What Makes a Video a Video: Analyzing Temporal Information in Video Understanding Models and Datasets

De-An Huang[1], Vignesh Ramanathan[2], Dhruv Mahajan[2],
Lorenzo Torresani[2,3], Manohar Paluri[2], Li Fei-Fei[1], and Juan Carlos Niebles[1]
[1]Stanford University, [2]Facebook, [3]Dartmouth College

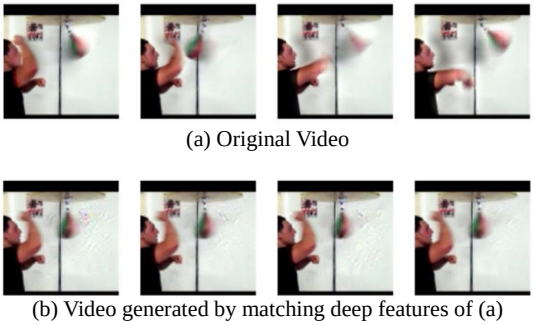
Abstract

The ability to capture temporal information has been critical to the development of video understanding models. While there have been numerous attempts at modeling motion in videos, an explicit analysis of the effect of temporal information for video understanding is still missing. In this work, we aim to bridge this gap and ask the following question: How important is the motion in the video for recognizing the action? To this end, we propose two novel frameworks: (i) class-agnostic temporal generator and (ii) motion-invariant frame selector to reduce/remove motion for an ablation analysis without introducing other artifacts. This isolates the analysis of motion from other aspects of the video. The proposed frameworks provide a much tighter estimate of the effect of motion (from 25% to 6% on UCF101 and 15% to 5% on Kinetics) compared to baselines in our analysis. Our analysis provides critical insights about existing models like C3D, and how it could be made to achieve comparable results with a sparser set of frames.

1. Introduction

Video understanding has progressed significantly in recent years with the introduction of better models [31, 36, 43] and larger datasets [14, 19, 20]. A common theme among most approaches is the emphasis on temporal modeling, which is seen as the main difference between videos and images. This includes works on low-level motion [31, 36, 41, 42], long/short term dependencies [5, 39, 47, 50], temporal structure [3, 8, 9, 10], and modeling the action as a sequence of events/states [33, 34, 45].

More specifically, a broad array of deep learning architectures [4, 36, 39] which attempt to capture low-level motion through temporal convolutions achieve state-of-the-art results [4, 37]. Hand-crafted features like iDT [41] have also advocated using motion for action recognition. However, the actual impact of modeling low-level motion remains unclear. As seen in Fig. 1, one could argue that the scene and objects in a frame are almost sufficient to deduce the action. Recreating the motion in a video by matching deep features from a C3D [36] model partly validates this conjecture. We observe that the visible motion in the video is lost at pool-5 layer in the network, while still preserving the full spatial information. Motivated by such observations, we conduct an in-depth quantitative and qualitative analysis of the effect of motion in video action recognition.



(a) Original Video

(b) Video generated by matching deep features of (a)

Figure 1. (a) The original video. (b) Video generated by network visualization [6] from the C3D pool5 features of the video in (a). The network loses visually perceptible motion as early as pool5.

In particular, we try to analyze if an existing model trained on videos utilizes motion information while classifying a new video. We could achieve this by drastically subsampling videos during testing, to the extent of retaining just a single frame. However, testing a model trained with full length videos on a single frame is non-trivial. A naive way of replicating the frame multiple times results in almost 25% performance drop on UCF-101. It is difficult to conclude that this is just due to the lack of *motion*. We observe that in addition to removing motion, subsampling results in two undesired artifacts: (i) significantly alter the temporal distribution, and (ii) potentially remove critical frames in the video that are important for recognizing the action.

We propose the following two frameworks to ablate the motion in a video for analysis while mitigating these undesired effects : (i) *class-agnostic temporal generator* that adds a temporal variance to the subsampled frames to bridge

the gap between training and testing (ii) *motion-invariant frame selector* that allows the model to choose good frames from the video by looking at each frame independently.

We exemplify our analysis on the widely used 3D convolution model [36, 39] on two video datasets: UCF101 [32] and Kinetics [20]. UCF101 has been the standard benchmark for comparing and analyzing video models [43] and Kinetics is the most recent large-scale dataset designed for classification. We choose 3D convolution because it has become a standard approach for video understanding, but the proposed frameworks (generator and frame selector) are general and can be used to analyze any video model.

Our analysis shows that, without using any motion from the video, and without changing the video model that we are analyzing, we are able to close the gap from 25% to 6% on UCF101 and 15% to 5% on Kinetics. This provides a much tighter upper bound for the effect of motion in the video compared to other analysis baselines. Our per class accuracy breakdown shows that over 40% of the UCF101 and 35% of the Kinetics classes do not require motion in the video to match the average class accuracy. In addition, retaining just 1/4 of the frames in a clip, we are able to achieve results comparable to that obtained by using all frames.

2. Related Work

Temporal modeling for action recognition: The emphasis on modeling the temporal information in a video has been the key difference between video and image models. This includes low-level motion [7, 36, 31, 41, 42, 16, 17], long/short term dependencies [39, 50, 5, 47, 26], temporal structure [9, 8, 3, 23], modeling the action as sequence of events/states [34, 45, 33, 29] and temporal pooling strategies [44, 52, 48, 10]. These methods are often evaluated based on overall performance, making it difficult to determine whether the models are really capturing motion information, and if motion is really critical for recognizing action in existing video datasets [1, 13, 14, 19, 20, 22, 32]

Model analysis: The most related to our work is the recent analysis of action categories by Sigurdsson *et al*. [30], where the recognition performance is analyzed by breaking down action categories based on different levels of object complexity, verb complexity, and motion. They attempt to answer questions regarding the choice of good action categories to learn effective models. In contrast, our work provides a data-driven approach to explicitly measure the effect of motion in temporal action recognition models like C3D. Similar ideas have been used in the past to analyze models for object detection [15, 28]. Another related line of work is the visualization of representations from deep neural networks [2, 46, 51, 53], and bias in datasets [21, 35].

