The 1st Place Solution for ROSE Challenge 2022

Jun Yu[1], Zhihong Wei[1] , Mohan Jing[1], Zepeng Liu[1], Xiaohua Qi[1],
Keda Lu[1], Liwen Zhang[1], Hao Chang[1], Hang Zhou[2]
[1]University of Science and Technology of China, [2]PAII Inc
{harryjun}@ustc.edu.cn, {weizh588,pppeng97, zlw1113, changhaoustc}@mail.ustc.edu.cn,
jingmohan1@gmail.com, xiaohua000109@163.com, wujiekd666@gmail.com, zhouhang395@paii-labs.com

Abstract

Although current computer vision systems can perform well on standard datasets, existing methods are far less robust than human vision systems. Videos in the real world may encounter many corruptions, such as camera motion and compression error. Compared with images, videos contain richer temporal information, which play an important role in video understanding tasks. Therefore it is of great significance to explore the robustness of the models on video understanding. In Robustness in Sequential Data Challenge 2022, we find that TimeSformer has the best robustness on video recognition, both in terms of spatial and temporal corruptions. We also find that the robustness of TimeSformer has improved when the sampling interval increases. For spatial corruption, we simulate 20 types of corruption methods to expand the training data. For temporal corruption, we propose a Multi-Scale Sampling strategy to mitigate the impact of temporal corruption by blending information from multiple sampling scales. In addition, we also try the TRADES++ loss function to make some trade-offs in generalizability and robustness. Finally, our method achieves 98.71%, 99.83% and 89.83% accuracy on HMDB-51P, UCF-101P and Kinetics-400P, respectively, and achieves the champion of CVPR2022 Robustness in Sequential Data challenge.

1. Introduction

In the last decade, advances in deep neural networks and large-scale datasets have led to rapid advances in image and video understanding. However, although current computer vision systems can perform well on test sets, i.e., with good generalization, existing computer vision systems are far less robust than human vision systems [1, 18]. Models in the real world may encounter common corruptions of input data, such as camera motion and systematic errors [27]. Humans are usually not confused by these minor disturbances, such as Gaussian noise, weather variations, etc. While there has been partial work to improve the robustness of models based on spatial domain visual content under such disruptions, temporal domain information has been largely ignored. Compared to images, videos contain richer temporal information, which can play an important role in video understanding tasks. And, studying robustness in sequential data can be a good contribution to the development of action recognition, autonomous driving, and other fields. Data augmentation is a very popular method to increase the generalization of machine learning models by generating additional data. It has been applied in many fields, such as machine translation [6], semi-supervised learning [26], etc. Early data augmentation strategies were designed purely by hand until Autoaugment [3], Randaugment [4], TrivialAugment [17] and other automatic data augmentations were proposed to reduce the difficulty of data augmentation design. Inspired by the above data augmentation,we simulated 20 types of corruptions such as video perturbation and compression using the imgaug [9] library for the dataset characteristics of Rose2022.

Similar to the automatic data augmentation described above, the augmentation method is FREE in terms of cost during the training process and does not require the addition of manual design. Unlike traditional data augmentation, this method focuses on the spatial domain perspective and transforms the input space with a large number of corrupions to improve the overall robustness of the model.

Robustness and accuracy are negatively correlated [16, 23, 29]. The generalization performance decreases as the robustness increases. Therefore, how to balance the robustness and accuracy of generalization has been the focus of academic research. In order not to pursue robustness excessively and lose accuracy, we propose to replace the traditional cross-entropy loss function with Trades++, a loss function that balances the robustness and accuracy.

In particular, we tested our proposed approach using three classical video datasets (UCF-101 [20], HMDB-51 [12] and Kinetics-400 [11]) based on an action recognition task. The test sets were provided by a challenge orga-

nized by the CVPR 2022 workshop Robustness in Sequential Data. First, we evaluated the CNN and Transformer-based models, respectively, and selected TimeSformer as our benchmark model. Second, we expanded the data by simulating nearly 20 types of data perturbations and compressions, which improved the robustness of the model to the spatial domain. In the temporal domain, we mitigate the effects of temporal disruptions on the model by expanding the sampling interval of frames and the Multi-Scale Sampling strategy.The framework of our approach is shown in Fig. 1

The contributions of this paper are as follows.

