Video-based action recognition (project report)

Tomiris Rakhimzhanova
Nazarbayev University
tomiris.khalimova@nu.edu.kz

Rasul Yermagambet
Nazarbayev University
rasul.yermagambet@nu.edu.kz

Olzhas Zhangeldinov
Nazarbayev University
olzhas.zhangeldinov@nu.edu.kz

Rustam Chibar
Nazarbayev University
rustam.chibar@nu.edu.kz

Abstract

With the development of the field of computer vision, the need to determine human movements using neural networks rises. However, with the increase in video volumes on Internet platforms, the difficulty of obtaining a model with high accuracy and low computational cost increases. Therefore, researchers develop past designs to meet new requirements. In this article, we want to show the solution for the problem of sampling the videos into frames and train model on each 2nd frame to reduce computational cost. However, such approach leads to poor accuracy. Thus, in this article, we present a efficient model for video recognition with a combination of time-shift module and popular neural networks as a backbone which can solve action recognition problem using full video and minimum computational cost.

1. Introduction

In these days and age, the rapid growth of video recorded using camera devices and which is posted on social media sharing platforms (such as Youtube and Instagram) has created a huge amount of video data. The ability to understand complex actions plays an important role in many important social applications, including intelligent surveillance, patient monitoring, sports analysis, and human-computer interaction. This makes human action recognition an active area in the computer vision community. This project aims to develop a software framework for demonstrating action recognition from video sequences. A framework based on deep learning models or a traditional input model should allow users to input video clips and automatically recognize the actions performed by a person in a given video. Thus, this project aims to automatically classify a data set of person actions on video.

1.1. Literature review

To solve the problem of recognizing human movements, most studies use CNN, but the problem with this neural network is the inability to capture spatiotemporal information. In one of the solutions proposed in [1], the authors propose to use the CNN, which is used for recognition of 3d images, thereby supplementing 2d images with data on spatiotemporal characteristics. The output is then sent to a recurrent neural network to classify the sequence of frames.

Another example is the paper [2] where 2d convolutional networks are used to detect motion and then LSTM networks are used to classify motion. However, these two methods have limitations such as the lack of joint spatiotemporal learning and difficulties such as lack of computational power in training [3]. In addition, to solve the problem of costly training, video sampling is used by cutting frames from the sequence, which leads to the loss of important information about the movement of a person, which leads to low learning outcomes [4].

In order to reduce the computational cost and required memory, some authors applied lower-dimensional filter such as Lego filter, however, the datasets that used for training and testing collected from smartphones or sensors as a time series signals [13]. One of the approach that has been applied in [16] for decreasing the computational time and cost is human motion tracking (isolation of the moving person from background) using Gaussian Mixture Model (GMM) and Kalman Filter (KF), as well as using Gated Recurrent Unit (GRU) in order to drop the number of variablesof RNN and parameters of hidden unit. Another method related to the sensor datasets described in [14][15] implemented the two-stream CNN + biLSTM architecture to achieve temporal signals, where global average pooling layers was used instead of FC layer for model parameters reduction.

Therefore, to solve this problem, our method uses the TSM module.The Time Shift Module (TSM) was proposed in [4] and has the advantage of combining modeling both spatial and temporal information.

As an example in the article [4], the authors describe the application of this module to understanding video, where a backbone is used to extract features from frames, and then the output is sent to TSM, where the spatiotemporal information of the frames is used for video classification. Comparing with other similar solutions such as TSN, TRN, ECO, I3d, which are also used for video motion recognition, the authors showed that this method shows the highest percentage of correct predictions, as shown in Fig. 2:

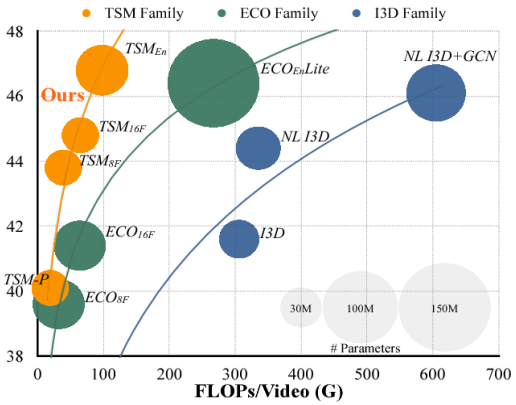

Figure 2: Comparison on different networks.