Generator: In order to properly analyze motion, we use a temporal generator to offset the differences in training and testing video temporal distribution. The generator is related to works in video prediction [24, 38, 40], and our architecture is inspired by recent image transformation approaches [18, 54]. It is worth noting that generators have been used as a way to analyze the shortcoming of deep networks in an adversarial setting [11, 25].

Frame selection: Frame selection to narrow down the temporal extent of an action before recognition has proven to be an effective approach for improving the performance of video models [27, 49, 55]. We leverage this idea to analyze the effect of choosing the right frame while subsampling videos to reduce motion.

3. Approach

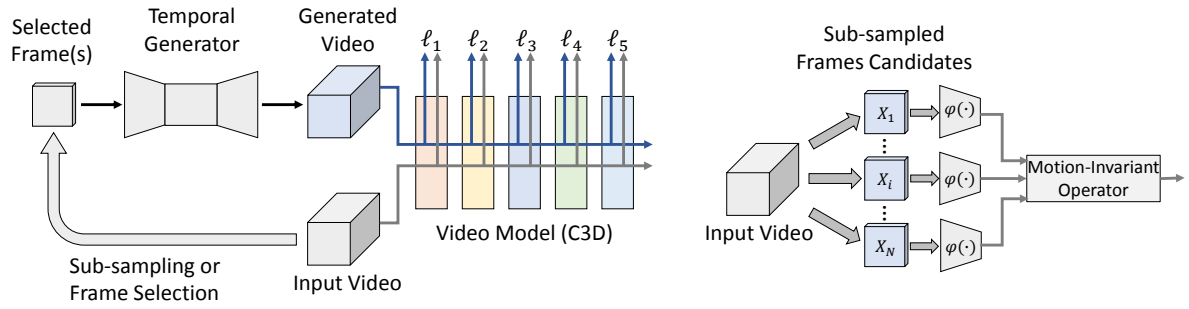
Our goal is to analyze the impact of motion on the performance of an existing model trained on videos (*e.g*. C3D trained on UCF101). The key challenge is that factoring out the motion in an existing model using simple strategies (*e.g*. replication) may lead to wrong or biased conclusions. We propose two frameworks that address this issue and allow us to accurately analyze the contribution of motion to recognition performance without modifying the model we are analyzing. We show later in Section 4 that the combination of the two provides a much tighter upper bound on the contribution of the motion information.

3.1. Class-Agnostic Temporal Generator

As discussed earlier in Section 1 and Figure 1, for many examples a single or a sparse number of frames might have sufficient information to recognize the action. However, since the model is trained on the full video (of 16 frames), the spatial and temporal dimensions are entangled in the model. In this case, naively subsampling the frames at the time of analysis/testing significantly alter the temporal distribution and affect the recognition performance.

We observe that spatial and temporal dimensions are highly correlated: It should be possible to hallucinate the motion from the subsampled images to compensate the difference in the temporal distribution. We propose a class-agnostic temporal generator (Figure 2(a)) that takes as input a subset of frames of the video and synthesize the full video, which serves as the input to the model. This makes the train and test distribution similar, which in turn allows us to analyze the effect of motion by doing frame sampling. We do not provide any additional motion information about the particular video that we aim to classify.

The challenge becomes, what should be the properties of the synthesized video? Do we have to accurately synthesize the last three frames from the first frame in Figure 1(a) for our analysis? Our answer is: No. As shown by network visualization work in [6], the convolutional neural network has strong invariance in higher layers in the hierarchy. For the purpose of analysis, as long as we can generate motion prior that recovers the desired feature activation in



(a) Class-Agnostic Temporal Generator **(b) Motion-Invariant Frame Selector**

Figure 2. We propose two frameworks for analyzing the video model. The goal is to ablate the effect of other information so as to make a meaningful conclusion on the impact of the motion information. The first is class-agnostic temporal generator, which offsets the difference in temporal distribution between video and sub-sampled frames. The second is motion-invariant frame selector, which introduce no additional motion information, but allows the video model to look at all the frames in the video.

the model, it can be used to provide a tighter upper bound in our analysis. Figure 1(b) shows our network visualization result of matching the pool5 feature of C3D trained on UCF101. We observe that the visible motion in the video is lost at the pool5 layer in the network.

Based on this observation, we use the perceptual loss [18] to match the features at different layers of the video model. In other words, our generator aims to generate motion from the given sub-sampled frames to reconstruct the features in each layer to compensate the difference in temporal distribution. The outline of our temporal generator is shown in Figure 2(a). We extend the generator of CycleGAN [54] to generate a video clip (16 frames for C3D) from a given amount of frames (1, 2, 4, or 8 frames in our experiments). We use the normalized L2 distance between the feature maps of synthesized video and the original video as the loss function. We will show that the perceptual loss plays an important role in the success of our generator to provide a tighter upper bound in our analysis. Note that we are doing an unsupervised training: class labels or supervised loss are not used at all for training the generator. This potentially allows us to leverage the abundance of unlabeled video data. In addition, our generator provides a way of qualitatively analyzing the video model. By visualizing the motion we learn from each network, we are able to understand what motion it sees in the video. Finally, note that the framework is generic and not tightly coupled with the video model we are trying to analyze. We simply need to specify the layers to define the perceptual loss.

3.2. Motion-Invariant Frame Selector

In the previous section, we proposed an approach to analyze motion given a subset of frames. We now try to answer the question: To what extent can the quality of the frames affect the performance? Taking it to an extreme, is there a single key frame that is sufficient for good accuracy? Naively sub-sampling the video frames can remove visual content important to understand the video. Potentially, there might exist a key frame that is crucial to recognize the action of the video without any additional motion.

As we are focusing on analyzing the temporal information, the frame selection process should not use extra motion information that is only available in the video we aim to classify. In other words, it is important to make sure that the frame selector is *motion-invariant*. Formally, given a set of candidate frames $\{X_i\}$ sampled from the video, the selection process should not introduce any order/motion information that is beyond each of the candidate X_i. We now briefly describe two simple, heuristics based frame selectors: *Max Response* and *Oracle*.

