Increasing the efficiency of action recognition models using Procrust analysis

Cody Wong

Stuyvesant High School, NY, NY

Abstract

The use of Action Recognition in everyday life is increasing: from security cameras to new apps being developed daily. With that comes concerns of whether older technologies could withstand new innovative algorithms. In this paper, I developed a method to derive a higher accuracy & speed benchmark. This improvement is done through procrust analysis, an algorithm that derives similarity values based on two 3d shapes. Most LSTM models use mediapipe, a library that contains a model that implants 33 landmarks in specific locations on a human body. Typically, one would use all 33 landmarks, which is 99 inputs(x, y, z for each landmark). However, mediapipe was designed for generalized usage. This is a problem that will limit the efficiency of action recognition, so I created a method to reduce the amount of data inputted into the action recognition model. To do so, I used procrust analysis to derive the similarity of an action to itself using the UCF101 dataset. We can derive which landmarks are useful for certain actions, creating a specialized set of landmarks for these actions. This selection improvement of landmarks allows for better models to come out that can be used in more specialized areas: monitoring crimes, monitoring work pace, and monitoring fitness.

Introduction

Computational cost is the biggest issue in the usability of action recognition models. To even train your own it would require multiple iterations, possibly taking over days based on your computer's GPU or CPU. This is mainly because action recognition models take video footage of an individual (or many) as input and return a classification type based on their training. When using these models in live settings with equipment, efficiency becomes a major concern. This can be resolved if the paradigm of action recognition models are at a higher standard. Current innovative action recognition models focus heavily on accuracy, which poses a problem because many of these models use slower algorithms to achieve higher accuracy. For example, the highest-accuracy model that uses the UCF101 dataset recently published in 2024 and relies on a VLM (Visual Language Model) and a VIT (Visual Transformer) algorithm. VLM is mainly used to derive information from images and Vit can change that information into a classification output (action). This method is excellent for detecting a wide range of actions with high accuracy, but is unusable for mobile devices due to its speed limitations due to VLM's speed constraints (Lu et al., 2023). SMART frame selection is an action recognition model that uses a frame selection system to derive the best inputs (Gowda et al., 2012). While the model is decent for accuracy, it requires preprocessing the entire video for data, making live detection slower because of the delay (Gowda et al., 2012). Conversely CNN and BI-directional LSTM models can achieve high accuracy in action recognition, they require mid-range GPUs to run at 30 fps (Anon., 2018). MediaPipe with bi-directional LSTM, on the other hand, can run at 30–60 fps on mobile devices due to its efficiency (Anon., 2018). The MediaPipe model can be further simplified by reducing the inputs to only the essential data needed to detect actions (Anon., 2018). Procrustes Analysis, Chamfer Distance, and Iterative Closest Point are three methods that can be used to reduce the inputs (Anon., 2018). Procrustes Analysis has a computational complexity of O(N), making it the fastest method (Anon., 2018). Chamfer Distance has a complexity of O(N * M), while Iterative Closest Point is O(K * N * logM) (Anon., 2018). Since the selected method would require pairwise grouping of each video, which requires N * N-1 iterations, Procrustes Analysis would be the most efficient option (Anon., 2018).