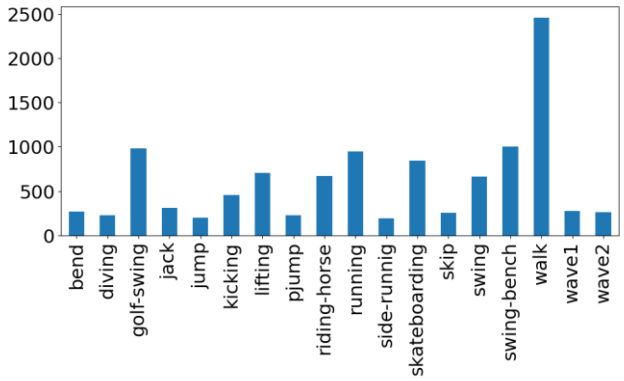

Figure 3: Class distribution in the dataset.

A similar method was used in [5], where TSM was modified and combined with ResNet50 as a backbone. The authors of this article fine-tune the model using the pretrained weights of the models that were trained on the Kinetics and ImageNet dataset. The authors also combine together similar frames based on the Hamming distance and the fact that these frames have the same gradients. One of the advantages of the method described in the article is the use of all frames for training the model, however, this approach is very demanding on memory and requires careful analysis of the frames.
Therefore, after reviewing the articles and taking memory costs into account, it was decided to focus on the original Temporal Shift module combined with ResNet as a base, as used in previous articles.

1.2. Dataset

For our dataset we combined UCF-Sports [6,7] and Weizmann [8,9] datasets. The datasets contained some videos for similar classes, therefore these videos (e.g. walk, run) were combined into a single class. UCF-Sports contains videos for the same action filmed from different angles. We labeled them with the same class. In total, there are 18 classes of actions with different durations, frame rates and resolutions.We needed to normalize the dataset so that the videos had the same format. To do this, we used ffmpeg [10] to downsample them to 10fps and downscale to the resolution of 180x144.
The final distribution of frames between classes is shown in Figure 3. The distribution shows that the dataset is highly imbalanced. This will be handled by setting each class an appropriate weight.

1.3. Baseline

For the baseline, we chose a simple per-frame CNN approach. For each video, the frames are split and evaluated separately. The model will produce an output for each frame, and then the model will choose the most appropriate label and output the final decision. The model consists of two parts - CNN and output evaluation.
The CNN part is a deep learning model consisting of a total of 3 deep learning layers. The input is passed into a 2D convolutional layer with 32 filters with dimensions of 3x3. The convolution layer uses a RELU activation to pass the output to the next layer. Then, a max-pooling layer is applied over 2x2 batches. The next layer is an FC layer with 128 nodes, followed by a dropout of rate 0.75. After that, there is another dense layer with 64 nodes. Before passing data to the output layer, it also has a dropout of rate 0.6. We needed such high dropout rates because for each video there are many frames, therefore there is a high probability that convolutional feature extractor might pick a background feature and overfit the model. The output layer is a dense layer with a corresponding number of nodes.
The output evaluation gets the probability distribution of each class for every frame. The model picks the most common guess out of all frames and outputs it as a final decision.
Since two datasets are too different we trained three baseline models to compare with our main approach. One model was trained on for both datasets, and the other two individually for each dataset.

1.4. Main approach

Overall, our goal is to implement the TSM [4] with several different convolutional neural network models including ResNet-101 and ResNeXT. The mentioned 2D CNN models are going to be pretrained on ImageNet dataset. They are going to be used to deal with spatial features on the video frames, while TSM will deal with both spatial and temporal features of a video. The output of the model is the

probability of each class of the dataset. The module and CNN architectures are described below.