Max Response: Given a set of candidates $\{X_i\}$, and a predefined response function $\phi(\cdot)$, pick the candidate with the highest response $i^* = \text{argmax}_i \phi(X_i)$. Note that since *argmax* is order-invariant, so is the selector. The quality of the selector depends on the definition of the response function $\phi(\cdot)$. Ideally, it is possible to learn this response function to maximize the recognition performance without using extra motion information from the video of interest. In our experiments, we define $\phi(X_i)$ as the maximum classification score of all the class, X_i is assigned to, after applying the generator and the video model. Formally, $\phi(X_i) = \max_c f_c(X_i)$, where $f_c(X_i)$ is the probability of X_i being classified as action class c, i.e., the response of softmax layer of the video model for class c. In other words, choose a frame that is most confident about its prediction.

Oracle: The oracle selector looks at the ground truth class label of the video to select the candidate frames that can actually give the correct result (only misclassify when no selection gives correct prediction). Note that unlike *Max Response*, it is *not* a valid frame selector and involves a "cheat". However, it is still motion-invariant and provides an upper bound for the performance of frame selectors that do not use extra motion information.

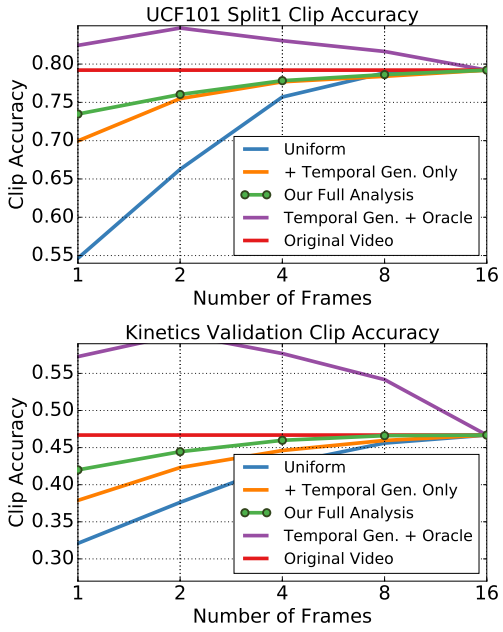


Figure 3. Analysis of UCF101 and Kinetics. Our full analysis provides a much tighter upper bound (6% for UCF101 and 5% for Kinetics) on the effect of motion in the video. This is much more meaningful conclusion than the naive approach, which provides 25% upper bound for UCF101 and 15% for Kinetics.

4. Analysis

So far we have discussed the use of class-agnostic temporal generator and motion-invariant frame selector for the ablation study of motion in video by reducing motion in the test video while being careful not to introduce other artifacts. By bridging the difference in temporal distribution between the train and test setups with our temporal generator and selecting the right frames without using additional motion information, we provide a tight upper bound on the effect of motion for action recognition. First, we discuss our main analysis on two standard video datasets: UCF101 and Kinetics. Next, we analyze the effect of temporal generator in Section 4.2 and frame selection in Section 4.3.

Video Model and Datasets. We demonstrate our analysis for the 3D convolution architecture by Tran *et al*. [36]. Note that our framework is not specific to the video model and can be easily extended to other architectures. We use two datasets for our analysis. The first is UCF101 [32] which consists of 101 action categories and 13,320 videos. We analyze the split 1 of the dataset following recent works [37] due to the computation cost. The second dataset is Kinetics [20], which consists of 306,245 videos from 400 action classes. We report analysis on the validation set.

Experimental Setup. We use the C3D model [36] pretrained on Sports1M [19] for our analysis. For UCF101, we train the original video model using the hyperparameters from the official C3D implementation and obtain comparable numbers. For Kinetics, we increase the learning rate to 0.001 and retain the same hyperparameters. For the temporal generator, we use the architecture by Zhu *et al*. [54], starting with C64 layers. We trained the model on the same training set as the video model. It is important to note that the generator is class-agnostic and trained *without* any supervised label and can be trained on abundant large-scale video dataset readily available. Empirically, we did not find a significant impact on performance when we used a different dataset (*e.g*. generator trained on Kinetics while analyzing UCF101) for training the generator. For the motion-invariant frame selector, we use the max response selector on the confidence score as discussed in Section 4.3. As the exact enumeration of all possible combination of frame selections is computationally too expensive (1820 ways of choosing 4 frames from 16), we restrict ourselves to 48 uniformly sampled frame selections for all reported numbers. We use clip-level action recognition accuracy for 16 frames clips as the metric for our analysis, to factor out the effect of video-level pooling and focus on the low-level motion. We verify that our video model has the same video-level accuracy reported in the original papers [20, 36].

4.1. Analyzing Motion Information

The clip accuracy obtained by varying the number of frames, and thus varying the amount of motion for the videos in UCF101 and Kinetics datasets are shown in Figure 3. "Uniform" is the baseline of naively sub-sampling the frames. "+ Temporal Gen Only" further incorporates our temporal generator. "Our Full Analysis" includes both the generator and the max response frame selector. The performance of the "Original Video" model is shown as reference. We also show the upper bound performance of an oracle frame selector with our temporal generator in "Temporal Gen + Oracle". We can observe from the results that:

Our framework provides a tighter upper bound. It can be seen from Figure 3 that naively removing all the temporal information by sampling a single frame out of the 16 frames leads to a drastic drop in performance (54% compared to 79% for UCF101, and 31% compared to 47% for Kinetics). With our proposed class-agnostic temporal generator and motion-invariant frame selector, we are able to close the gap (from 25% to 6% for UCF101, and 15% to 5% for Kinetics) without using additional motion information from the video, and more importantly without modifying/finetuning the video model. This provides a much tighter upper bound on the effect of motion in the given model trained on UCF101 or Kinetics. In summary, C3D trained on Kinetics relies more on the motion in the video (5% out of 47% accuracy) and benefits more from the frame selection process. On the other hand, C3D trained on UCF101 uses less motion information from the video (6%