Methods

First, we used MediaPipe, which provides 99 data points corresponding to 33 landmarks, each with X, Y, and Z coordinates (Fig. 1). However, landmarks 0–10 are irrelevant, so I excluded them when building the LSTM action recognition models. To rank the importance of landmarks, I applied Procrustes Analysis, which measures the similarity of two shapes based on moving, rotating, and scaling (Fig. 2). The result is a disparity value $d^2 \in [0, 1]$, where "0" means perfect similarity and "1" means no match at all. To derive similarity values for each action, I used Tkinter (a Python GUI library) to select an action folder and then applied Procrustes Analysis to compare landmarks within each video. This process generated 33 similarity paths, one for each landmark. A pairwise Procrustes Analysis was then performed on these paths, and the disparity values(d^2) were averaged and converted to a score out of 100 using $100\times(1-d^2)100 \times (1 - d^2)100\times(1-d^2)$. These scores were graphed to reveal how similar each landmark is when compared to itself across actions. To refine the selection method for landmarks, I used trial and error. Once a method was determined, I applied a Bi-Directional LSTM model to analyze the chosen landmarks. This model identifies patterns by looking both backward and forward in video sequences. For comparison, I created a control group using 22 landmarks (controlL) and three reduced landmark groups. These three groups represent a range of similarity values: high similarity landmarks, average similarity landmarks, and chaotic (low similarity) landmarks. To determine which landmarks to use, I selected those that appeared in 2 or 3 of the 3 actions, ensuring enough data for training while correlating landmarks shared across multiple actions. For the average similarity landmarks, I chose landmarks within the bounds of average ± std allowing for a balanced variety. For high and low similarity landmarks, I selected those with maximum from a range using 2d standard deviations(max−2×std, max) and minimum (min,bmin+×std). Finally, I graphed speed (ms/steps), computational time, accuracy, and loss (validation and training) for all four sets of landmarks: control, least similar landmarks, most similar landmarks, and average similarity landmarks. After identifying the method that achieved the highest speed and accuracy, I repeated the experiment with this selection method across four different action sets to confirm its effectiveness and ensure it was the optimal approach.

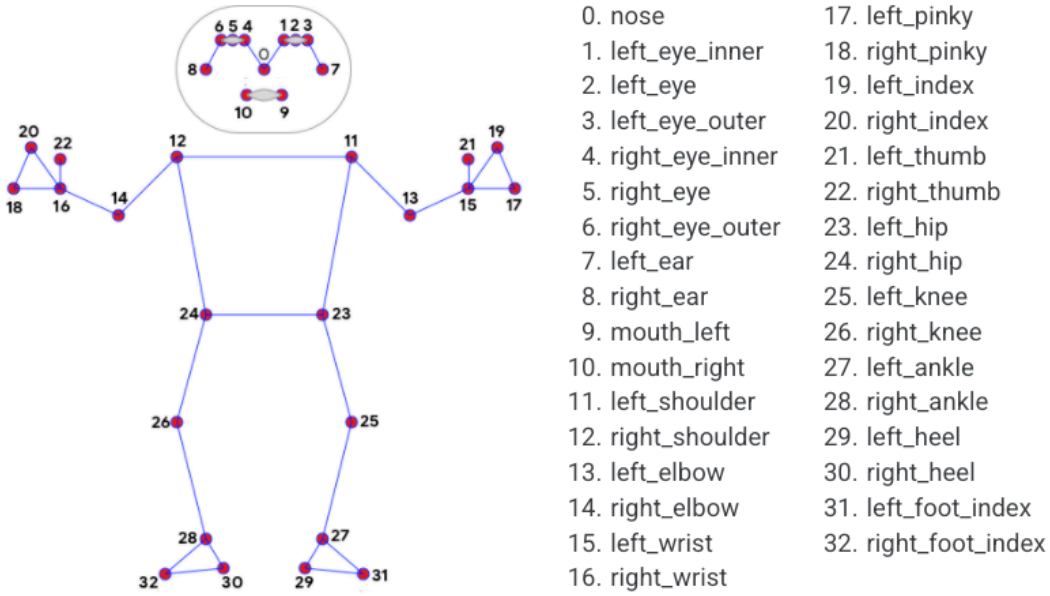


Figure 1. *Landmark Dictionary.* (retrieved from Google LLC)

This illustrates all 33 landmarks in correlation to the placement of the body.

