Benchmarking the Robustness of CNN-based Spatial-Temporal Models (Supplementary material)

Chenyu Yi[1*] **Siyuan Yang**[1,2*] **Haoliang Li**[3] **Alex C. Kot**[1]
[1]School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore
[2]Interdisciplinary Graduate Programme, Nanyang Technological University, Singapore
[3]Department of Electrical Engineering, City University of Hong Kong, China
{yich0003,siyuan005}@e.ntu.edu.sg haoliali@cityu.edu.hk eackot@ntu.edu.sg

1 Creating Benchmark Datasets

Since our work synthesizes corruptions on existing Kinetics400 and Something-Something-V2 datasets which are available online, we provide complete code for generating all types of corruptions on them. The code is available at `https://github.com/Newbeeyoung/Video-Corruption-Robustness`. The parameters for corruptions with different severity levels are shown in corresponding code functions. In this URL, we list the detail of samples extracted from the official Kinetics400 and Something-Something-V2 datasets. It includes the class name of data, the index of data, and the number of frames in each video data. We also provide code for pre-processing existing datasets Kinetcs and Something-Something-V2, including converting video to HDF5 file and creating JSON files for dataloading. With the code, researchers can recreate the same benchmark datasets with following steps:

1. Download Kinetics400 and Something-Something-V2 from the public websites
2. Extract Mini Kinetics and Mini SSV2 based on class name list
3. Preprocess video data for data loading
4. Apply the proposed corruptions on the Mini Kinetics and Mini SSV2 validation datasets.

2 Benchmark Maintenance

We will maintain our code for generating the benchmark in this link: `https://github.com/Newbeeyoung/Video-Corruption-Robustness`. Any enquiry on the benchmark creation and evaluation can be sent to yich0003@ntu.edu.sg. Besides, we will update the benchmark leaderboard in the link for any work beating the state-of-the-art performance on benchmark. The leaderboard will consist of approach, reference, backbone, input length, sampling method and the approach performance, including clean accuracy, mPC and rPC. Any result can be submitted to us via pull request in the link.

3 Sample of Mini SSV2-C

We show the visualization samples from Mini SSV2-C in Figure 1. From the samples shown, Mini SSV2 consists of first-person view human action videos, while the Mini Kinetics is constructed by third-person view human action videos. With the variety in Mini Kinetics and Mini SSV2, we can evaluate the corruption robustness of models in video classification comprehensively.

[*]equal contribution

Preprint. Under review.

Figure 1: Our proposed corruption robustness benchmarks consists of 12 types of corruptions with five levels of severity for each video. In the visualization examples of Mini SSV2-C, we use uniform sampling to extract 3 frames from each corrupted video, the sampling interval is 20 frames.

4 List of Mini Kinetics Video Classes

The complete list of 200 classes and their corresponding Parent-Child Group grouping is shown in Table 1. The number of videos for each action class is given by the number in brackets following each class name. The number of child-actions of each action group is shown by the number in brackets following each action group name. The class label in blue color means that it belongs to multiple action groups and has occurred above. The class label in red color means that it doesn't belong to any action group.

Table 1: List of Mini Kinetics Video Classes

Class Labels	**Action Groups**
arranging flowers (583) blowing glass (1145) brush painting (532) carving pumpkin (711) getting a tattoo (737)	arts and crafts (**5**)
high jump (954) parkour (504) pole vault (984)	athletics - jumping (**3**)
catching or throwing frisbee (1060) disc golfing (565) javelin throw (912) throwing axe (816) throwing ball (634)	athletics - throwing + launching (**5**)
pumping gas (544)	auto maintenance (**1**)
catching or throwing softball (842) dunking basketball (1105) golf driving (836) golf putting (1081) hitting baseball (1071) juggling soccer ball (484) passing American football (not in game) (1045) playing kickball (468) playing squash or racquetball (980) playing tennis (1144) playing volleyball (804) shooting basketball (595) shooting goal (soccer) (444)	ball sports (**13**)
applauding (411) bending back (635) drumming fingers (409) finger snapping (825) headbanging (1090) pumping fist (1009) shaking head (885) swinging legs (409)	body motions (**8**)
cleaning floor (874) cleaning pool (447) cleaning shoes (706) cleaning toilet (576) cleaning windows (695) making bed (679) setting table (478)	cleaning (**7**)
folding clothes (695) folding napkins (874) ironing (535) making bed (679) tying bow tie (387)	cloths (**5**)
answering questions (478) bartending (601) crying (1037) giving or receiving award (953) news anchoring (420) presenting weather forecast (1050)	communication (**6**)
baking cookies (927) barbequing (1070) breading or breadcrumbing (454)	cooking (**7**)