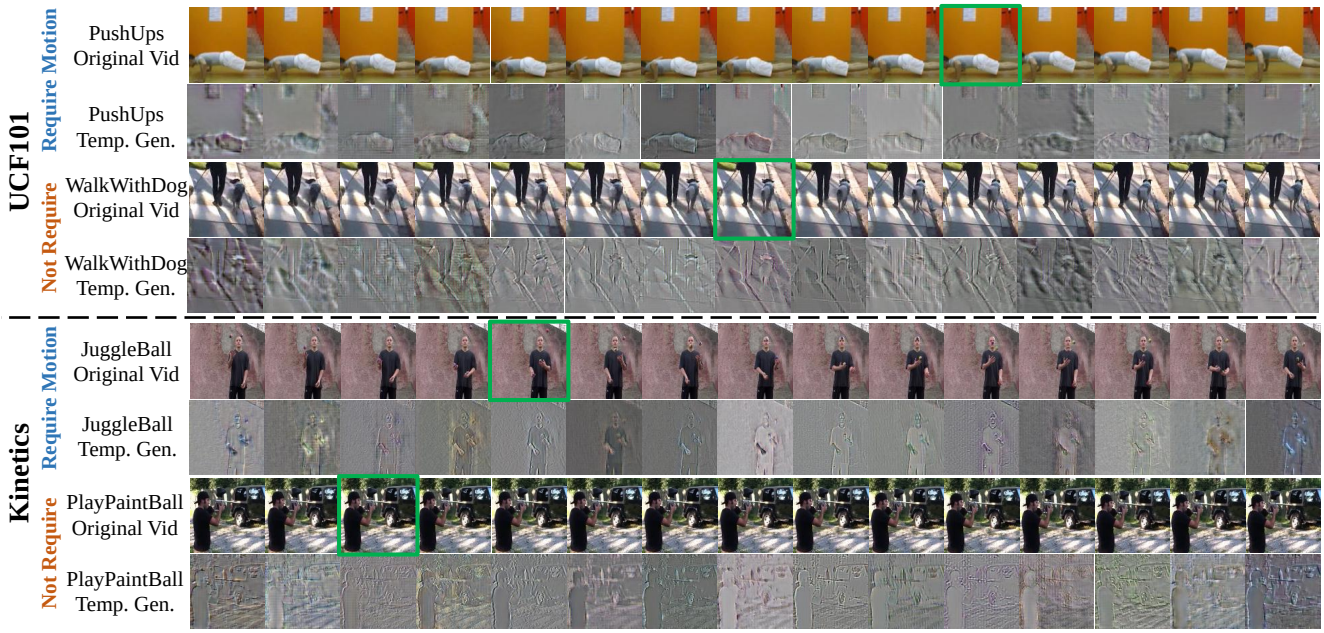


Figure 4. Qualitative results of classes needing the most/least motion from the video. Temp. Gen. rows are motion of our generated video. For both datasets, we do not need the motion to recognize action that can be identified by the salient object (*i.e.* dog in WalkWithDog). On the other hand, while our temporal generator can accurately hallucinate movement around the critical area to bridge the temporal distribution, PushUps in UCF101 and JuggleBall in Kinetics still require further motion from the video to be recognized. Green box indicates the selected frame by our max-response selector. The motion is generated by *only* looking at the single image selected.

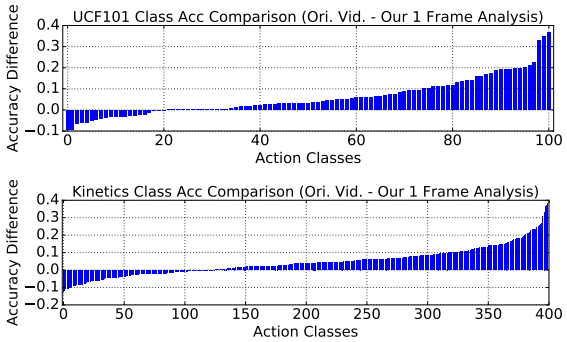


Figure 5. UCF101 and Kinetics class accuracy comparing original video model with our analysis model. For around 40% of the classes in UCF101 and 35% in Kinetics, we can achieve similar performance ($< 1\%$ difference) without motion from the video.

out of 79% accuracy), and the drastic accuracy drop comes mainly from the distribution shift, which is mostly bridged by our temporal generator. We provide more detailed analysis in Section 4.2 and Section 4.3 to identify the contribution of each component of our framework for both the datasets.

Some classes do not use motion. Figure 5 breaks down the 6% and 5% upper bound of motion for UCF101 and Kinetics into per class accuracy. For around 40% in UCF101 and 35% of the classes in Kinetics, we have already closed the gap with the proposed frameworks without using motion from the video and without modifying the model. This indicates that C3D did not learn to use motion to classify these classes. In particular, "Walking With Dog" from UCF101 and "Playing Paintball" from Kinetics are the classes where the motion in the video of interest is least important for the C3D model. As shown in Figure 4, our generated video is similar to the static image in this case.

Some classes use motion. On the other hand, there are classes that C3D learns to use motion from the video beyond our approach. In particular, "PushUps" in UCF101 and "JuggleBall" in Kinetics are the classes that use the most motion from the video in our analysis. However, our frameworks have already significantly improved the performance on both of the classes ($+25\%$ for PushUps and $+17\%$ for JuggleBall), it is just that the actions still require more motion from the video. For example, the motion of the ball in "JuggleBall" is subtle but plays an important role in identifying the action. As shown in Figure 4, our temporal generator accurately hallucinates movement in the critical areas around the person of interest, to bridge the distribution difference between the video and the sub-sampled frames.