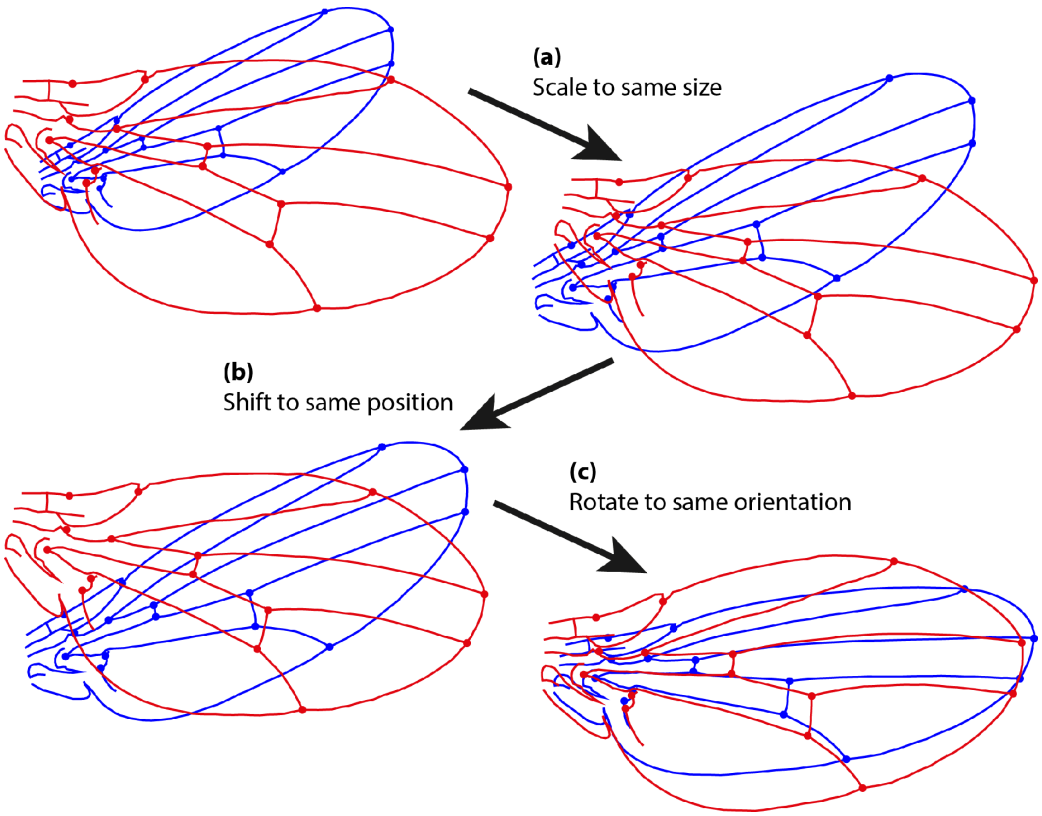


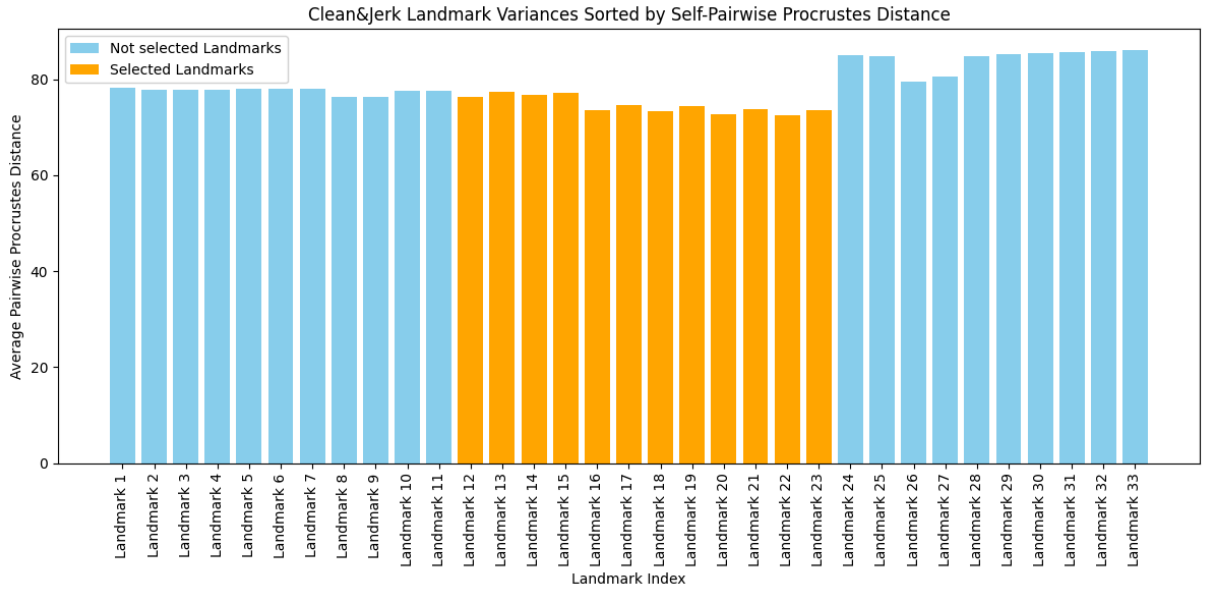
Figure 1. *Procrust Analysis Diagram.* (retrieved from Wikipedia)