1.4.1 Temporal Shift Module

Traditional 2D CNNs work without regard for the dimension T; as a result, no temporal modeling is performed. On the other hand, the Temporal Shift Module (TSM) moves the channels along the temporal dimension in both directions.
The convolution operation consists of shift and multiply-accumulation. The time dimension is shifted by ±1 while the multiply-accumulate is folded from time dimension to channel dimension. A uni-directional TSM is used to execute online video understanding because future frames cannot be shifted to the present for real-time online video understanding[4].
It is important to balance the model capacity for spatial feature learning and temporal feature learning. Therefore, the TSM can be placed in a residual block's residual branch. This type of shift can referred to as residual shift. Due to the fact that identity mapping allows for continued access to all of the data from the initial activation even after a temporal shift, residual shift can solve the impaired spatial feature learning issue [4].

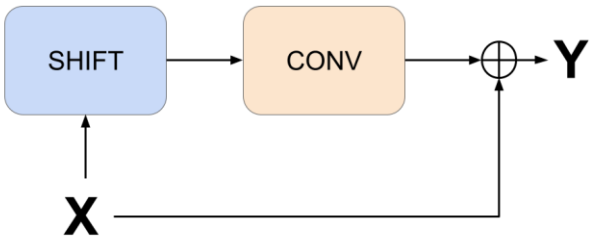

Figure 4, Residual Shift [4]

Initially, T frames F1,F2, …, FT are taken from a video V as a sample. Following frame sampling, 2D CNN baselines independently process each frame, averaging the output logits to produce the final forecast. The parameters and computational cost of the proposed TSM model are identical to those of the 2D model. The frames continue to operate independently during the inference of convolutional layers, just like in 2D CNNs. The distinction is that each residual block has a TSM injected into it, allowing for the computation-free unification of temporal information. The temporal receptive field will increase by 2 for each inserted temporal shift module, as if running a convolution with a 3 kernel size along the temporal axis.

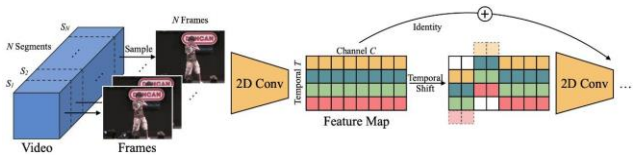

Figure 5. Architecture of TSM with 2D CNN

1.4.2. ResNet

ResNet is deep residual learning framework.In ResNet ,the few stacked layers were explicitly allowed to suit a residual mapping rather than assuming that each layer would match a desired underlying mapping directly.
The architecture of the model can be described as follows. Most convolutional layers contain 3x3 filters and adhere to two straightforward design principles: In order to maintain the time complexity per layer, (i) the layers have the same number of filters for the same output feature map size, and (ii) the number of filters is doubled for a feature map size reduction. Direct convolutional layers with a stride of 2 are used to downsample. A 1000-way fully-connected layer with softmax and a global average pooling layer make up the network's final layer. Moreover, identity shortcuts are introduced into the network when input and output have same dimensions [11].

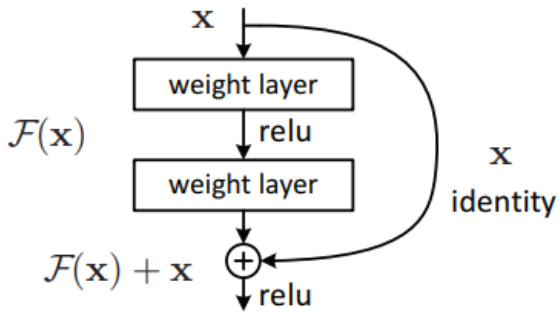

Figure 6. Building Block of ResNet [11]

The model which are going to be implemented is ResNet 101. It has the same architecture as described above with 101 convoluted layers making the model very deep. We believe that it will be the perfect spatial extractor and outperform the results obtained in [4].