(1).Comparing the characteristics of CNN and Transformer backbone, we argue that the TimeSformer [2] model based on Transformer's backbone is more robust to temporal data, and evaluate the models of CNN and Transformer through experiments.

(2).In terms of spatial domain robustness, we improve the robustness of the model to the spatial domain by simulating nearly 20 types of data perturbations and compression to expand the data.

(3).In terms of temporal domain robustness, inspired by the "multi-scale" approach for object detection tasks, we innovatively propose a Multi-Scale Sampling strategy to learn different "temporal scale" features by setting different "clip len" and "frame interval" to train the model, which makes the model more robust to temporal corruptions.

(4).With the above mentioned main strategies, we achieved the best accuracy in three datasets, HMDB-51P, UCF-101P, and Kinetics-400P.

2. Related Work

With the booming development of deep learning, researchers have started to apply it to the field of video understanding. DeepVideo [10] firstly proposed to extract features using 2D CNN, and also studied several temporal connectivity patterns to learn spatial and temporal features in videos. [19] proposed two-stream network, which includes a spatial stream and a temporal stream. By adding additional temporal streams, the CNN-based models achieved similar performance of IDT [25] for the first time. However, optical stream costs great computational consumption and is difficult to train and deploy on a large scale. Therefore, [21] proposed C3D to understand video as a 3D tensor with two spatial dimensions and one temporal dimension. In addition to the traditional CNN models, TimeSformer splits the frames in the video into multiple patches and completes the recognition based on self-attention. Compared with 3D CNN, TimeSformer is 3 times faster and the inference time is only one tenth of it.While video understanding is becoming more accurate, research on model robustness is lacking. This poses some difficulties for practical applications.

Model	Accuracy
R(2+1)D	74.71%
SlowFast	76.68%
TSM	81.51%
TimeSformer	89.94%

Table 1. We take the HMDB-51 dataset as an example to compare the baselines of different models.The results demonstrate the excellent robustness of TimeSformer.

3. Method

3.1. TimeSformer

In order to balance the computational complexity and model accuracy, we use divided space-time attention, which is also the conventional application form of TimeSformer. In the temporal dimension, each patch performs an attention operation with only one patch extracted from the rest of the frame at the corresponding position, instead of performing a self-attention operation with all the patches of the rest of frames. In the spatial dimension, the patch performs attention computation with all other patches extracted from the same frame. The temporal and spatial attention operations are performed sequentially, with the self-attention operation on the temporal sequence being computed first, followed by the spatial self-attention operation.

[27] conducted extensive experiments and confirmed that TimeSfomer has good robustness performance on video data. Transformer [24] itself has better robustness than CNN to some extent, so TimeSformer is usually more robust compared to 2D CNN or 3D CNN-based models.

In fact, TimeSformer is an extension of ViT [5] in the field of video understanding, which inherits the advantages of ViT. CNN can effectively learn low-level features, i.e., image edges, corners, pixels, and other small details of images, but CNN cannot effectively learn global semantic information due to the local perceptual characteristics of convolution. Moreover, self-attention of ViT can dynamically adjust the perceptual domain, so that it can better cope with many types of noise.

Without any additional processing, we successively tried R(2+1)D [22], SlowFast [7], TSM [13] and TimeSformer. The results of HMDB-51P are shown in Tab. 1. The accuracy of TimeSformer reached 89.94%, which is much higher than other models.