This describes what action procrust analysis takes to derive the similarity between two different actions.

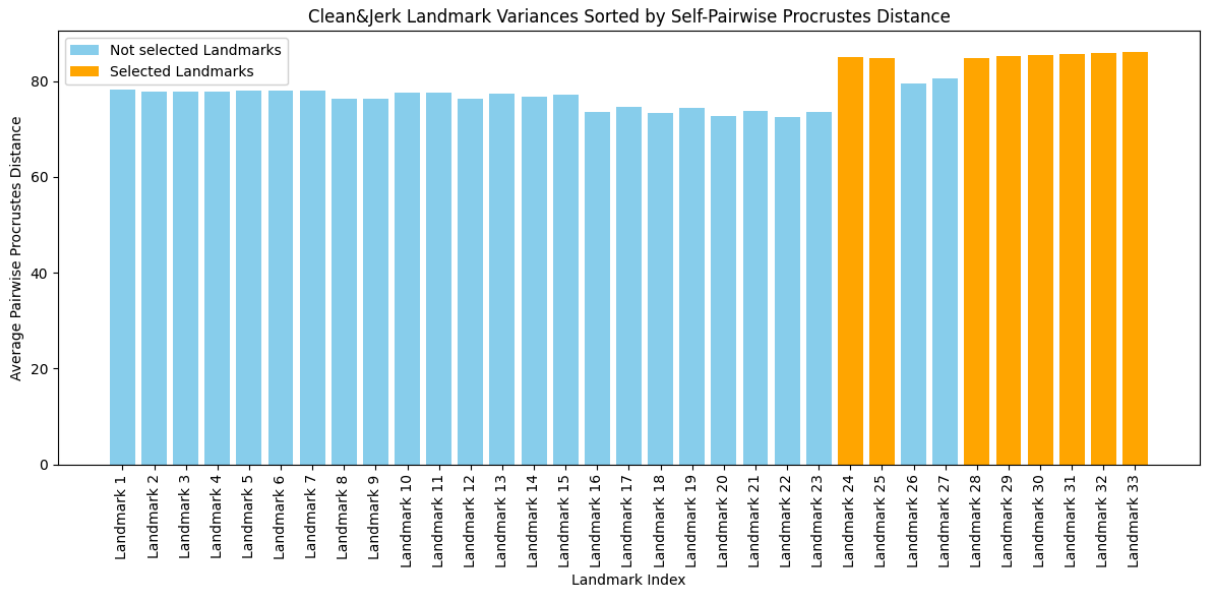
Experimental Design

We first used one set to figure the optimal range. I made 3 Action Recognition models, which all used two folders, “test” and “val”, both containing “Bench Press”, “CleanAndJerk”, and “Lunges”. The UCF101 dataset has 3 folders(“test”, “train”, and “val”) each filled with 101 different actions with many videos in the .avi format.The exercises in test * val were combined into the val in my folder and the train stained train. I created 4 Python scripts: Procrust analysis for a folder with videos of the same action, one LSTM using only high similarity landmarks, one LSTM using average similarity landmarks, and one using very chaotic landmarks. These 3 different landmark scripts will help me determine which method is most useful. I believe that the average distance would be the best because the average should best represent the action.

