Interpretability of Video Classifiers

Mayuka Jayawardhana [*1] **Sakshi Kakde** [*1] **Vishnu Sashank Dorbala** [*1]

Abstract

As deep learning models become more commonplace, there is a rising need to accurately interpret them. One way to do this involves the usage of different types of saliency methods. While saliency has been well explored on single sample data in vision and language, the applicability of these methods in analyzing sequential or time-series data remains relatively unexplored. In our work, we look at tackling this problem of interpretability of sequential data, employing saliency approaches. In particular, we focus on interpreting real world videos, which poses its own set of challenges. Our results show a heavy computation load, demonstrating the need for more efficient algorithms for computing sequential saliency.

1. Introduction

Several advances in computational hardware over the last decade have impacted the accessibility of Machine Learning in several areas, including finance (Rundo et al., 2019; Vachhani et al., 2019), medicine (Vayena et al., 2018; Rajkomar et al., 2019) and IoT (Deng et al., 2020). As such, there is a need to interpret or explain the outcome of these models (Lipton, 2018; Gilpin et al., 2018). This becomes even more critical when Machine Learning gets applied to areas such as healthcare (Caruana et al., 2015) and transport (Das et al., 2020).

Evaluating the interpretability of a model is particularly challenging, due to the absence of quantitative ground truth data (Hooker et al., 2018; Ghorbani et al., 2019). Even assuming that informative features can be defined a priori, there exists uncertainty in whether choosing from these features was what influenced prediction performance (Adebayo et al., 2018). In case of sequential data, this is an even harder task, as qualitative human annotated ground truth cannot be well defined.

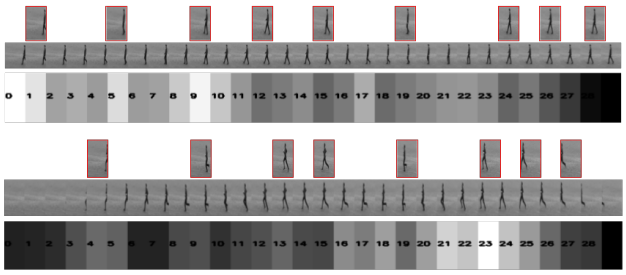

Figure 1. Relative time saliency contribution results on walking and running sequences of the KTH dataset. Observe that the most salient frames (lighter color) for the most part correspond to the parts containing the activity. The red frames above are the keyframes detected (threshold of 30%).

(Ismail et al., 2020) in their work, tackle the problem of saliency on synthetic time-series data. They benchmark results on various architectures, and observe that the standard architectures fail to capture temporal saliency. They also propose a novel rescaling based approach to tackle this problem.

A key drawback with (Ismail et al., 2020) is that they use synthetic time-series data, which has a well defined ground truth. This is not the case when it comes to real world temporal data, where saliency ground truth are hard to, if not impossible to define.

In an attempt to tackle this issue, in our work, we explore saliency on real world temporal data. Following a similar approach as presented in (Ismail et al., 2020), we first look at the performance of saliency models on standard architectures, and then use rescaling to improve saliency. We perform a thorough analysis on an action recognition dataset, and draw conclusions on the impact of their work on real world data.

The key contributions of our work are as follows:

- We study the interpretability of sequence models utilizing an attention based saliency scheme on real world videos.
- We obtain results on the KTH action recognition dataset, focusing on analyzing two classes (walking

[*]Equal contribution [1]University of Maryland, College Park. Correspondence to: Mayuka Jayawardhana <mayukaj@umd.edu>, Sakshi Kakde <sakshi@umd.edu>, Vishnu Sashank Dorbala <vdorbala@umd.edu>.

Proceedings of the 37^{th} International Conference on Machine Learning, Online, PMLR 119, 2020.

and running).

- Specifically, we look at obtaining feature as well as temporal saliency on this data using a gradient based approach, and study the time contribution of each frame.
- Finally, we present our inference on the obtained results, comparing the outcomes with a traditional keyframe detection method.