3.2. Robustness w.r.t Spatial Corruptions

For the spatial domain Robustness, we simulate common spatial corruption methods to expand the data. The specific corruption methods can be seen in Tab. 2 We set the corruption probability to 50%, which can better preserve the original image features and ensure the generalization of the

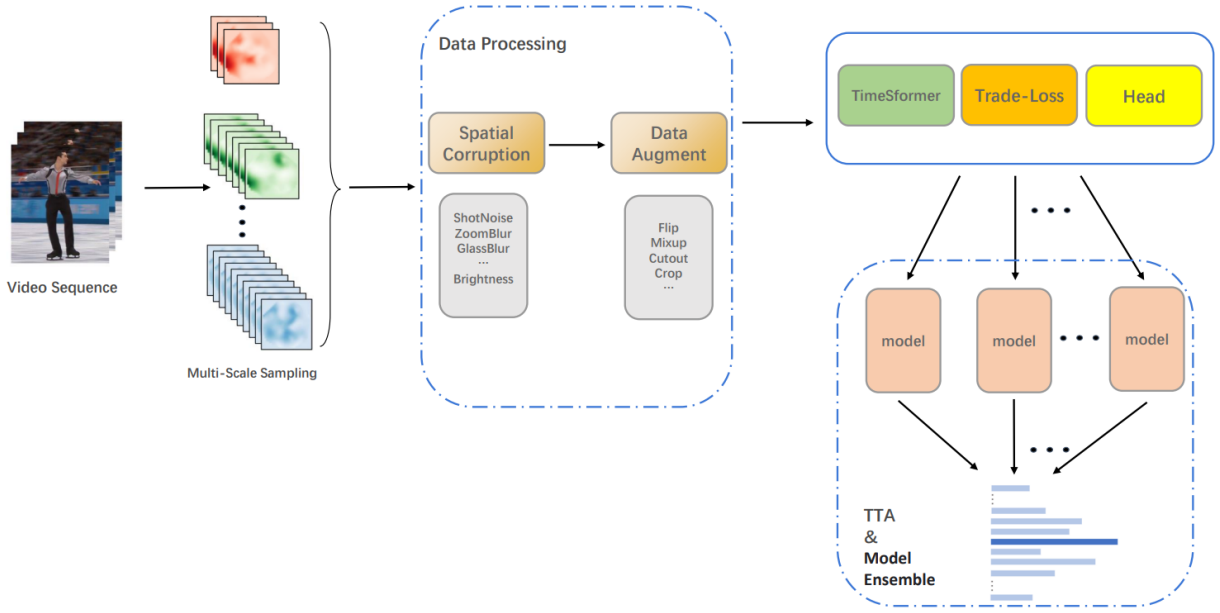

Figure 1. First we perform Multi-Scale Sampling on the video.After that multiple random corruptions are performed on the sampled frames with a probability of 50% and the severity level of corruption is 2. Finally the confidence scores obtained from the inference of the multi-scale models are weighted and ensembled.

model. We tried to make the severity level equal to 2, 3, and a mixture of 2 and 3. Experimentally, the model is most stable when it is 2. By expanding the data by simulating common spatial corruption methods, the robustness and accuracy of the model are greatly improved.

3.3. Robustness w.r.t Temporal Corruptions

Unlike the robust strategy for the spatial domain, we do not simulate corruption in the temporal domain because we believe that corruption in the temporal domain is more random, such as randomly swapping the order of frames, which is difficult for us to simulate realistically, and because temporal corruption may also increase or decrease the length of our video, which makes it difficult to control some of the parameters of sampling.

Therefore, to address the above issues, firstly, we believe that increasing the sampling interval of frames can make the network pay more attention to the global frame sequence and thus be robust to local frame temporal corruptions. Due to the powerful global modeling capability of the TimeSformer model, we find that increasing sampling interval appropriately not only does not affect the modeling of temporal features , but also weakens the impact of temporal corruptions.

Secondly, inspired by the "multi-scale" approach for object detection tasks [8, 14, 15], we propose the Multi-Scale Sampling strategy. The most challenging problem in object detection is the scale variance of the object. In object detection, the objects have different shapes and sizes, and there may even be some very small, very large, or extreme shapes (e.g., elongated, narrow and tall), which makes the accurate identification and precise localization of objects extremely difficult. Inspired by this idea, we believe that by setting different "clip len" and "frame interval", the model can learn different temporal scale features and be more robust to temporal corruption.