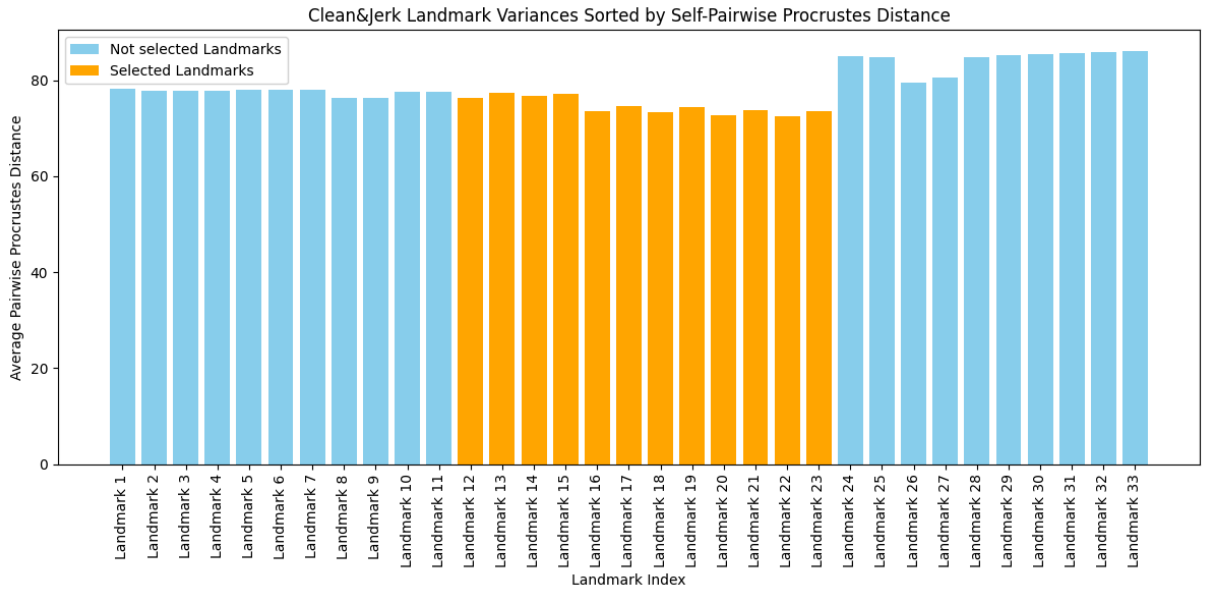
Results



(A) Clean & Jerk Landmarks 12-23 were selected using the average similarity method



(B) Clean & jerk landmarks 24-25, 28-33 selected using the highest similarity method.