2. Related Work

Saliency methods are used to understand the importance of input features for a deep learning model in making predictions (Simonyan et al., 2014). Recent work have introduced architectures that can create highly accurate saliency maps (Shrikumar et al., 2017).

Ismail et al. 2020 have benchmarked the effectiveness of these methods on time series models. They have demonstrated that general, network architectures and saliency methods fail to reliably and accurately identify feature importance over time, and have developed a temporal saliency rescaling (TSR) approach to better capture important features for each time step.

Ismail et al. 2019 have observed saliency signals vanish over time for RNNs which makes it difficult to reliably detect important features in a arbitrary time interval. They have mathematically proved that the information preserved depends on the ”forget gate” for a LSTM, thus resulting in this decay over time. To address this problem the authors have proposed a novel structure called ”input-cell attention” we intend to study this attention (Vaswani et al., 2017) based mechanism and provide a mathematical formulation for why it helps in the vanishing saliency problem.

More recently, saliency methods that look into finer, more precise details of the gradient (of the loss) have been developed.

Singla et al. 2019 observes that most saliency methods use first order approximation of the loss. They look into the effects of using higher order approximations for generating saliency maps. To this end they have characterized a closed-form formula for the Hessian matrix of a deep ReLU network. Srinivas & Fleuret 2019 look at including both input sensitivity and per-neuron sensitivity components in the final (full) gradient calculation for generating saliency maps.

In order to study the saliency, we hypothesize that it must look similar to the optical flow. While the optical flow accurately detects motion in the direction of intensity gradient, the temporal saliency is not perfectly equal to the amplitude of all the motion between each adjacent frame pair (Zhong et al., 2013). We however still compute the flow, as it gives us a standard baseline for comparison.

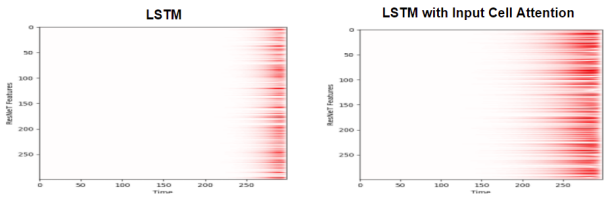

Figure 2. Comparison of the effect of vanishing saliency between video classifiers based on LSTM and LSTM with input cell attention. Observe the improved performance on earlier timesteps when input cell attention is used.

3. Problem Definition and Setup

We study the problem of identifying both visually and temporally salient features on real world videos. To this end we implement and train a LSTM based action classification network and generate temporally robust saliency maps.

We first directly generate saliency maps using the input gradient method (Simonyan et al., 2014) and observe that only features present in the last several input frames are assigned high saliency scores. This is known as the vanishing saliency problem and is observed in specific deep sequence models. Ismail et al. 2019 provide theoretical evidence for this effect in LSTM networks and propose a novel RNN cell structure known as input cell attention which mitigates it. We implement input cell attention for our video classifier and observe a notable improvement in the saliency map representations of the earlier time steps. Our results are shown in figure 2.

Ismail et al. 2020 experimentally show that in general, network architectures and saliency methods fail to reliably and accurately identify feature importance over time in time series data and propose an approach named two step Temporal Saliency Rescaling (TSR), shown in figure 3. We adopt this approach in our model to generate robust saliency maps that capture temporally salient features. Rescaling saliency maps using the two step method is extremely computationally intensive for video inputs as it uses a processes which calculates saliency scores by iterative masking of each individual feature (pixel) in each frame. Therefore we additionally experiment with one-step re-scaling which is a simplified version of TSR that only performs frame-wise masking.