3.4. Trade off between Robustness, Generalization

The goal of both of our previously proposed methods is to improve the robustness of the model as much as possible, but may ignore the generalization of the model. Therefore, we look for a simple and effective method to trade-off robustness and generalization, which only modifies the loss function and is almost completely free in the training phase.TRADES [29] is a loss function that trade-offs robustness and generalization for more aggressive attacks, specifically against samples. We consider natural perturbations that are more common in practical application scenarios, including spatial corruptions, temporal corruptions, camera related perturbations, and compression These minor perturbations are likely to cause the model to produce recognition results that differ significantly from those of clean samples or even incorrectly.

Therefore, we borrowed from TRADES and proposed TRADES++. A regularization term is added to the original cross-entropy function, in order to make the clean and corrupted samples consistent in the input and output spaces. Formally, for each batch of clean samples $\{X, Y\}$, define

GaussianNoise	ShotNoise	ImpulseNoise	SpeckleNoise
GlassBlur	DefocusBlur	MotionBlur	ZoomBlur
Brightness	Saturate	JpegCompression	Pixelate
Rot90	ShearX	CropToFixedSize	Fliplr
GaussianBlur	Contrast	Rotate	RandomResizedCrop

Table 2. 20 kinds of video corruption are applied to the training set to enhance the generalization performance of the model.

the computed TRADES++ by

$$\mathcal{J}(f(X), Y) + \mathcal{J}(f(X_c), Y) + \lambda \cdot \mathcal{K}(f(X)f(X_c)) \quad (1)$$

where $\mathcal{J}$ represents the cross-entropy loss function, X_c represents a batch of samples generated from clean sample perturbations, and $\mathcal{K}$ represents a measure of the difference in output space between two samples that are similar in the input space, where we use the Kullback-Leibler divergence. λ is a balancing factor that adjusts to the robustness and generalization required by the scenario.

3.5. Data Augmentation

We adopted mixup [28] and TTA strategies for data augmentation. Mixup is to add the sample-label data of the two videos according to the proportion λ to generate new sample-label data. The proportion λ obeys the β distribution with parameters (α, α). In the experiment, α is 0.2. During the test, three randomly cropped pictures with a size of 224 x 224 are randomly selected from the original frame, and the data expansion is carried out on the test data set. The introduction of the Mixup strategy improved the results by 0.7%, and the TTA strategy improved the results by 0.5%.

4. Experiments

4.1. Datasets

The training set provided by ROSE 2022 is UCF-101, HMDB-51 and Kinetics-400, which are commonly used for action recognition. The test dataset is based on these three datasets, and various kinds of corruptions are applied to obtain UCF-101P, HMDB-51P and Kinectics-400P. To speed up inference and iterate the model quickly, ROSE 2022 provides mini-test sets: UCF-101PMini, HMDB-51PMini, and Kinetics-400PMini, the size of the mini-test set is about one-quarter of the full test set.

4.2. Setup

In this section, we describe the full training and testing process, including hardware configuration, parameter settings, and training strategy. The models were all loaded with publicly available Kinetics-400 pre-training weights for initialization. All experiments were done on a Tesla A100 using the PyTorch framework. All neural networks take a stochastic gradient descent optimizer with a momentum of 0.9. The initial learning rate was set to 0.005 for 100 training epochs, decaying to 0.1 by the 30th and 50th epoch, respectively, and the batch size was set between 4 and 16 depending on the GPU memory and the number of GPUs used for training. For the training data, we first performed a corruption process, applying a corruption to each frame sampled with a 50% probability. After that, the resolution of the input frames was randomly scaled to 256×320, then randomly cropped to 224×224, and finally the images were flipped with 50% probability. We also adopted the mixup data augmentation strategy to enhance the robustness of the model. In the testing phase, in addition to resize, we also used three-Crop on the test frames. Specifically, to speed up the training, we adopted a half-precision training strategy when training Kinetics-400, which nearly doubled the training speed.

Corruption Rate	25%	50%	75%	100%
Top-1	95.20%	97.16%	96.14%	94.67%

Table 3. Video corruption scale verification using HMDB-51 as example. Corrupt 25%, 50%, 75%, and 100% videos respectively and the results show that the accuracy of the model is highest when the ratio is 50%.