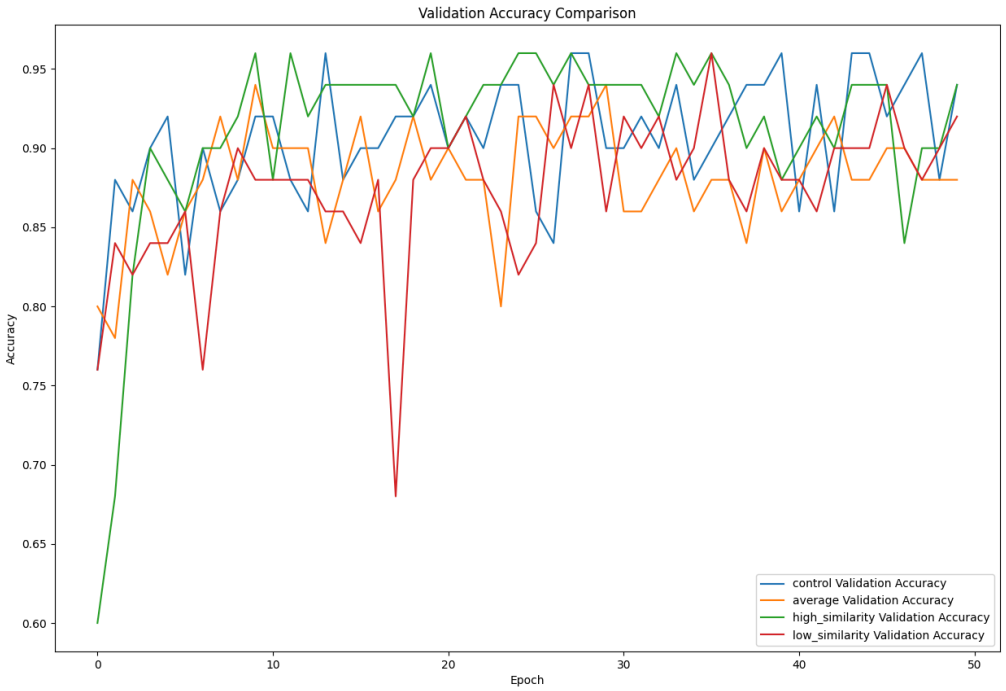


(C) Clean & jerk landmarks 24-25, 28-33 selected using the highest similarity method.

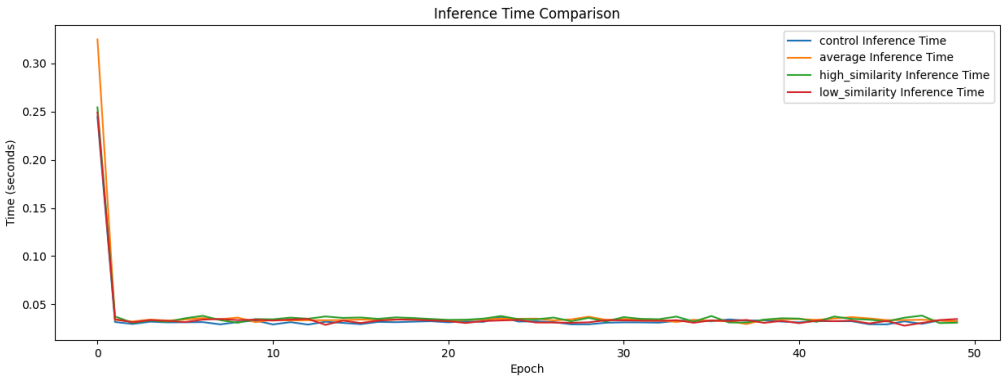
Figure 3 - Shows selected landmarks using the average distance method

Figure 3. *Procrust Analysis Diagram.* (Author generated)

This describes what landmarks were selected using each method.



(A) This Graph shows how each of the 4 graphs correlated in terms of accuracy as epochs went along(one iteration that went to 50 epochs)



(B) This graph shows the speed of inference time, which describes how fast the model at the epoch makes its first thought on the validation set.

Figure 3. *Procrust Analysis Diagram.* (Author generated)

This describes what action procrust analysis takes to derive the similarity between two different actions.

```
Landmark Set: control
  Average Accuracy (after first epoch): 0.8940
  Average Inference Time (after first epoch): 0.0315 seconds
Landmark Set: average
  Average Accuracy (after first epoch): 0.8796
  Average Inference Time (after first epoch): 0.0311 seconds
Landmark Set: high_similarity
  Average Accuracy (after first epoch): 0.8948
  Average Inference Time (after first epoch): 0.0324 seconds
Landmark Set: low_similarity
  Average Accuracy (after first epoch): 0.8655
  Average Inference Time (after first epoch): 0.0335 seconds
```

Figure 4. *Procrust Analysis Diagram.* (Author generated)

This shows the how the 4 sets of landmark data compares to when training

Discussion / Conclusion:

Models using landmarks with higher similarity values achieved higher average accuracy (0.9283) and higher speed (0.0899 seconds). The high_similarity model has, on average, 1.347% higher accuracy and a 6.743% increase in speed. This is quite good considering the upper echelon of accuracy is 96%. High Similarity landmarks are by far the best-resulting landmarks in terms of both speed and accuracy. It is possible to reduce the input data of action recognition models and derive higher efficiency. Low similarity and average similarity do not compare to the high similarity model, showing a type of favoritism within the LSTM models.