We formulate our approach as a binary classification problem and choose the two classes ‘walking’ and ‘running’ from the KTH action recognition Dataset (Schuldt et al., 2004). All classes in the KTH dataset have the same individuals performing different actions in a similar background. We argue that this leads to the classifier relying more heavily on causal features to make accurate decisions. In contrast

Algorithm 1: Temporal Saliency Rescaling (TSR)

Given: input X, a baseline interpretation method $R(.)$
Output: TSR interpretation method $R^{TSR}(.)$
for $t \leftarrow 0$ **to** T **do**
  Mask all features at time t: $\overline{X}_{:,t} = 0$, otherwise $\overline{X} = X$;
  Compute Time-Relevance Score $\Delta_t^{time} = \sum_{i,t} |R_{i,t}(X) - R_{i,t}(\overline{X})|$;
for $t \leftarrow 0$ **to** T **do**
  for $i \leftarrow 0$ **to** N **do**
    if $\Delta_t^{time} > \alpha$ **then**
      Mask feature i at time t: $\overline{X}_{i,:} = 0$, otherwise $\overline{X} = X$;
      Compute Feature-Relevance Score $\Delta_i^{feature} = \sum_{i,t} |R_{i,t}(X) - R_{i,t}(\overline{X})|$;
    else
      Feature-Relevance Score $\Delta_i^{feature} = 0$;
    Compute (time,feature) importance score $R_{i,t}^{TSR} = \Delta_i^{feature} \times \Delta_t^{time}$;

Figure 3. Two-Step Temporal Saliency Rescaling (TSR) algorithm introduced by Ismail et al. 2020.

choosing two classes such as 'playing badminton' vs 'playing the piano' could result in the classifier relying on spurious features (background). 'Walking and 'running' in particular have high visual similarity but have different temporal characteristics. Combining these arguments we hypothesize that by choosing these two specific classes the saliency maps will closely resemble the optical flow. As optical flow can be easily calculated this provides a good baseline for the qualitative evaluation of the generated saliency maps.

4. Experimentation and Results

4.1. Temporal Saliency

This section shows plots for various walking and running sequences. Due to space constraints, we are presenting only three sequence plots from each class.

In our experimentation, we first obtain a sequence of thirty frames extracted from each video. The dataset is such that these 30 frames contain enough information about walking/running for us to perform inference.

We observe that the vanishing saliency problem 2 does not occur in our setup. We believe this to be the case due to the small size of the data sequence chosen.

To obtain the time contribution on each extracted sequence, we perform saliency masking, similar to (Ismail et al., 2020). This is a common approach to evaluate interpretability, where a part of the data is masked before prediction. This gives us an idea as to whether the model was utilizing features from the masked region for prediction or not.

In our case, we mask each frame individually, and compute prediction accuracy. The resultant accuracies for each mask are used to measure the time contribution of the sequence. These results are presented in figures [4-9].

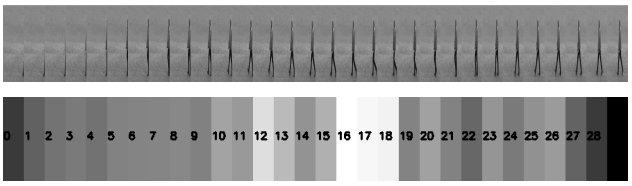

Figure 4. Frames and time contribution for person01_walking_d3 from KTH dataset

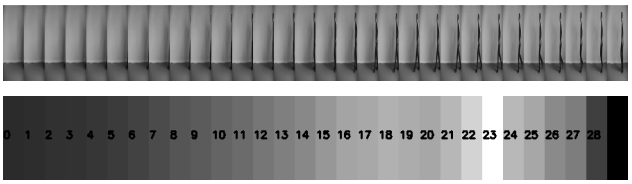

Figure 5. Frames and time contribution for person6_walking_d4 from KTH dataset

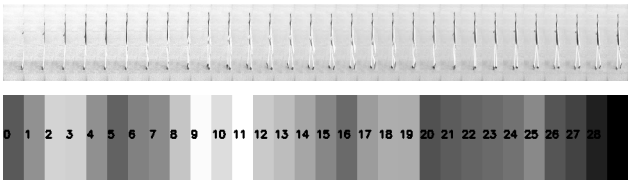

Figure 6. Frames and time contribution for person11_walking_d3 from the KTH dataset