4.3. Simulation Spatial Corruptions

In order to enhance the generalization ability and anti-interference ability of the model, we randomly apply 20 kinds of spatial corruptions to the frames with a certain probability. Multiple corruptions are not superimposed, i.e., only one method is randomly selected to act on the sampling frame.

To determine the optimal corruption probability, We conduct experiments on the HMDB-51 dataset. We performe different ratio of corruption to HMDB-51 and check its accuracy on HMDB-51P. Tab. 3 shows that the robustness of the model is the best when the corruption probability is 50%.

After determining the corruption probability, it is also

Severity Level	2	3	random 2 or 3
Top-1	97.16%	96.67%	96.55%

Table 4. Based on the HMDB-51 data, experiments with severity level of 2, 3 and random 2 or 3. The results show that the model reaches the highest accuracy when the severity level is 2.

Frame Interval	Top-1
4	96.77%
6	97.05%
8	97.71%
10	97.73%
12	97.84%

Table 5. Experiments are conducted using the HMDB-51 dataset with sampling intervals of 4, 6, 8, 10 and 12 frames, and the number of frames sampled is always 12. The results show that the larger the sampling interval, the higher the accuracy of the model.

Frame Interval	Clip Len	Top-1
6	16	97.54%
7	14	97.56%
8	12	97.71%
10	10	97.76%
12	9	97.88%

Table 6. The average number of frames of HMDB-51 is 96, so we set multiple sampling frames and sampling intervals to ensure that the product is around 96. The experimental results show that the highest accuracy is achieved when the sampling interval is 12. The confidence scores calculated from the five sampling scales are weighted and summed, and the accuracy of the model ensemble reaches 98.71%.

necessary to determine the severity of corruption. There are 5 kinds of severity levels when performing corruption. Taking Gaussian blur as an example, the five kinds of severity levels are shown in Fig. 2 In order to avoid the deterioration of the model caused by introducing too much noise information, we choose 2 and 3 severity levels to conduct experiments on the HMDB-51 dataset. Tab. 4 shows that the model achieves the highest accuracy when the severity level is 2.

4.4. Enlarge Frame Interval

Intuitively, we believe that because the TimeSformer has a powerful global modelling capability, appropriately increasing the sampling interval not only will not affect the modelling of temporal features, but also weaken the effect of temporal corruption, thus effectively enhancing the robustness of the model. To verify this idea, we use the HMDB-51 dataset for experiments, and successively try to set the sampling interval to 4, 6, 8, 10, 12, and "clip len" as 12. The results are shown in Tab. 5. The results show that the increase in sampling interval does improve the robustness and accuracy of the model.

4.5. Multi-Scale Sampling

Because TimeSformer has excellent long sequence video understanding, we set different "video scales" for the ablation experiment to capture video information at different scales. Different "video scales" mean changing both the number of frames sampled and the sampling interval, so that "Clip Len" × "Frame Interval" varies within a certain range. We use the HMDB-51 dataset for the experiment. In order to try to sample the whole video frame information, we make the "clip len" × "frame interval" as close as possible to the average number of frames of HMDB-51 in order to ensure that the sampling length can cover the whole video as much as possible, neither exceeding the longest length of the video, nor omitting the tail of the video. The sampling results at different scales will be used in the final model ensemble, so that the model can understand the video more comprehensively. Using HMDB-51 dataset as example, the experimental results are shown in Tab. 6. Using the averaging method to ensemble five models at different scales, the confidence scores of the five models are weighted and summed to achieve the complementary multi-scale information and improve the model accuracy. The accuracy reached 98.71% when the experiments were conducted under the same conditions.