Bibliography :

Caba Heilbron, F.; Escorcia, V.; Ghanem, B.; and Carlos Niebles, J. 2015. Activitynet: A large-scale video benchmark for human activity understanding. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 961–970.

Carreira, J., and Zisserman, A. 2017. Quo vadis, action recognition? a new model and the kinetics dataset. In proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 6299–6308.

Choudhury, S. K., Sa, P. K., Bakshi, S., and Majhi, B. 2016. An evaluation of background subtraction for object detection vis-a-vis mitigating challenging scenarios. IEEE Access, vol. 4, pp. 6133–6150.

DOLLÁR P, RABAUD V, COTTRELL G, et al. 2005. Behavior recognition via sparse spatio-temporal features. In 2005 IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance. IEEE, 2005: 65-72.

Dong, J.; Zhang, X.; and Tan, T. 2019. Action-aware spatial-temporal attention for video action recognition. In Proceedings of the 27th ACM International Conference on Multimedia, 239–247.

Fan, H.; Xu, Z.; Zhu, L.; Yan, C.; Ge, J.; and Yang, Y. 2018. Watching a small portion could be as good as watching all: Towards efficient video classification. In IJCAI International Joint Conference on Artificial Intelligence.

Feichtenhofer, C.; Pinz, A.; and Wildes, R. P. 2017. Spatiotemporal multiplier networks for video action recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition, 4768–4777.

Girdhar, R.; and Ramanan, D. 2017. Attentional pooling for action recognition. In Advances in Neural Information Processing Systems, 34–45.

Google LLC. (n.d.). Pose Landmarker Model. Mediapipe. Retrieved April 27, 2023, from https://ai.google.dev/edge/mediapipe/solutions/vision/pose_landmarker

Gorelick, L., Blank, M., Shechtman, E., Irani, M., and Basri, R. 2007. Actions as space-time shapes. IEEE transactions on pattern analysis and machine intelligence, 2007, 29(12): 2247-2253.

Goyal, R.; Kahou, S. E.; Michalski, V.; Materzynska, J.; Westphal, S.; Kim, H.; Haenel, V.; Fruend, I.; Yianilos, P.; Mueller-Freitag, M.; et al. 2017. The"something something"

video database for learning and evaluating visual common sense. In Proceedings of the IEEE International Conference on Computer Vision, 5842–5850.

He, J.-Y., Wu, X., Cheng, Z.-Q., Yuan, Z., & Jiang, Y.-G. (2021). DB-LSTM: Densely-connected Bi-directional LSTM for human action recognition. Neurocomputing, 444, 319–331. https://doi.org/10.1016/j.neucom.2020.05.118

Hu, Y.; Cao, L.; Lv, F.; Yan, S.; Gong, Y.; and T. S. Huang. 2009. Action detection in complex scenes with spatial and temporal ambiguities. In Proc. IEEE 12th Int. Conf. Comput. Vis., Sep./Oct. 2009, pp. 128–135.

Jiang, Y.-G.; Wu, Z.; Wang, J.; Xue, X.; and Chang, S.-F. 2017b. Exploiting feature and class relationships in video categorization with regularized deep neural networks. IEEE Transactions on Pattern Analysis and Machine Intelligence 39(8): 1560–1571.

KEY, SUKTHANKAR R, HEBERT M. 2007. Spatio-temporal shape and flow correlation for action recognition. In 2007 IEEE conference on computer vision and pattern recognition. IEEE, 2007: 1-8.

Khan Muhammad, Ahmad, J., Khan, M., Sajjad, M., & Baik, S. W. (2021). Human action recognition using attention-based LSTM network with dilated CNN features. Future Generation Computer Systems, 120, 197–208. DOI: 10.1016/j.future.2021.06.045.

Liu, J.; Luo, J.; and Shah, M. 2009. Recognizing realistic actions from videos 'in the wild.' In Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), Jun. 2009, pp. 1996–2003.