1.4.3 ResNeXt

Another model that are going to be used along with the TSM to extract spatial features from video frames is RexNeXt. ResNeXt features split-transform-merge method adaptation, layer stacking, and shortcuts from one block to the next. Moreover, the authors included a new parameter name cardinality. It refers to the number of transformations in the set. The design consists of 32 identical topology

Table 1.
Results with accuracy of models which is tested on test sets.

Model	Backbone	Pre-train	Accuracy - Weizmann	Accuracy - UCF-Sports
TSM	ResNet18	Imagenet	63.333	-
TSM	ResNet34	Imagenet	63.333	-
TSM	ResNet50	Imagenet	61.655	-
TSM	ResNet101	Imagenet	75	-
TSM	ResNet152	Imagenet	67	-
TSM	ResNeXt50	Imagenet	50	-
TSM	ResNet18	Kinetics	85	69
TSM	ResNet34	Kinetics	90	70
TSM	ResNet50	Kinetics	80	68
TSM	ResNet101	Kinetics	80	75
TSM	ResNet152	Kinetics	85	75
TSM	ResNeXt50	Kinetics	55	72

blocks, making 32 the cardinality value. Less parameters are needed as new layers are added to this architecture because it uses the same topology [12].
In our case we will use ResNeXt-50. It has 3x3 max pool with stride 2. Output of the model is going to 56x56. The model is 50 layers of 32 grouped convolutions. This architecture allows better accuracy than ResNet 101 but also not going deeper. We prone that combining TSM with ResNeXt will increase accuracy with computation time of video classification.

1.4.4 Full Video Action Recognition
This model cluster all frame activations along the temporal dimension based on their similarity w.r.t. the classification task, followed by temporally aggregating the frames in the clusters into less number of representations. The concept of this method can be seen in Figure 6.

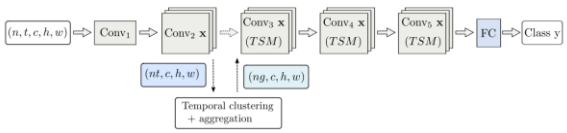

Figure 6. Architecture of main approach

1.5. Evaluation metric

To evaluate the results of our method, in the article will be using metrics such as accuracy, which uses correct video motion predictions divided by the total number of predictions, and running time to evaluate model performance. These metrics are the most common and are used in similar articles [4], [5], so it will be easier to compare the results with the authors' models. Moreover, one of the most intuitive way to observe the results of classification is confusion matrix which illustrate the comparison between groundtruth and predicted classes [14][15][16].

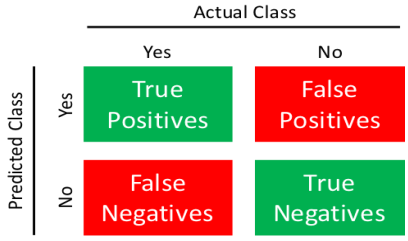

Figure 7. Concept of confusion matrix

1.6. Results & Analysis

1.6.1 Baseline

Expectedly, the baseline did not show good results, with a final accuracy on both sets 59%. While it could pick specific features for each frame, giving a good estimation for lifting and swinging on a bench, it did not identify actions containing similar poses (e.g. skipping and running). This confusion is illustrated in Figure 8.

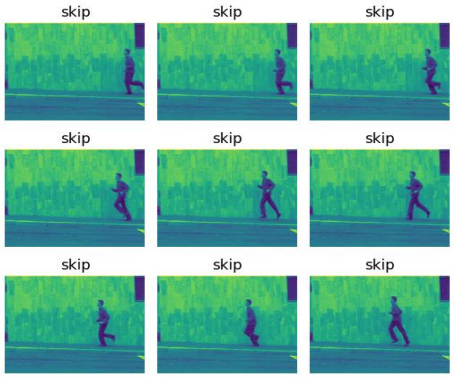

Figure 8. The running frames that are identified as skipping.