4.6. Conclusion

In this work, we adopt various strategies to resist various perturbations and corrupt information. For model selection, we experimentally compare CNN and Transformer. The results show that the Transformer-based backbone is more robust to corrupted video. For spatial corruption, we simulate 20 corruption methods with a 50% probability to be randomly applied on video sampling frames for training, which improves the robustness of the model in the spatial domain. For temporal corruption, we propose a Multi-Scale Sampling strategy to mitigate the impact of temporal chaos on the model by blending multiple sampling scale information. We experimentally find that the robustness of TimeSformer improves to a certain extent when the sampling interval becomes larger. So we set the sampling interval as large as possible to further improve the robustness under the consideration of video frames, GPU memory size, training speed, etc. The mixup and TTA strategies are also adopted to improve the model accuracy. In addition, we also try the Trades++ loss function to make some trade-offs between generalization and robustness to further improve the model performance. Our method achieves 98.71%, 99.83%, and

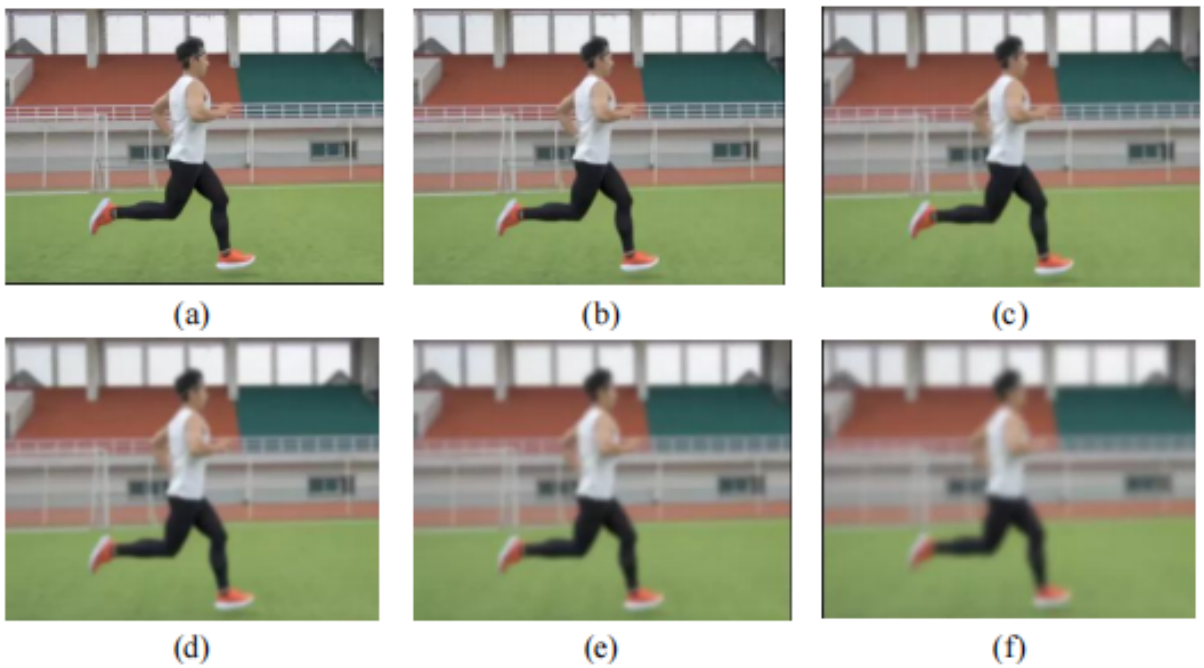

Figure 2. Different severity levels of GaussianBlur.(a) is the original figure, (b)∼(f) are the severity levels of 1∼5 respectively.

89.83% accuracy on HMDB-51P, UCF-101P, and Kinetics-400P, respectively, and wins the champion of the Robustness in Sequential Data challenge in CVPR2022.

References

[1] Aharon Azulay and Yair Weiss. Why do deep convolutional networks generalize so poorly to small image transformations? *arXiv preprint arXiv:1805.12177*, 2018. 1

[2] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding. *arXiv preprint arXiv:2102.05095*, 2(3):4, 2021. 2

[3] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 113–123, 2019. 1

[4] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 702–703, 2020. 1

[5] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 2

[6] Marzieh Fadaee, Arianna Bisazza, and Christof Monz. Data augmentation for low-resource neural machine translation. *arXiv preprint arXiv:1705.00440*, 2017. 1

[7] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. Slowfast networks for video recognition. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 6202–6211, 2019. 2

[8] Sanghoon Hong, Byungseok Roh, Kye-Hyeon Kim, Yeongjae Cheon, and Minje Park. Pvanet: Lightweight deep neural networks for real-time object detection. *arXiv preprint arXiv:1611.08588*, 2016. 3