We don't need the entire clip. As shown in Figure 3, the performance with 4 frames in our analysis is comparable to the original video on both UCF101 and Kinetics. This indicates the possibility of a 4-frame based model that focuses on a smaller temporal support. This is in contrast to recent observations of using longer temporal support for 3D convolution [39]. We conjecture that longer temporal support

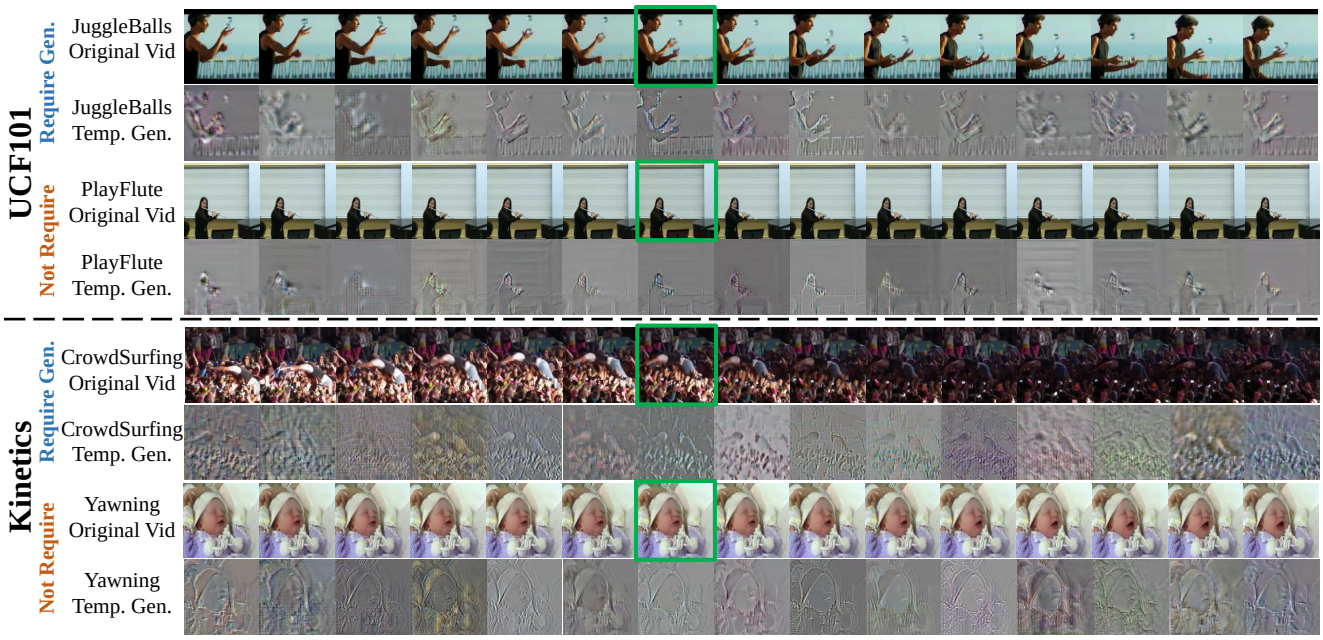


Figure 6. Qualitative results of the most effective/ineffective action classes for our class agnostic temporal generator. Temp. Gen. rows depict motion predicted by our generator from the single frame highlighted in green. For classes like JuggleBall and CrowdSurfing, it hallucinates movements around critical regions where we anticipate motion. On the other hand, for more static action like PlayFlute in UCF101, the generated motion is very subtle. In the Yawning action, our generator produces movements that are not specific to the action.

is beneficial for selecting better frames (which could be an internal side-effect of using C3D) but not necessarily capturing fine-grained motion.

Frame selection is important. Continuing the discussion that we do not need the full clip for recognizing the action, we show that if we have an oracle for picking the frame that can provide the correct action class, the resulting performance outperforms the original video model when combined with our temporal generator. The effect is especially significant in the Kinetics dataset. The upper bound of oracle single frame selection is 11% higher than the original model. This suggests that a good frame selection model can go a long way in boosting the action recognition performance. However, it can be challenging or even impossible to obtain good frame selection without using additional motion information when the ground truth label is not available. Nevertheless, for the purpose of our analysis we note that the oracle frame is still motion-invariant when ground truth action label is available. A more in-depth discussion will be provided in Section 4.3.

Importance of temporal generator. As can be seen from the results, temporal generator significantly reduces the gap between the original video and the sub-sampled frames for both datasets. The difference is especially significant for C3D trained on UCF101 (reduced to 9% from 25%). Figure 6 shows the generated temporal motion from a single frame. As can be seen from the figure, our model is able to hallucinate patterns around the person, although not entirely reconstructing the exact video. This is consistent with our observation from the network visualization in Figure 1. We perform further analysis on the synthesized temporal information in the next section (Section 4.2).

We have shown that the combination of our temporal generator and frame selector can lead to a more meaningful data-driven way of analyzing a video model without changing the model weights. Next, we provide more detailed discussion of the individual components.

4.2. Analyzing Class-Agnostic Temporal Generator

The goal of our class-agnostic temporal generator is to bridge the distribution gap between the original video and the sampled frames to provide a more accurate analysis on the effect of motion. We have shown that our temporal generator leads to around 16% improvements in UCF101 and 6% in Kinetics. We further analyze the gains we achieved with the temporal generator, and compare two different loss functions for training the temporal generator.

Perceptual loss is important. One approach for bridging the distribution difference is to train a generator which can directly predict the pixel values of the other frames in the video from sub-sampled frames. This is directly related to the future frame synthesis problem [24, 38, 40], which has shown to be a challenging task on its own. We argue that we do not need to solve this challenging problem to im-

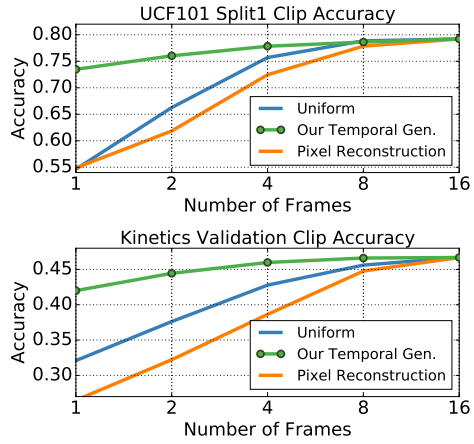


Figure 7. Our temporal generator results for UCF101 and Kinetics. While the main contribution of our work is providing a thorough analysis of motion in video models and datasets, it is our contribution to propose this temporal generator to offset the difference in distribution to provide meaningful analysis of motion.