This simple baseline model can not differentiate between such videos, and therefore the action recognition requires a more complex approach.

To prove this hypothesis we run tests on two models that were trained on a separate dataset. As a result, the accuracy on the Weizmann dataset [8,9] was 25%, while for UCF-Sports [6,7] it was 58%. UCF-Sports is much more distinctive for CNN because the videos' background is usually specific for the action. Therefore, baseline CNN can differentiate between distinctive features on each frame, but fails when videos have the same features but different actions. Which means that despite good results for UCF-Sports, the model clearly overfits and will probably fail when classifying actions on a new background.

1.6.2 Main approach

Our model was trained on DGX server using multiple GPUs. Also, it can be noticed that our model performed training for each architecture with less than 1 hour.

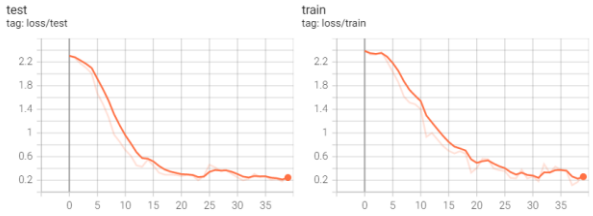

Figure 9. Train and val losses

In Table 1, we report the accuracy of our basic approach with various ResNet backbone architectures and TSM, which was tested on a test set. As we may observe from Table 1 the accuracy of Weizmann dataset is around 20% higher (60-75% vs 80-90%) using Kinetics pre-train in all ResNet architectures due to the size of Kinetics (~72M training frames in Kinetics vs. ~14M in Imagenet). Another reason why Kinetics pre-trained model depicts better results is that Kinetics dataset is the action recognition dataset (400 human action categories[17]), while Imagenet consists of diverse images. If we compare backbones of the model, we may notice that ResNet34 shows the highest result (90%) with Kinetics pre-train, while ResNet101 presents the highest result (75%) with Imagenet pre-train. In original paper of Weizmann dataset authors got 86.6% using MACH Filter for Vector Fields [18]. Another publication received 89.9% accuracy on Weizmann dataset using SVM [19].

According to the Table 1, the better accuracy for ucf-sports dataset we obtained using the TSM models with the ResNet-152, ResNet-101 as a backbone. So, we can state that in order to predict actions in this dataset, we need a deeper neural network with more layers. It was unexpected that the ucf-sport dataset performed worse than the models trained on the first dataset, but we attribute this to the imbalanced data in this dataset. Overall, we obtained better results than using simple baseline.

1.7. Error Analysis

The trained models showed higher results compared to the baseline. The distinction is more significant for models trained on the Weizmann [8,9] dataset. The best model for this dataset was ResNet34. However, due to the small number of videos in the dataset, we could not conduct a comprehensive error analysis using a confusion matrix. A sample confusion matrix is shown in the Figure 10. We could only include 2 videos of each class into the testing set, therefore the unit value in the matrix is 0.5. Despite the small amount of information in the figure, we can notice that the model might confuse jump-in-place with a hand-wave. It probably confuses them because in both cases there is a motion above the person's head. It also confuses running with walking, which is probably because every other motion does not occur steadily in a horizontal axis. And both running and walking might lay near each other because they both describe steady horizontal motion.

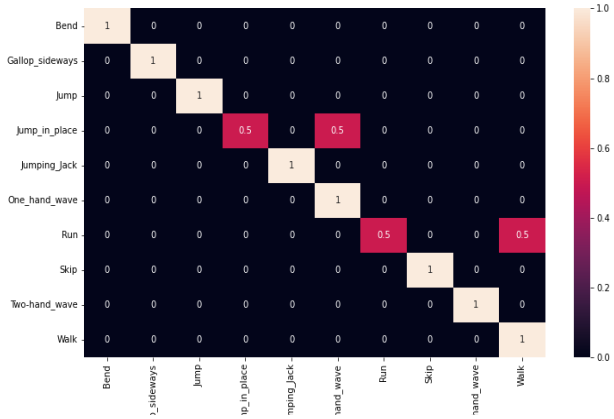

Figure 10. Confusion matrix for ResNet34 on Weizmann dataset.