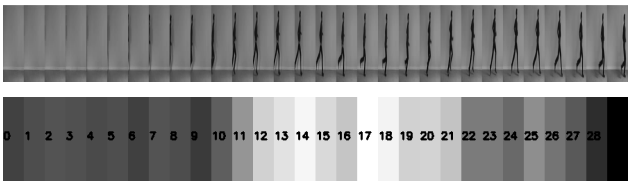

Figure 7. Frames and time contribution for person01_running_d4 from KTH dataset

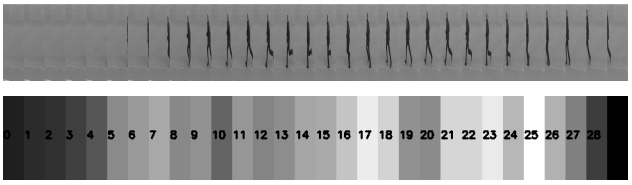

Figure 8. Frames and time contribution for person6_running_d3 from KTH dataset

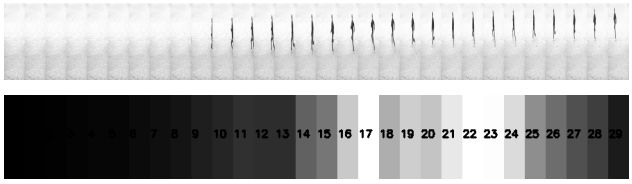

Figure 9. Frames and time contribution for person17_running_d2 from KTH dataset

4.2. Feature Saliency

4.2.1. One Step Method (Input Grad)

We use the Captum tool provided by Pytorch which has various saliency methods implemented. As mentioned earlier in the setup, we look at generating saliency using (Simonyan et al., 2014), which uses an input gradient based method.

The output saliency is then scaled using the time contribution computed in section 4.1. We also obtain the optical flow using the farneback method (Farnebäck, 2003) in OpenCV, and use this for comparison. Refer figures [10-15].

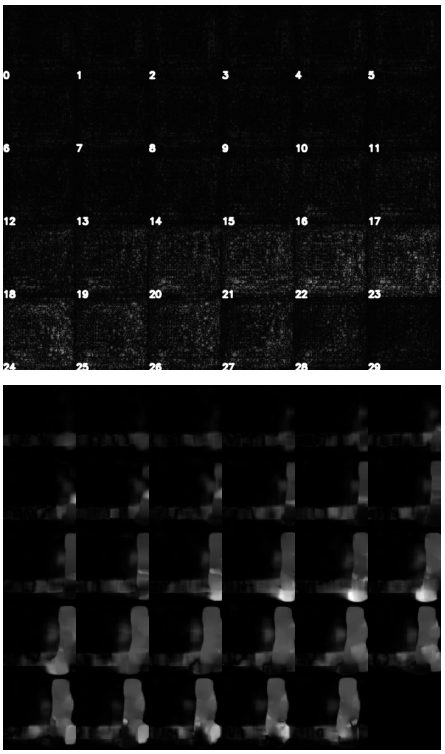

Figure 11. Feature saliency and optical flow for person06_walking_d4 from KTH dataset