cooking chicken (1000) cutting pineapple (712) cutting watermelon (767) grinding meat (415)	
belly dancing (1115) country line dancing (1015) dancing ballet (1144) dancing charleston (721) dancing gangnam style (836) dancing macarena (958) marching (1146) swing dancing (512) tango dancing (1114) tap dancing (947) zumba (1093)	dancing (**11**)
bartending (601) dining (671) drinking beer (575) eating burger (864) eating cake (494) eating carrots (516) eating chips (749) eating hotdog (570) eating spaghetti (1145) eating watermelon (550)	eating + drinking (**10**)
assembling computer (542) texting (704) using computer (937) using remote controller (not gaming) (549)	electronics (**4**)
blowing leaves (405) carving pumpkin (711) mowing lawn (1147) watering plants (680)	garden + plants (**4**)
golf driving (836) golf putting (1081)	golf (**2**)
curling hair (855) fixing hair (676) waxing back (537)	hair (**3**)
applauding (411) cutting nails (560) drumming fingers (409) finger snapping (825) pumping fist (1009) washing hands (916)	hand (**6**)
balloon blowing (826) beatboxing (943) blowing nose (597) blowing out candles (1150) gargling (430) headbanging (1090) shaking head (885) singing (1147) smoking hookah (857)	head + mouth (**9**)
abseiling (1146) bungee jumping (1056) climbing ladder (662) diving cliff (1075) ice climbing (845) jumping into pool (1133)	heights (**11**)

paragliding (800) slacklining (790) springboard diving (406) swinging on something (482) trapezing (786)	
bee keeping (430) feeding goats (1027) holding snake (430) ice fishing (555) milking cow (980) riding elephant (1104) riding or walking with horse (1131) training dog (481) walking the dog (1145)	interacting with animals (**9**)
contact juggling (1135) hula hooping (1129) juggling soccer ball (484) spinning poi (1134)	juggling (**4**)
filling eyebrows (1085) getting a tattoo (737)	makeup (**2**)
high kick (825) punching bag (1150) side kick (991) sword fighting (473) tai chi (1070) wrestling (488)	martial arts (**6**)
garbage collecting (441) laying bricks (432) moving furniture (426) unloading truck (406)	miscellaneous (**4**)
driving car (1118) faceplanting (441) jogging (417) motorcycling (1142) parkour (504) pushing car (1069) riding a bike (476) skateboarding (1139) surfing crowd (876) using segway (387)	mobility - land (**10**)
crossing river (951) diving cliff (1075) jumping into pool (1133) scuba diving (968) snorkeling (1012) springboard diving (406) swimming backstroke (1077) swimming butterfly stroke (678) water sliding (420)	mobility - water (**9**)
beatboxing (943) playing accordion (925) playing bagpipes (838) playing bass guitar (1135) playing cello (1081) playing didgeridoo (787) playing guitar (1135) playing harp (1149) playing keyboard (715) playing organ (672)	music (**18**)

playing piano (691) playing recorder (1148) playing saxophone (961) playing trumpet (989) playing ukulele (1146) playing violin (1142) playing xylophone (746) singing (1147)	
bookbinding (914) counting money (674) folding napkins (874) ripping paper (605) shredding paper (403) writing (735)	paper (**6**)
taking a shower (378) washing hands (916)	personal hygiene (**2**)
flying kite (1063) playing cards (737) playing chess (850) playing poker (1134) rock scissors paper (424) shuffling cards (828) skipping rope (488)	playing games (**7**)
catching or throwing softball (842) hitting baseball (1071) playing squash or racquetball (980) playing tennis (1144)	racquet + bat sports (**4**)
biking through snow (1052) bobsledding (605) hockey stop (468) ice climbing (845) ice fishing (555) making snowman (756) shoveling snow (879) ski jumping (1051) snowboarding (937) snowkiting (1145) snowmobiling (601) tobogganing (1147)	snow + ice (**12**)
swimming backstroke (1077) swimming butterfly stroke (678)	swimming (**2**)
massaging legs (592) shaking hands (640) tickling (610)	touching person (**3**)
bending metal (410) pumping gas (544) sanding floor (574) sharpening knives (424)	using tools (**4**)
kitesurfing (794) parasailing (762) windsurfing (1114)	water sports (**3**)
waxing back (537)	waxing (**1**)
bench pressing (1106) deadlifting (805) exercising arm (416) lunge (759) pull ups (1121) yoga (1140)	**No Parent-Group (6)**