The best models for UCF-sports dataset were ResNet-101, ResNet152. In Figure 11 we can see the confusion matrix of the ResNet-101,so here clearly see what is the problem of low accuracy. First, model predicted classes Golf-Swing-Side, Golf-Swing-Front as a Golf-Swing-Front and Kicking-Side as a Kicking-Front. These classes are similar action,but with different angles,so because of that for model it was hard to see the difference between actions. In addition, Kicking-Front and Walk-Front videos has other peoples in background and is we analyze in detail they walk,so because of the third persons performing the actions the model predicted wrong.
In Figure 12 we can see the confusion matrix of the ResNet-152. Thus, here we can see similar problems that were indicated earlier. It's also surprising that the Walk-Front was misclassified as a riding horse, we think it's because the person in the video is walking dogs and the model misclassified video because of that as a Riding-Horse class. As a result, we can observe that despite to the advantage of our model which are efficient training,because training takes less than hour, and full video understanding, it also has drawbacks such as distractibility to third-party objects and people. So, it will be better as the future work to work with this module and add bounding boxes or something like that to obtain information about several actions in one video.

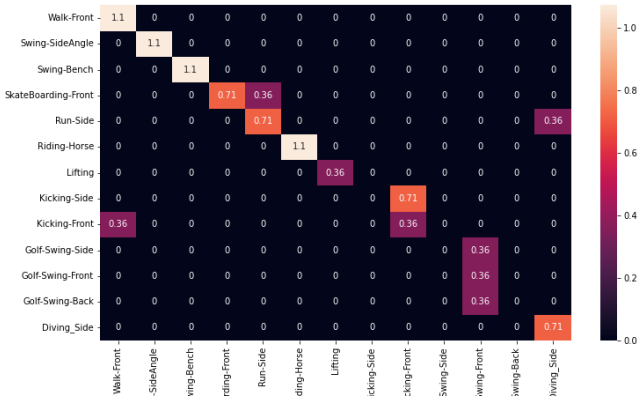

Figure 11. Confusion matrix for Resnet101 on UCF-sports dataset.

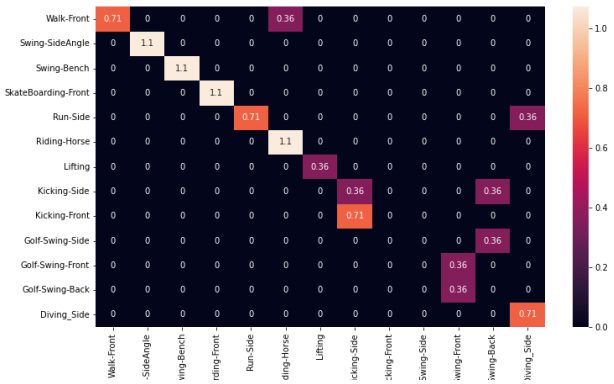

Figure 10. Confusion matrix for Resnet152 on UCF-sports dataset.

Conclusion

Due to the sheer volume of videos, researchers are looking for a solution to use fewer resources to train new models, one of the solutions we proposed in this article. Thus, to solve the problem of action recognition we used method which is rely on TSM model, and we implemented different ResNet topology as a backbone for TSM in order to improve the accuracy. It was shown that this model gave better results than using convolution layer and some cited paper which is used classifiers and filters. Overall, we obtained 90% and 75% accuracy for both datasets.
Through experiments, it was analyzed that this model has its drawbacks such as classifying a single movement in a video, so as a future work, it was possible to work on the output of several classes with bounding boxes in each video.