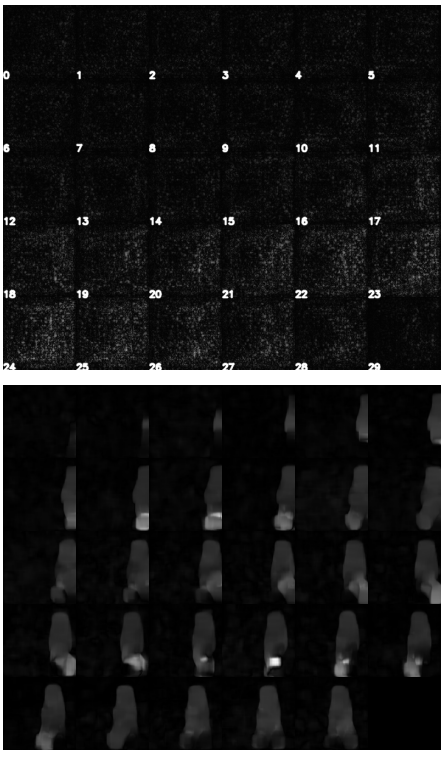

Figure 10. Feature saliency and optical flow for person01_walking_d3 KTH dataset

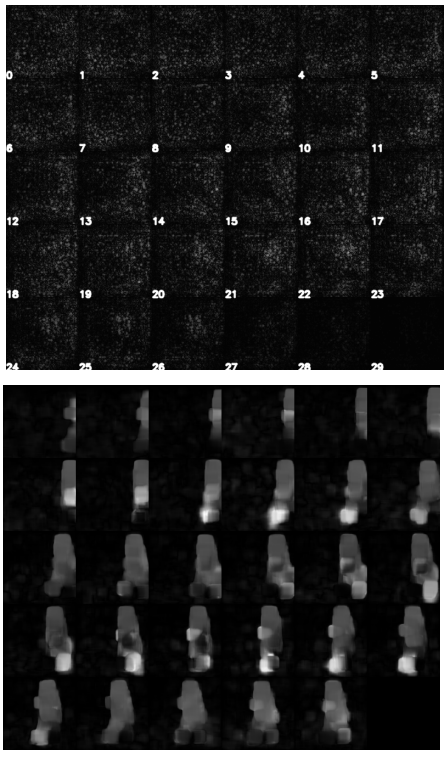

Figure 12. Feature saliency and optical flow for person11_walking_d3 from KTH dataset

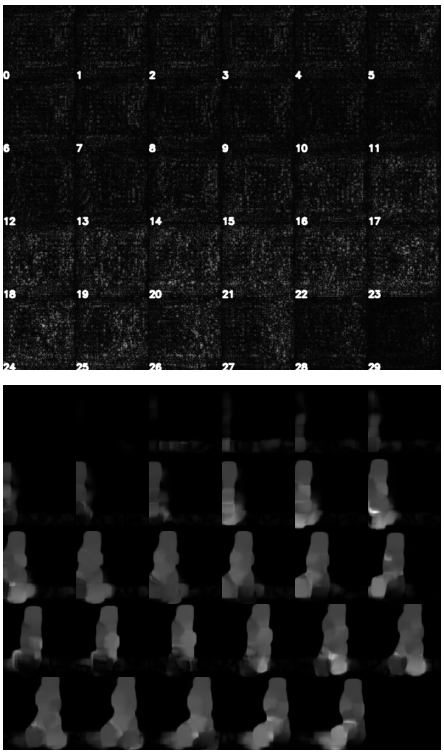

Figure 13. Feature saliency and optical flow for person01_running_d4 from KTH dataset

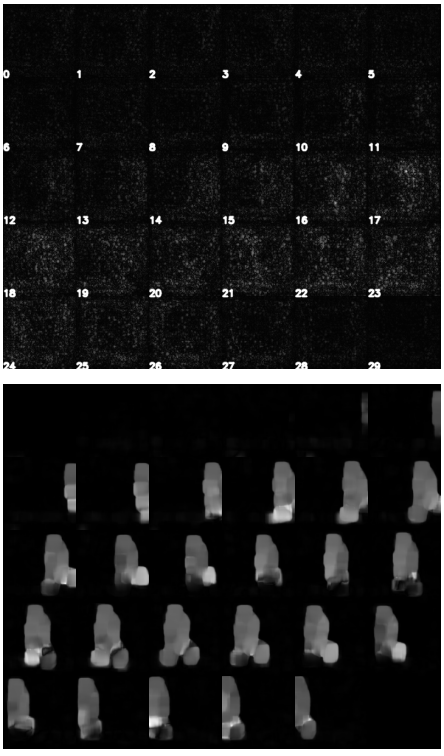

Figure 14. Feature saliency and optical flow for person06_running_d3 from KTH dataset