[9] Alexander B. Jung, Kentaro Wada, Jon Crall, Satoshi Tanaka, Jake Graving, Christoph Reinders, Sarthak Yadav, Joy Banerjee, Gábor Vecsei, Adam Kraft, Zheng Rui, Jirka Borovec, Christian Vallentin, Semen Zhydenko, Kilian Pfeiffer, Ben Cook, Ismael Fernández, François-Michel De Rainville, Chi-Hung Weng, Abner Ayala-Acevedo, Raphael Meudec, Matias Laporte, et al. imgaug. https://github.com/aleju/imgaug, 2020. Online; accessed 01-Feb-2020. 1

[10] Andrej Karpathy, George Toderici, Sanketh Shetty, Thomas Leung, Rahul Sukthankar, and Li Fei-Fei. Large-scale video classification with convolutional neural networks. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 1725–1732, 2014. 2

[11] Will Kay, Joao Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, et al. The kinetics human action video dataset. *arXiv preprint arXiv:1705.06950*, 2017. 1

[12] Hildegard Kuehne, Hueihan Jhuang, Estíbaliz Garrote, Tomaso Poggio, and Thomas Serre. Hmdb: a large video database for human motion recognition. In *2011 Inter-*

national conference on computer vision, pages 2556–2563. IEEE, 2011. 1

[13] Ji Lin, Chuang Gan, and Song Han. Tsm: Temporal shift module for efficient video understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7083–7093, 2019. 2

[14] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2117–2125, 2017. 3

[15] Songtao Liu, Di Huang, et al. Receptive field block net for accurate and fast object detection. In *Proceedings of the European conference on computer vision (ECCV)*, pages 385–400, 2018. 3

[16] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017. 1

[17] Samuel G Müller and Frank Hutter. Trivialaugment: Tuning-free yet state-of-the-art data augmentation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 774–782, 2021. 1

[18] Benjamin Recht, Rebecca Roelofs, Ludwig Schmidt, and Vaishaal Shankar. Do cifar-10 classifiers generalize to cifar-10? *arXiv preprint arXiv:1806.00451*, 2018. 1

[19] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. *Advances in neural information processing systems*, 27, 2014. 2

[20] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *arXiv preprint arXiv:1212.0402*, 2012. 1

[21] Du Tran, Lubomir Bourdev, Rob Fergus, Lorenzo Torresani, and Manohar Paluri. Learning spatiotemporal features with 3d convolutional networks. In *Proceedings of the IEEE international conference on computer vision*, pages 4489–4497, 2015. 2

[22] Du Tran, Heng Wang, Lorenzo Torresani, Jamie Ray, Yann LeCun, and Manohar Paluri. A closer look at spatiotemporal convolutions for action recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pages 6450–6459, 2018. 2

[23] Dimitris Tsipras, Shibani Santurkar, Logan Engstrom, Alexander Turner, and Aleksander Madry. Robustness may be at odds with accuracy. *arXiv preprint arXiv:1805.12152*, 2018. 1

[24] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017. 2

[25] Heng Wang and Cordelia Schmid. Action recognition with improved trajectories. In *Proceedings of the IEEE international conference on computer vision*, pages 3551–3558, 2013. 2

[26] Qizhe Xie, Zihang Dai, Eduard Hovy, Thang Luong, and Quoc Le. Unsupervised data augmentation for consistency training. *Advances in Neural Information Processing Systems*, 33:6256–6268, 2020. 1

[27] Chenyu Yi, Siyuan Yang, Haoliang Li, Yap-peng Tan, and Alex Kot. Benchmarking the robustness of spatial-temporal models against corruptions. *arXiv preprint arXiv:2110.06513*, 2021. 1, 2

[28] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017. 4

[29] Hongyang Zhang, Yaodong Yu, Jiantao Jiao, Eric Xing, Laurent El Ghaoui, and Michael Jordan. Theoretically principled trade-off between robustness and accuracy. In *International conference on machine learning*, pages 7472–7482. PMLR, 2019. 1, 3