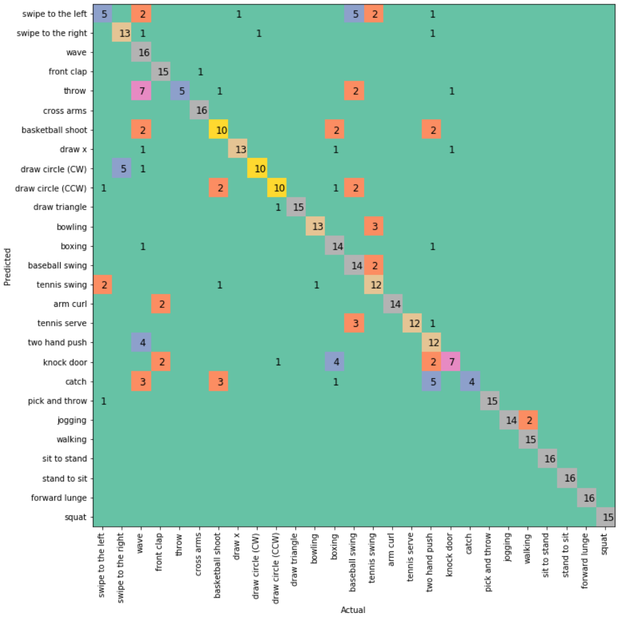

Fig. 26. Confusion matrix of the Conv LSTM model on both Inerital and Skeleton Data

2) UNet LSTM: Similiarly, the same was done with the UNet LSTM model and it achieve an accuracy of 0.742. The relevant plots are shown below.

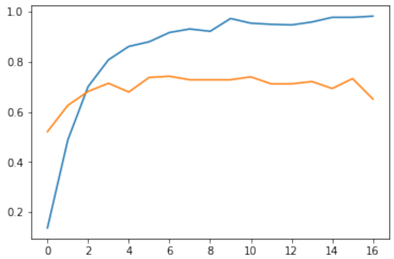

Fig. 27. Train (blue)-val (green) accuracy plot of the UNet LSTM model on both Inertial and Skeleton Data

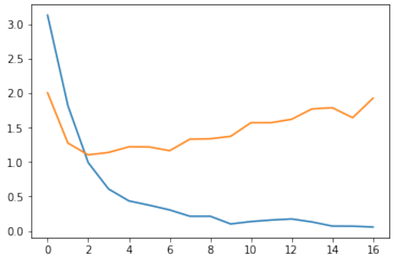

Fig. 28. Train (blue)-val (green) loss plot of the UNet LSTM model on both Inertial and Skeleton Data

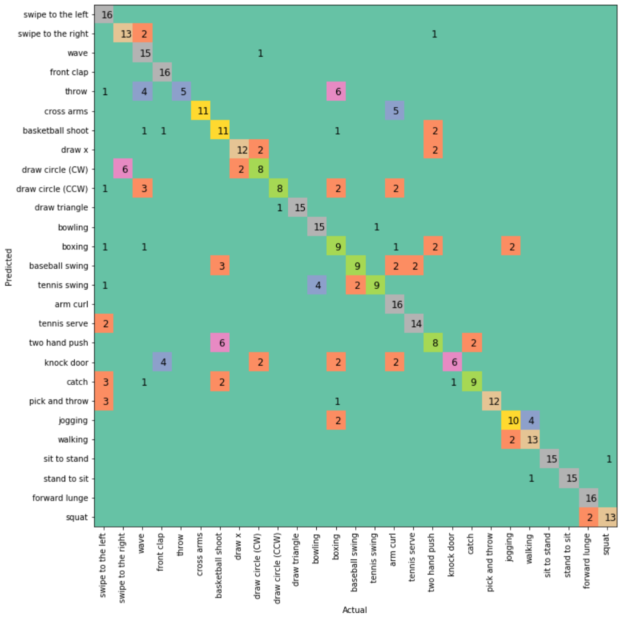

Fig. 29. Confusion matrix of the UNet LSTM model on both Inertial and Skeleton Data

3) Ensemble of Conv LSTM and UNet LSTM: Lastly, we combined all our stocks together and put together an ensemble which achieved an accuracy of 0.821. The confusion matrix is shown below in Fig 30.