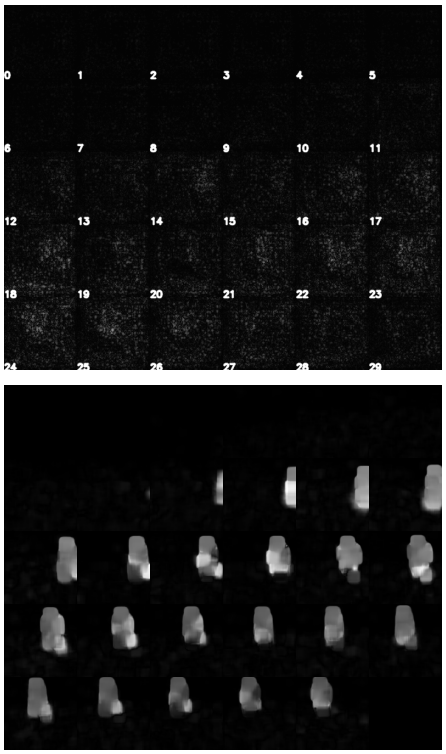

Figure 15. Feature saliency and optical flow for person17_running_d2 from KTH dataset

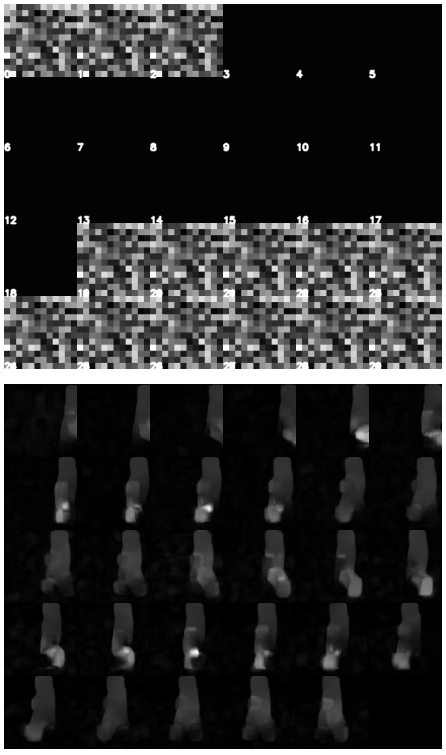

Figure 16. TRS Feature saliency and optical flow for person01_d1 from KTH Dataset

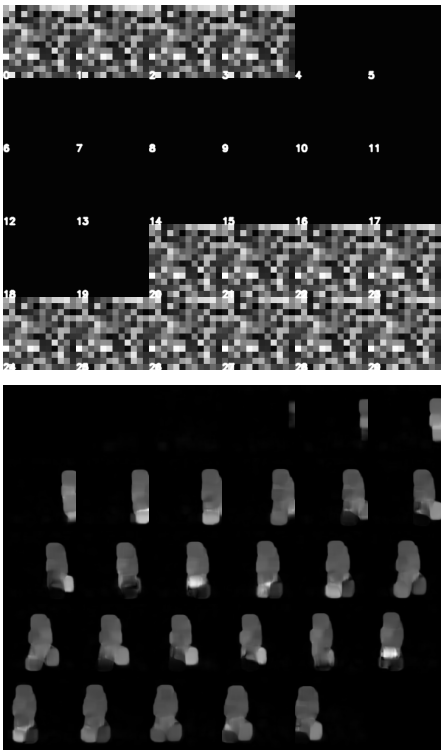

Figure 17. TSR Feature saliency and optical flow for person07_running_d3 from KTH dataset

4.2.2. Two Step Method (Temporal Saliency Rescaling)

We perform two step rescaling, in order to figure out which pixels in each frame are relevant to the time sequence.Here, we mask out each pixel and compute the saliency on the entire sequence. As a result, it is extremely computationally heavy.

To reduce the cost of computations, during feature masking, we masked a block of size 10×10, instead of a single pixel. Figure 16 and 17 show the results of this approach, and we can observe that they are very pixelated, and cannot be inferred.