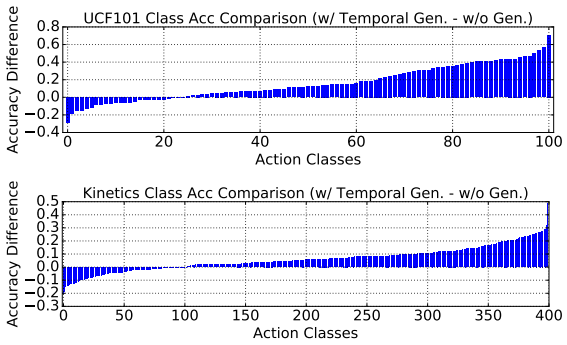


Figure 8. UCF101 and Kinetics class accuracy comparing models with/without our temporal generator. For over 75% and the classes in both datasets, our temporal generator is able to improve the performance and provides better analysis.

prove our analysis of video models. Our key observation is that the network exhibits some level of invariance to the pixel space as shown in the example in Figure 1. Therefore, we propose to learn the temporal generator with the perceptual loss [18], to directly optimize what is perceived by the video model. We observe that this approach can successfully recover the motion agnostic performance lost by sub-sampling the frames. Figure 7 shows the comparison of pixel reconstruction loss with the proposed temporal generator. It can be seen that our approach significantly improves the upper bound estimation of the effect of motion.

Distribution shift is critical for most classes. We further break down the improvements from our class-agnostic temporal generator by action classes. The results are shown in Figure 8. Our temporal generator successfully offsets the temporal distribution difference on 77% of the UCF101 classes and 75% of the Kinetics classes. In particular, the effect is most significant in "JugglingBalls" of UCF101 and "SurfingCrowd" of Kinetics. As shown by their examples in Figure 6, our temporal generator is able to anticipate movements in critical areas around the person of interest. Interestingly, "JugglingBalls" is also the class in Kinetics that also needs further motion from the video. On the other hand, our temporal generator is less helpful for more static classes like "PlayingFlute" in UCF101. It is important to note that our temporal generator is trained without the action label and uses no additional motion information from the video, so it can wrongly hallucinate motion that is not helpful for discriminating classes. The "Yawning" action of Kinetics in Figure 6 is an example.

4.3. Analyzing Frame Selection

The goal of our motion-invariant frame selector is to enable the model to look at all the frames in the video while obtaining no additional motion information. We have shown that this is an effective approach for our analysis (4% gain on both datasets). Now we further visualize and discuss the results of frame selection.

Max response selector can avoid noisy frames. While in both datasets the max response selector is able to improve the performance by 4%, this is more significant for the Kinetics as the original accuracy is lower. In Figure 9 we visualize classes where the max selector gives maximum improvement compared to using just the temporal generator. It can be seen that the ones in the UCF101 have more static appearance across the frames, while the frames in the Kinetics can be drastically different. In particular, the max selector is able to avoid the empty scene in the middle of the "IceSkating" clip and the "SledDogRacing" clip.

Oracle frame selector outperforms original video. One interesting observation from Figure 3 is that the oracle selector, when combined with our temporal generator, provides a model that outperforms the original video model. However, it is possible that this cannot be easily achieved without using the motion information in the video. We visualize in Figure 10, the classes that gain the most from the oracle frame selection. Quite surprisingly, we do not find salient visual features in the "oracle" frames that are distinguishable by the human eye. While this could be an effect related to adversarial examples [12], we believe that there should be a systematic way of leveraging this effect. In addition, we do see promising results from the max selector, which shows the need for latching on to the correct frames in videos with large appearance variations across frames.

5. Future Research Directions

In the previous section, we presented an in-depth analysis of the performance gain achieved by modeling motion

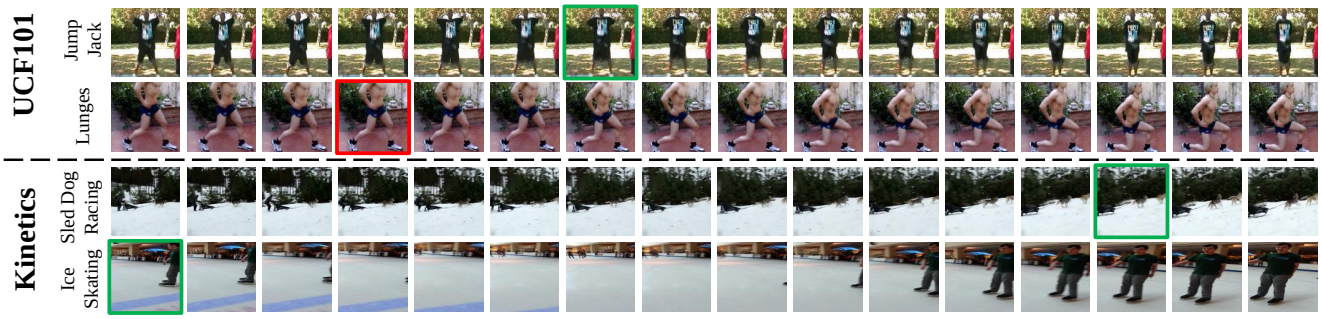


Figure 9. Qualitative results of max response selector. Our motion-invariant selector allows the model to look at the frames for better prediction while obtaining no extra motion information. In particular, our max selector is able to avoid the empty scene in the middle of the "IceSkating" clip and the "SledDogRacing" clip. (Green for correct selection, red for incorrect selection)