5 List of Mini SSV2 Video Classes

The complete list of 87 classes and their corresponding action-groups is shown in Table 2. The number of videos for each action class is given by the number in brackets following each class name. The number of child-actions of each action group is shown by the number in brackets following each action group name.

Table 2: List of Mini SSV2 Video Classes

Class Labels	**Action Groups**
Trying but failing to attach [something] to [something] because it doesn't stick (660) Attaching [something] to [something] (1227)	Attaching/Trying to attach (**2**)
Approaching [something] with your camera (1349) Moving away from [something] with your camera (1199) Turning the camera right while filming [something] (1239) Turning the camera left while filming [something] (1239) Moving [something] away from the camera (986) Turning the camera downwards while filming [something] (976)	Camera motions (**6**)
[Something] colliding with [something] and both are being deflected (653)	Collisions of objects (**1**)
Uncovering [something] (3004) Covering [something] with [something](3530)	Covering (**2**)
Putting [something similar to other things that are already on the table] (2339) Taking [one of many similar things on the table] (2969)	Crowd of things (**2**)
Dropping [something] onto [something] (1623) Dropping [something] next to [something] (1232) Dropping [something] in front of [something] (1222)	Dropping something (**3**)
Showing [something] behind [something] (2315)	Filming objects, without any actions (**1**)
Folding [something] (1542)	Folding something (**1**)
Hitting [something] with [something] (2234)	Hitting something with something (**1**)
Holding [something] behind [something] (1374) Holding [something] over [something] (1804)	Holding something (**2**)
Lifting up one end of [something], then letting it drop down (1850) Lifting up one end of [something] without letting it drop down (1613) Lifting [something] up completely, then letting it drop down (1851) Lifting [something] up completely without letting it drop down (1906)	Lifting and (not) dropping something (**4**)
Tilting [something] with [something] on it until it falls off (1272)	Lifting/Tilting objects with other objects on them (**1**)
Moving [something] closer to [something] (1426)	Moving two objects relative to each other (**1**)
Touching (without moving) [part] of [something] (1763)	Moving/Touching a part of something (**1**)
Opening [something] (1869) Pretending to open [something] without actually opening it (1911)	Opening or closing something (**2**)
Picking [something] up (1456)	Picking something up (**1**)
Plugging [something] into [something] but pulling it right out as you remove your hand (1176) Plugging [something] into [something] (2252)	Plugging something into something (**2**)
Poking [something] so it slightly moves (1599) Poking [something] so lightly that it doesn't or almost doesn't move (2430) Poking a stack of [something] without the stack collapsing (276) Poking a hole into [something soft] (258) Pretending to poke [something] (754) Poking a stack of [something] so the stack collapses (367) Poking a hole into [some substance] (115)	Poking something (**7**)
Pouring [something] onto [something] (403) Pouring [something] into [something] (1530) Pouring [something] into [something] until it overflows (352)	Pouring something (**3**)
Pulling [something] from right to left (1886) Pulling [something] from left to right (1908)	Pulling something (**2**)
Pulling two ends of [something] so that it separates into two pieces (313)	Pulling two ends of something (**1**)
Pushing [something] with [something] (1804) Pushing [something] off of [something] (687) Pushing [something] so that it slightly moves (2418)	Pushing something (**3**)
Laying [something] on the table on its side, not upright (950)	Putting something upright/on its side (**1**)
Putting [something] onto [something else that cannot support it] so it falls down (442) Taking [something] out of [something] (2259) Pretending to put [something] onto [something] (740) Putting [something] and [something] on the table (1353) Putting [something] onto [something] (1850) Pretending to put [something] underneath [something](373) Putting [something] next to [something] (2431) Putting [something] behind [something] (1428) Putting [something] on the edge of [something] so it is not supported and falls down (638)	Putting/Taking objects into/out of/next to other objects (**11**)