4.3. Inference with Optical Flow

From a visual analysis, we observe that the One Step method gives us far more interpretable results as opposed to the two step method. Playing the obtained saliency maps using the One Step method in sequence shows some resemblance of the action, but it is not very clear. The two step method on the other hand, cannot be inferred at all due to computational limitations.

5. Conclusion

The obtained saliency maps are not as expected. Upon closer inspection, we do notice a correlation in the temporal contribution of different frames with manually chosen important timesteps. A comparison of the time contribution with respect to the important key frames obtained further validates coherence in temporal saliency.

The feature saliency maps however, are very hard to decipher. On careful observation, we can see an overlap between the computed optical flow and the obtained saliency maps from the one step method. We could not draw any conclusions on the two step method due to computational limitations. As such, we believe that both these approaches are insufficient to evaluate saliency on real world temporal sequence data.

6. Discussion

6.1. Violation of IID assumption

The TSR method used in section 3 masks the timeframe/feature to get the change in saliency. However, the obtained change in saliency can be due to the masking of important features or due to violation on the IID assumption. We think that though the method is able to highlight the important timestamps in the proposed setup, it may not be robust and the results may not be generalized for other video classification tasks.

6.2. Benchmarking

We do not have a quantitative measure for the accuracy of the saliency maps. One way to check is to mask the important time frames obtained in section 4.1, however, that would violate the IID assumption. (Ismail et al., 2020) discusses about benchmarking using the synthetic dataset where the IID assumption can be satisfied as the data generation process is known. The problem to quantify the accuracy of saliency maps on real world data is a relatively new and challenging problem.

6.3. Computational Challenges

Computing saliency maps using the two step TSR method is extremely computationally intensive for video data as it needs to iteratively mask each individual pixel (in each frame) to compute the saliency scores. Even by using windows of size 10×10 it takes approximately 1 hour to generate saliency maps for 1 video on a NVIDIA 3080 GPU, with 32GB of RAM.

6.4. Comparison with Key Frame Detection

We cross validated our temporal saliency scores with a key frame detector as a sanity check. We observed that the frames chosen by the key frame detection algorithm had high temporal saliency scores. We believe this is a good result, as the obtained saliency corresponds well with the key frames.

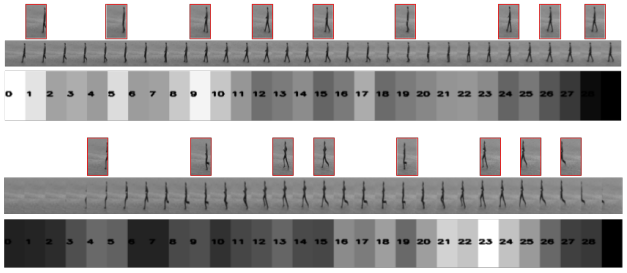

Figure 18. Time contribution along with detected key frames on the sequences.

6.5. Future Work

Interpretability of time series data is an active field of research. Future directions to our own attempt at this task would include testing on multi-class data and developing more efficient algorithms to calculate temporal saliency scores.

A simple extension on that remains to be explored increasing the window size of the masking scheme (masking multiple sets of frames). We hypothesize that saliency in time series data depends not just on one frame, but rather a set of frames.

Another direction would be to improve the efficiency of the two step approach. One way to approach this might be to use image features (superpixels or deep features) as opposed to pixels to compute TSR.

References

Adebayo, J., Gilmer, J., Muelly, M., Goodfellow, I., Hardt, M., and Kim, B. Sanity checks for saliency maps. *arXiv preprint arXiv:1810.03292*, 2018.

Caruana, R., Lou, Y., Gehrke, J., Koch, P., Sturm, M., and Elhadad, N. Intelligible models for healthcare: Predicting pneumonia risk and hospital 30-day readmission. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pp. 1721–1730, 2015.

Das, S., Dutta, A., Dey, K., Jalayer, M., and Mudgal, A. Vehicle involvements in hydroplaning crashes: Applying interpretable machine learning. *Transportation research interdisciplinary perspectives*, 6:100176, 2020.