References

[1] Baccouche, M., Mamalet, F., Wolf, C., Garcia, C., Baskurt, A. (2011). Sequential Deep Learning for Human Action Recognition. In: Salah, A.A., Lepri, B. Human Behavior Understanding. HBU 2011. Lecture Notes in Computer Science, vol 7065. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-25446-8_4

[2]] J. Y.-H. Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga and G. Toderici, "Beyond short snippets: Deep networks for video classification", Proc. IEEE Conf. Comput. Vis. Pattern Recognit., pp. 4694-4702, Jun. 2015.

[3] X. Song, C. Lan, W. Zeng, J. Xing, X. Sun and J. Yang, "Temporal–Spatial Mapping for Action Recognition," in IEEE Transactions on Circuits and Systems for Video Technology, vol. 30, no. 3, pp. 748-759, March 2020, doi: 10.1109/TCSVT.2019.2896029.

[4] X. Liu, S. L. Pintea, F. K. Nejadasl, O. Booij, and J. C. van Gemert, "No frame left behind: Full video action recognition," 2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2021. https://doi.org/10.48550/arXiv.2103.15395

[5] J. Lin, C. Gan, and S. Han, "TSM: Temporal shift module for efficient video understanding," 2019 IEEE/CVF International Conference on Computer Vision (ICCV), 2019.

[6] Mikel D. Rodriguez, Javed Ahmed, and Mubarak Shah, Action MACH: A Spatio-temporal Maximum Average Correlation Height Filter for Action Recognition, Computer Vision and Pattern Recognition, 2008.

[7] Khurram Soomro and Amir R. Zamir, Action Recognition in Realistic Sports Videos, Computer Vision in Sports. Springer International Publishing, 2014.

[8] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri, "Actions as Space-Time Shapes," Transactions on Pattern Analysis and Machine Intelligence, vol. 29, no. 12, pp. 2247–2253, Dec. 2007. [2]M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri, "Actions as Space-Time Shapes," in The Tenth IEEE International Conference on Computer Vision (ICCV'05), 2005, pp. 1395–1402.

[9] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri, "Actions as Space-Time Shapes," in The Tenth IEEE International Conference on Computer Vision (ICCV'05), 2005, pp. 1395–1402.

[10] S. Tomar, "Converting video formats with FFmpeg," Linux Journal, vol. 2006, no. 146, p. 10, 2006.

[11] He, K., Zhang, X., Ren, S. and Sun, J., 2016. Deep residual learning for image recognition. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 770-778).

[12] Xie, S., Girshick, R., Dollár, P., Tu, Z. and He, K., 2017. Aggregated residual transformations for deep neural networks. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 1492-1500).

[13] Tang, Y., Teng, Q., Zhang, L., Min, F., & He, J. (2020). Layer-wise training convolutional neural networks with smaller filters for human activity recognition using wearable sensors. *IEEE Sensors Journal*, *21*(1), 581-592.

[14] Nafea, O.; Abdul, W.; Muhammad, G.; Alsulaiman, M. Sensor-Based Human Activity Recognition with Spatio-Temporal Deep Learning. *Sensors* **2021**, *21*, 2141. https://doi.org/10.3390/s21062141

[15] K. Xia, J. Huang and H. Wang, "LSTM-CNN Architecture for Human Activity Recognition," in *IEEE Access*, vol. 8, pp. 56855-56866, 2020, doi: 10.1109/ACCESS.2020.2982225.

[16] Jaouedi, N., Boujnah, N., & Bouhlel, M. S. (2020). A new hybrid deep learning model for human action recognition. *Journal of King Saud University-Computer and Information Sciences*, *32*(4), 447-453.

[17]*Kinetics Pretrained Models*. (n.d.). http://yjxiong.me/others/kinetics_action/

[18] M. D. Rodriguez, J. Ahmed, and M. Shah, "Action mach a spatio-temporal maximum average correlation height filter for action recognition," *2008 IEEE Conference on Computer Vision and Pattern Recognition*, 2008.

[19] P. S. Dhillon, S. Nowozin, and C. H. Lampert, "Combining appearance and motion for Human Action Classification in videos," *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2009.