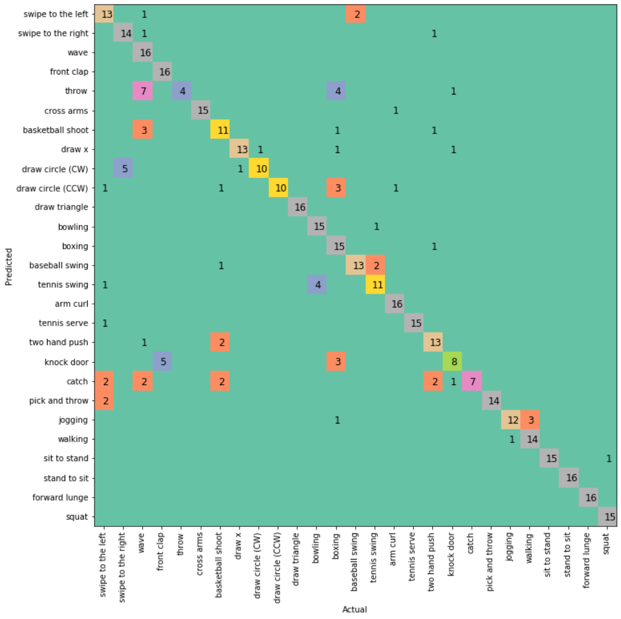

Fig. 30. Confusion matrix of the Ensemble of Conv LSTM and UNet LSTM model on both Inertial and Skeleton Data

C. Summary

The summary of results can be found below in Table. I.

TABLE I
SUMMARY OF VALIDATION ACCURACY OF THE DIFFERENT MODELS

	S-LSTM	B-LSTM	C-LSTM	U-LSTM	Ensemble
Iner	0.238	0.465	0.700	0.712	0.765
Iner + skel	-	-	0.784	0.742	0.821

V. Discussion, and Conclusion

In this paper, we have demonstrated an ensemble approach in HAR, achieving an accuracy of 0.821 on the validation set using the Inertial and Skeleton Data which is an improvement compared to the baseline paper of 0.672. This could be due to the convolutional networks being able to capture more generic, higher dimensional features compared to the Collaborative Representation Classifier (CRC) method used in [17]. However, we find that there are still some things we can work on and in the following sub sections, we shall explore other methods as future work to improve our validation accuracy.

A. Issue of model over-fitting

The main recurring challenge we faced was that our model was over-fitting at early epochs. Adding dropout layers seemed to reduce this, and ensembling the UNet LSTM with the Conv LSTM allows the our model to explore other solution spaces [24]. Over-fitting tends to happen when our model tries to learn high frequency features that may not be useful. Adding Gaussian Noise with zero mean and data points in all frequencies might enhance the learning capability of our model. Similarly, the time sequences of different subjects are quite varied even for the same activities. Performing data augmentation using time scaling and translation would increase the amount of training data, allowing our model to generalize better. On a side note, our model could also be trimmed further to reduce its complexity, and also its risk of over-fitting.

B. Data Fusion of Depth and RGB data

Fusion with the Depth and RGB data would allow more training input variables for our models to learn from, hence improving our validation accuracy.

C. Ensemble Learning

Right now, our ensemble simply takes the average of our models' softmax activations. We could further enhance the validation accuracy and reduce over-fitting, by exploring different ensemble learning approaches [25] such as Boosting [26], Voting and Bagging mechanisms [27].