Deng, S., Zhao, H., Fang, W., Yin, J., Dustdar, S., and Zomaya, A. Y. Edge intelligence: The confluence of edge computing and artificial intelligence. *IEEE Internet of Things Journal*, 7(8):7457–7469, 2020.

Farnebäck, G. Two-frame motion estimation based on polynomial expansion. In *Scandinavian conference on Image analysis*, pp. 363–370. Springer, 2003.

Ghorbani, A., Abid, A., and Zou, J. Interpretation of neural networks is fragile. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pp. 3681–3688, 2019.

Gilpin, L. H., Bau, D., Yuan, B. Z., Bajwa, A., Specter, M., and Kagal, L. Explaining explanations: An overview of interpretability of machine learning. In *2018 IEEE 5th International Conference on data science and advanced analytics (DSAA)*, pp. 80–89. IEEE, 2018.

Hooker, S., Erhan, D., Kindermans, P.-J., and Kim, B. A benchmark for interpretability methods in deep neural networks. *arXiv preprint arXiv:1806.10758*, 2018.

Ismail, A. A., Gunady, M., Pessoa, L., Bravo, H. C., and Feizi, S. Input-cell attention reduces vanishing saliency of recurrent neural networks, 2019.

Ismail, A. A., Gunady, M., Corrada Bravo, H., and Feizi, S. Benchmarking deep learning interpretability in time series predictions. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., and Lin, H. (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 6441–6452. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/47a3893cc405396a5c30d91320572d6d-Paper.pdf.

Lipton, Z. C. The mythos of model interpretability: In machine learning, the concept of interpretability is both important and slippery. *Queue*, 16(3):31–57, 2018.

Rajkomar, A., Dean, J., and Kohane, I. Machine learning in medicine. *New England Journal of Medicine*, 380(14): 1347–1358, 2019.

Rundo, F., Trenta, F., di Stallo, A. L., and Battiato, S. Machine learning for quantitative finance applications: A survey. *Applied Sciences*, 9(24):5574, 2019.

Schuldt, C., Laptev, I., and Caputo, B. Recognizing human actions: a local svm approach. In *Proceedings of the 17th International Conference on Pattern Recognition, 2004. ICPR 2004.*, volume 3, pp. 32–36 Vol.3, 2004. doi: 10.1109/ICPR.2004.1334462.

Shrikumar, A., Greenside, P., and Kundaje, A. Learning important features through propagating activation differences. In Precup, D. and Teh, Y. W. (eds.), *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pp. 3145–3153. PMLR, 06–11 Aug 2017. URL https://proceedings.mlr.press/v70/shrikumar17a.html.

Simonyan, K., Vedaldi, A., and Zisserman, A. Deep inside convolutional networks: Visualising image classification models and saliency maps, 2014.

Singla, S., Wallace, E., Feng, S., and Feizi, S. Understanding impacts of high-order loss approximations and features in deep learning interpretation. In Chaudhuri, K. and Salakhutdinov, R. (eds.), *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 5848–5856. PMLR, 09–15 Jun 2019. URL https://proceedings.mlr.press/v97/singla19a.html.

Srinivas, S. and Fleuret, F. Full-gradient representation for neural network visualization. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/80537a945c7aaa788ccfcdf1b99b5d8f-Paper.pdf.

Vachhani, H., Obiadat, M. S., Thakkar, A., Shah, V., Sojitra, R., Bhatia, J., and Tanwar, S. Machine learning based stock market analysis: A short survey. In *International Conference on Innovative Data Communication Technologies and Application*, pp. 12–26. Springer, 2019.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. Attention is all you need, 2017.

Vayena, E., Blasimme, A., and Cohen, I. G. Machine learning in medicine: addressing ethical challenges. *PLoS medicine*, 15(11):e1002689, 2018.

Zhong, S.-h., Liu, Y., Ren, F., Zhang, J., and Ren, T. Video saliency detection via dynamic consistent spatio-temporal attention modelling. In *Twenty-seventh AAAI Conference on Artificial Intelligence*, 2013.