Meng, F., Richer, M., Tehrani, A., La, J., Kim, T. D., Ayers, P. W., & Heidar-Zadeh, F. (2022). Procrustes: A python library to find transformations that maximize the similarity between matrices. Computer Physics Communications, 280, 108334. https://doi.org/10.1016/j.cpc.2022.108334

RAHMANI H, BENNAMOUN M. 2017. Learning action recognition model from depth and skeleton videos. In Proceedings of the IEEE International Conference on Computer Vision. 2017: 5832-5841.

RAHMANI H, MIAN A. 2016. 3d action recognition from novel viewpoints. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016: 1506-1515.

RAHMANI H, MIAN A, SHAH M. 2017. Learning a deep model for human action recognition from novel viewpoints. IEEE transactions on pattern analysis and machine intelligence, 2017, 40(3): 667-681.

Robusto KM, Trost SG. 2012. Comparison of three generations of ActiGraph™ activity monitors in children and adolescents. Journal of Sports Sciences. 2012;30(13):1429–1435.

Sevilla-Lara, L.; Sun, Y.; Jampani, V.; Achille, A.; Yazdani, S.; Shah, M.; and Feichtenhofer, C. 2019. Only time can tell: Distinguishing static and dynamic actions. In Proceedings of the IEEE/CVF International Conference on Computer Vision, 9520–9529.

Simonyan, K., and Zisserman, A. 2014. Two-stream convolutional networks for action recognition in videos. In Advances in neural information processing systems, 568–576.

Van Houdt, G., Mosquera, C., & Nápoles, G. (2020). A Review on the Long Short-Term Memory Model. Artificial Intelligence Review, 53. https://doi.org/10.1007/s10462-020-09838-1.

Wang, L.; Xiong, Y.; Wang, Z.; Qiao, Y.; Lin, D.; Tang, X.; and Van Gool, L. 2016. Temporal segment networks: Towards good practices for deep action recognition. In European conference on computer vision, 20–36. Springer.

Wu, Y.; Chen, S.; Li, X.; and Tian, Q. 2019a. Multi-agent reinforcement learning based frame sampling for effective action recognition. In Proceedings of the 27th ACM International Conference on Multimedia, 33–41.

Wu, Y.; Chen, S.; Li, X.; and Tian, Q. 2019b. Liteeval: A coarse-to-fine framework for resource efficient video recognition. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 11 187–11 196.

Wu, Y.; Chen, S.; Li, X.; and Tian, Q. 2019c. Adaframe: Adaptive frame selection for fast video recognition. In Proceedings of the IEEE/CVF International Conference on Computer Vision, 4151–4160.

Yeung, S.; Russakovsky, O.; Mori, G.; and Fei-Fei, L. 2016. End-to-end learning of action detection from frame glimpses in videos. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2678–2687.

Wikipedia contributors. (2024, November 26). *Procrustes analysis*. Wikipedia. Retrieved December 17, 2024, from https://en.wikipedia.org/wiki/Procrustes_analysis

Zhang, H.; Lan, C.; Zeng, W.; and Chen, Z. 2020. Relation feature learning for video action recognition. IEEE Transactions on Image Processing 29: 6488–6501.

ZHANG S, YANG Y, XIAO J, et al. 2018. Fusing geometric features for skeleton-based action recognition using multi-layer LSTM networks. IEEE Transactions on Multimedia, 2018, 20(9): 2330-2343.

ZHANG H, NAN Z, YANG T, et al. 2020. A Driving Behavior Recognition Model with Bi-LSTM and Multi-Scale CNN. In 2020 IEEE Intelligent Vehicles Symposium (IV). IEEE, 284-289.

Zdravevski, Eftim, Biljana Risteska Stojkoska, Marie Standl, and Holger Schulz. 2017. "Automatic machine-learning based identification of jogging periods from accelerometer measurements of adolescents under field conditions." PloS one 12, no. 9: e0184216.

Zheng, J.; Jiang, Z.; Che, H.; Zhao, Z.; and Xing, J. 2020. Dynamic sampling networks for efficient action recognition. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 34, 12 970–12 977.