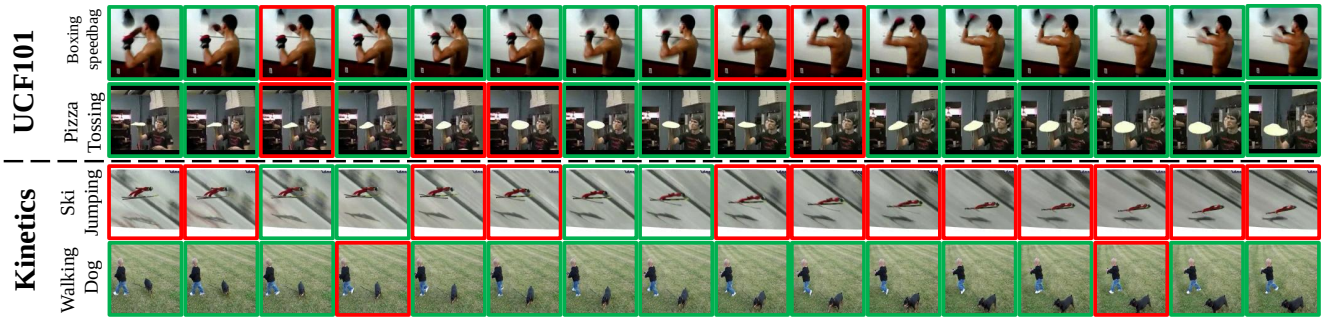


Figure 10. Qualitative results of oracle selector. Interestingly, oracle selector with our temporal generator is able to outperform the original model. However, when visualizing the selection, it is unclear visually what contributes to the difference. We believe this could be an important direction for further investigation. (Green for correct selection, red for incorrect selection)

in video models. Our finding could be useful in informing future research directions for building better video models as well as dataset. We discuss a few briefly:

Motion specific datasets. Based on our analysis, the C3D model trained on UCF101 and Kinetics does not learn to utilize motion for recognizing a significant number of action classes. An analysis framework like ours could be used to identify and build video datasets where the model is required to learn to use motion for better performance. While recent papers [30] have tried to analyze the effect of action classes, we believe that a quantitative study like ours can lead to systematic creation of video datasets, where the effect of motion is more dominant.

More efficient video model. Even for classes requiring motion, we have shown that the trained C3D does not need the full video for recognition. This has two implications. First, 3D convolutional models [36] need fewer than 16 frames and can be made computationally more efficient. Second, while working on a restricted computational budget, it might be worthwhile to investigate deeper architectures while reducing computation on temporal modeling.

Key frame selection. We show that selecting the right frames from the video can lead to huge gains over the original model. It is possible that apart from modeling low-level motion, existing models like C3D are inherently selecting the key frames. While this area holds promise, there are many open questions: How hard is this key-frame selection problem? Is temporal information from the video required to select these key frames? Is attention mechanism a good choice for selecting key frames in an end-to-end fashion?

6. Conclusion

We propose two frameworks to analyze the effect of motion: (i) class-agnostic temporal generator, and (ii) motion-invariant frame selector. This enables us to more accurately bound the impact of motion in C3D trained with UCF101 to 6% out of the 79% accuracy, and 5% out of 47% accuracy on Kinetics. Our analysis shows that the temporal distribution shift constitutes a larger role (16% of the accuracy) in UCF101, while frame selection is important for Kinetics. Interestingly, the oracle frame selector can actually outperform the original model. We have provided in-depth quantitative and qualitative analysis of the video model with general analysis frameworks that can be applied elsewhere. We believe our analysis of motion is critical to design better models and collect better datasets in the future.

Acknowledgement. We thank Du Tran for helpful discussion and feedback on our analysis and implementation.

References

[1] S. Abu-El-Haija, N. Kothari, J. Lee, P. Natsev, G. Toderici, B. Varadarajan, and S. Vijayanarasimhan. Youtube-8m: A large-scale video classification benchmark. *arXiv preprint arXiv:1609.08675*, 2016.

[2] D. Bau, B. Zhou, A. Khosla, A. Oliva, and A. Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *CVPR*, 2017.

[3] H. Bilen, B. Fernando, E. Gavves, A. Vedaldi, and S. Gould. Dynamic image networks for action recognition. In *CVPR*, 2016.

[4] J. Carreira and A. Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. *arXiv preprint arXiv:1705.07750*, 2017.

[5] J. Donahue, L. Anne Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *CVPR*, 2015.

[6] A. Dosovitskiy and T. Brox. Generating images with perceptual similarity metrics based on deep networks. In *NIPS*, 2016.

[7] C. Feichtenhofer, A. Pinz, and R. Wildes. Spatiotemporal residual networks for video action recognition. In *NIPS*, 2016.

[8] B. Fernando, E. Gavves, J. Oramas, A. Ghodrati, and T. Tuytelaars. Rank pooling for action recognition. *IEEE TPAMI*, 39(4):773–787, 2017.

[9] B. Fernando, E. Gavves, J. M. Oramas, A. Ghodrati, and T. Tuytelaars. Modeling video evolution for action recognition. In *CVPR*, 2015.

[10] R. Girdhar, D. Ramanan, A. Gupta, J. Sivic, and B. Russell. ActionVLAD: Learning spatio-temporal aggregation for action classification. In *CVPR*, 2017.

[11] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio. Generative adversarial nets. In *NIPS*, 2014.

[12] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. In *ICLR*, 2015.

[13] A. Gorban, H. Idrees, Y.-G. Jiang, A. Roshan Zamir, I. Laptev, M. Shah, and R. Sukthankar. THUMOS challenge: Action recognition with a large number of classes. http://www.thumos.info/, 2015.

[14] F. C. Heilbron, V. Escorcia, B. Ghanem, and J. C. Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *CVPR*, 2015.

[15] D. Hoiem, Y. Chodpathumwan, and Q. Dai. Diagnosing error in object detectors. *ECCV*, 2012.

[16] M. Jain, H. Jegou, and P. Bouthemy. Better exploiting motion for better action recognition. In *CVPR*, 2013.

[17] Y. e. a. Jian. Trajectory-based modeling of human actions with motion reference points. In *CVPR*, 2012.