VI. Appendix

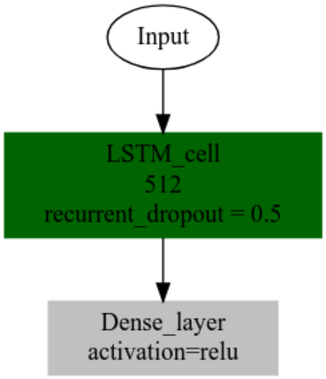

Fig. 31. Full model of Simple LSTM

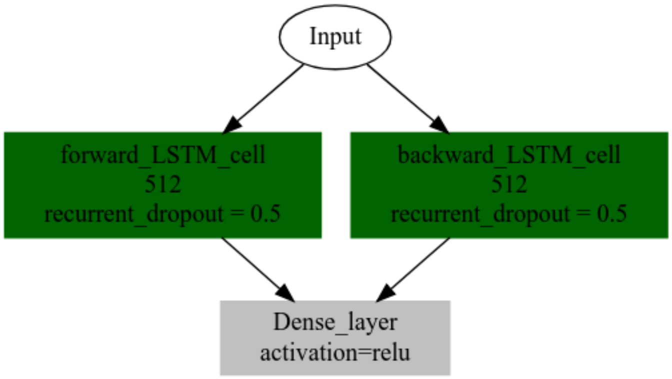

Fig. 32. Full model of Bi-Directional LSTM

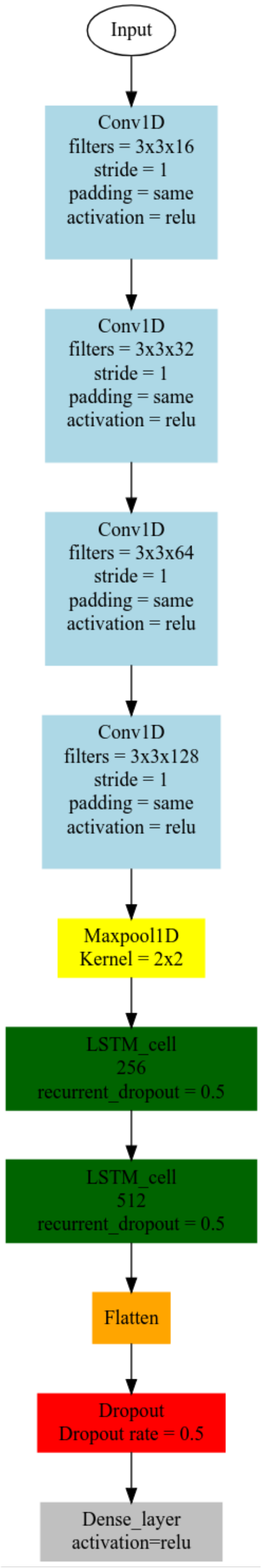

Fig. 33. Full model of Conv LSTM

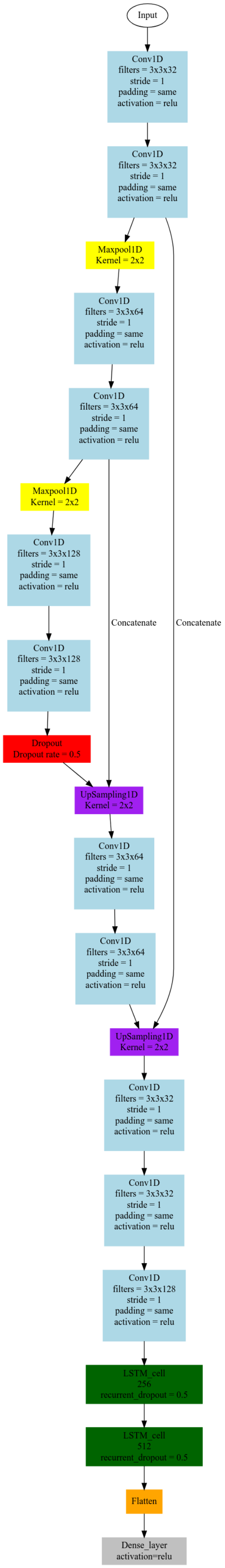

Fig. 34. Full model of UNet LSTM

REFERENCES

[1] T. Choudhury, G. Borriello, S. Consolvo, D. Haehnel, B. Harrison, B. Hemingway, J. Hightower, P. . Klasnja, K. Koscher, A. LaMarca, J. A. Landay, L. LeGrand, J. Lester, A. Rahimi, A. Rea, and D. Wyatt, "The mobile sensing platform: An embedded activity recognition system," *IEEE Pervasive Computing*, vol. 7, no. 2, pp. 32–41, April 2008.

[2] S.-M. Lee, S. M. Yoon, and H. Cho, "Human activity recognition from accelerometer data using convolutional neural network," in *2017 IEEE International Conference on Big Data and Smart Computing (BigComp)*, Feb 2017, pp. 131–134.

[3] H. Wang, A. Kläser, C. Schmid, and C. Liu, "Action recognition by dense trajectories," in *CVPR 2011*, June 2011, pp. 3169–3176.

[4] Laptev and Lindeberg, "Space-time interest points," in *Proceedings Ninth IEEE International Conference on Computer Vision*, Oct 2003, pp. 432–439 vol.1.

[5] 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*, Oct 2005, pp. 65–72.

[6] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei, "Large-scale video classification with convolutional neural networks," in *CVPR*.