[18] J. Johnson, A. Alahi, and L. Fei-Fei. Perceptual losses for real-time style transfer and super-resolution. In *ECCV*, 2016.

[19] A. Karpathy, G. Toderici, S. Shetty, T. Leung, R. Sukthankar, and L. Fei-Fei. Large-scale video classification with convolutional neural networks. In *CVPR*, 2014.

[20] W. Kay, J. Carreira, K. Simonyan, B. Zhang, C. Hillier, S. Vijayanarasimhan, F. Viola, T. Green, T. Back, P. Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017.

[21] A. Khosla, T. Zhou, T. Malisiewicz, A. A. Efros, and A. Torralba. Undoing the damage of dataset bias. In *ECCV*, 2012.

[22] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. Hmdb: a large video database for human motion recognition. In *ICCV*, 2011.

[23] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *CVPR*, 2008.

[24] M. Mathieu, C. Couprie, and Y. LeCun. Deep multi-scale video prediction beyond mean square error. In *ICLR*, 2015.

[25] M. Mathieu, C. Couprie, and Y. LeCun. Deep multiscale video prediction beyond mean square error. In *ICLR*, 2016.

[26] J. Y.-H. N. Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici. Beyond short snippets: Deep networks for video classification. *arXiv preprint arXiv:1609.08675*, 2015.

[27] M. Raptis and L. Sigal. Poselet key-framing: A model for human activity recognition. In *CVPR*, 2013.

[28] O. Russakovsky, J. Deng, Z. Huang, A. C. Berg, and L. Fei-Fei. Detecting avocados to zucchinis: what have we done, and where are we going? In *ICCV*, 2013.

[29] S. Sadanand and J. J. Corso. Action bank: A highlevel representation of activity in video. In *CVPR*, 2012.

[30] G. A. Sigurdsson, O. Russakovsky, and A. Gupta. What actions are needed for understanding human actions in videos? In *ICCV*, 2017.

[31] K. Simonyan and A. Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*, 2014.

[32] K. Soomro, A. Roshan Zamir, and M. Shah. UCF101: A dataset of 101 human actions classes from videos in the wild. In *CRCV-TR-12-01*, 2012.

[33] C. Sun and R. Nevatia. Active: Activity concept transitions in video event classification. In *ICCV*, 2013.

[34] K. Tang, L. Fei-Fei, and D. Koller. Learning latent temporal structure for complex event detection. In *CVPR*, 2012.

[35] A. Torralba and A. A. Efros. Unbiased look at dataset bias. In *CVPR*, 2011.

[36] D. Tran, L. Bourdev, R. Fergus, L. Torresani, and M. Paluri. Learning spatiotemporal features with 3d convolutional networks. In *ICCV*, 2015.

[37] D. Tran, J. Ray, Z. Shou, S.-F. Chang, and M. Paluri. Convnet architecture search for spatiotemporal feature learning. *arXiv preprint arXiv:1708.05038*, 2017.

[38] J. van Amersfoort, A. Kannan, M. Ranzato, A. Szlam, D. Tran, and S. Chintala. Transformation-based models of video sequences. *arXiv preprint arXiv:1701.08435*, 2017.

[39] G. Varol, I. Laptev, and C. Schmid. Long-term temporal convolutions for action recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2017.

[40] C. Vondrick, H. Pirsiavash, and A. Torralba. Generating videos with scene dynamics. In *NIPS*, 2016.

[41] H. Wang and C. Schmid. Action recognition with improved trajectories. In *ICCV*, 2013.

[42] L. Wang, Y. Qiao, and X. Tang. Action recognition with trajectory-pooled deep-convolutional descriptors. In *CVPR*, 2015.

[43] L. Wang, Y. Xiong, Z. Wang, Y. Qiao, D. Lin, X. Tang, and L. Van Gool. Temporal segment networks: Towards good practices for deep action recognition. In *ECCV*, 2016.

[44] P. Wang, Y. Cao, C. Shen, L. Liu, and H. Shen. Temporal pyramid pooling based convolutional neural networks for action recognition. *arXiv preprint arXiv:1503.01224*, 2015.

[45] X. Wang, A. Farhadi, and A. Gupta. Actions˜ transformations. In *CVPR*, 2016.

[46] D. Wei, B. Zhou, A. Torrabla, and W. Freeman. Understanding intra-class knowledge inside cnn. *arXiv preprint arXiv:1507.02379*, 2015.

[47] Z. Wu, X. Wang, Y.-G. Jiang, H. Ye, and X. Xue. Modeling spatial-temporal clues in a hybrid deep learning framework for video classification. In *Proceedings of the 23rd ACM international conference on Multimedia*, pages 461–470. ACM, 2015.

[48] Z. Xu, Y. Yang, and A. G. Hauptmann. A discriminative cnn video representation for event detection. *arXiv preprint arXiv:1411.4006v1*, 2015.

[49] L. Yao, A. Torabi, K. Cho, N. Ballas, C. Pal, H. Larochelle, and A. Courville. Describing videos by exploiting temporal structure. In *ICCV*, 2015.

[50] J. Yue-Hei Ng, M. Hausknecht, S. Vijayanarasimhan, O. Vinyals, R. Monga, and G. Toderici. Beyond short snippets: Deep networks for video classification. In *CVPR*, 2015.

[51] M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *ECCV*, 2014.

[52] S. Zha, F. Luisier, W. Andrews, N. Srivastava, and R. Salakhutdinov. Exploiting image-trained cnn architectures for unconstrained video classification. *arXiv preprint arXiv:1411.4006v1*, 2015.

[53] B. Zhou, A. Khosla, A. Lapedriza, A. Olivia, and A. Torralba. Object detectors emerge in deep scene cnns. In *ICLR*, 2015.

[54] J.-Y. Zhu, T. Park, P. Isola, and A. A. Efros. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *ICCV*, 2017.

[55] W. Zhu, J. Hu, G. Sun, X. Cao, and Y. Qiao. A key volume mining deep framework for action recognition. In *CVPR*, 2016.