[7] J. Donahue, L. A. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell, "Long-term recurrent convolutional networks for visual recognition and description," *CoRR*, vol. abs/1411.4389, 2014. [Online]. Available: http://arxiv.org/abs/1411.4389

[8] D. Tran, L. D. Bourdev, R. Fergus, L. Torresani, and M. Paluri, "C3D: generic features for video analysis," *CoRR*, vol. abs/1412.0767, 2014. [Online]. Available: http://arxiv.org/abs/1412.0767

[9] L. Yao, A. Torabi, K. Cho, N. Ballas, C. Pal, H. Larochelle, and A. Courville, "Describing Videos by Exploiting Temporal Structure," *ArXiv e-prints*, Feb. 2015.

[10] K. Simonyan and A. Zisserman, "Two-stream convolutional networks for action recognition in videos," *CoRR*, vol. abs/1406.2199, 2014. [Online]. Available: http://arxiv.org/abs/1406.2199

[11] C. Feichtenhofer, A. Pinz, and A. Zisserman, "Convolutional two-stream network fusion for video action recognition," *CoRR*, vol. abs/1604.06573, 2016. [Online]. Available: http://arxiv.org/abs/1604.06573

[12] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. V. Gool, "Temporal segment networks: Towards good practices for deep action recognition," *CoRR*, vol. abs/1608.00859, 2016. [Online]. Available: http://arxiv.org/abs/1608.00859

[13] R. Girdhar, D. Ramanan, A. Gupta, J. Sivic, and B. C. Russell, "Actionvlad: Learning spatio-temporal aggregation for action classification," *CoRR*, vol. abs/1704.02895, 2017. [Online]. Available: http://arxiv.org/abs/1704.02895

[14] Y. Zhu, Z. Lan, S. D. Newsam, and A. G. Hauptmann, "Hidden two-stream convolutional networks for action recognition," *CoRR*, vol. abs/1704.00389, 2017. [Online]. Available: http://arxiv.org/abs/1704.00389

[15] J. Carreira and A. Zisserman, "Quo vadis, action recognition? A new model and the kinetics dataset," *CoRR*, vol. abs/1705.07750, 2017. [Online]. Available: http://arxiv.org/abs/1705.07750

[16] A. Diba, M. Fayyaz, V. Sharma, A. H. Karami, M. M. Arzani, R. Yousefzadeh, and L. V. Gool, "Temporal 3d convnets: New architecture and transfer learning for video classification," *CoRR*, vol. abs/1711.08200, 2017. [Online]. Available: http://arxiv.org/abs/1711.08200

[17] C. Chen, "Utd-mhad: A multimodal dataset for human action recognition utilizing a depth camera and a wearable inertial sensor," 09 2015.

[18] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *CoRR*, vol. abs/1412.6980, 2014. [Online]. Available: http://arxiv.org/abs/1412.6980

[19] X. Glorot and Y. Bengio, "Understanding the difficulty of training deep feedforward neural networks," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, AISTATS 2010, Chia Laguna Resort, Sardinia, Italy, May 13-15, 2010*, 2010, pp. 249–256. [Online]. Available: http://www.jmlr.org/proceedings/papers/v9/glorot10a.html

[20] E. Hoffer, I. Hubara, and D. Soudry, “Train longer, generalize better: closing the generalization gap in large batch training of neural networks,” in *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, 2017, pp. 1729–1739.

[21] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997. [Online]. Available: http://dx.doi.org/10.1162/neco.1997.9.8.1735

[22] M. Schuster, K. K. Paliwal, and A. General, “Bidirectional recurrent neural networks,” *IEEE Transactions on Signal Processing*, 1997.

[23] O. Ronneberger, P. Fischer, and T. Brox, “U-net: Convolutional networks for biomedical image segmentation,” *CoRR*, vol. abs/1505.04597, 2015. [Online]. Available: http://arxiv.org/abs/1505.04597

[24] R. Maclin and D. W. Opitz, “Popular ensemble methods: An empirical study,” *CoRR*, vol. abs/1106.0257, 2011. [Online]. Available: http://arxiv.org/abs/1106.0257

[25] C.Zhang and Y.MA, *Ensemble Machine Learning ,Methods and Applications*. Landon: Springer, 2012, pp.254-270.

[26] R. E. Schapire, Y. Freund, P. Bartlett, and W. S. Lee, “Boosting the margin: A new explanation for the effectiveness of voting methods,” 1997.

[27] R. E. Schapire, *The strength of weak learnability*. Kluwer Academic Publishers, 1990, p. 197-227.