Removing [something], revealing [something] behind (1069) Pretending to put [something] next to [something] (1297)	
Letting [something] roll along a flat surface (1163) Rolling [something] on a flat surface (1773) Putting [something] that can't roll onto a slanted surface, so it slides down (442) Lifting a surface with [something] on it but not enough for it to slide down (268) Putting [something] on a flat surface without letting it roll (553)	Rolling and sliding something (**5**)
Showing [something] to the camera (1061) Showing a photo of [something] to the camera (916)	Showing objects and photos of objects (**2**)
Showing that [something] is empty (2209)	Showing that something is full/empty (**1**)
[Something] falling like a feather or paper (1858)	Something falling (**1**)
Moving [something] and [something] so they collide with each other (577)	Something passing/hitting another thing (**1**)
Spinning [something] so it continues spinning (1168)	Spinning something (**1**)
Spreading [something] onto [something] (535) Pretending to spread 'air' onto [something] (225)	Spreading something onto something (**2**)
Pretending to sprinkle 'air' onto [something] (543)	Sprinkling something onto something (**1**)
Putting number of [something] onto [something] (1180)	Stacking or placing N things (**1**)
Stuffing [something] into [something] (1998)	Stuffing/Taking out (**1**)
Pretending to take [something] from somewhere (1437)	Taking something (**1**)
Tearing [something] just a little bit (2025)	Tearing something (**1**)
Throwing [something] (2626) Throwing [something] in the air and catching it (1177) Pretending to throw [something] 1019 Throwing [something] in the air and letting it fall (1038) Throwing [something] onto a surface (1035)	Throwing something (**5**)
Tipping [something] over (896)	Tipping something over (**1**)
Twisting (wringing) [something] wet until water comes out (408) Twisting [something] (1131)	Twisting something (**2**)

6 More Training Details

We follow the training protocol in [1], as shown in Table 3, for I3D, S3D, 3D-ResNet 18, Inception V1 and Resnet 18 in our experiments. We progressively train the model with different input frames. We first train a starter model using 8 frames. The model is either inflated with (3D models) or initialized from (2D models) its corresponding ImageNet pre-trained weight. We then finetune the model using more $N(N = 16, 32, 64)$ frames from the model using $N/2$ frames. For the started model, we trained 75 epochs with cosine annealing learning rate schedule starting with 0.01. For models with more input frames, the learning rate is also set as 0.01 and is divided by 10 at every 15 epochs. The training process is ended at the 50_{th} epoch. For TAM, 3D-ResNet 50, SlowFast, and X3D models, we only trained these models using 32 frames. We use synchronized SGD with momentum 0.9 and weight decay 0.0001 for all models mentioned above.

Table 3: Training protocol

	8-frame	16-frame	32-frame	64-frame
Weight Init.	ImageNet	8-frame	16-frame	32-frame
Epochs	75	50	50	50
Learning rate	0.01			
LR scheduler	cosine	multisteps	multisteps	multisteps
Weight decay	0.0005			
Optimizer	Synchronized SGD with moment 0.9			

7 Training with Gaussian noise

Gaussian data augmentation has been widely used to improve the robustness of models in image-related tasks [3]. To examine the impact of it on corruption robustness, we train models with additive Gaussian noise on clean data for both datasets. We use the same standard deviation of 0.2 as [3]. In Table 4, we show that the clean accuracy, mPC and rPC of models trained with Gaussian data augmentation are lower than models trained on clean data. It counters the results for image corruption robustness, where the robustness is improved when the Gaussian data augmentation is applied.

Table 4: Corruption robustness of 3D ResNet-18 with/without Gaussian data augmentation on the corrupted Mini Kinetics and Mini SSV2. The std indicates the standard deviation of Gaussian noise.

				Spatial						Temporal					
Approach	Clean	mPC	rPC	Shot	Rain	Fog	Contrast	Brightness	Saturate	Motion	Frame Rate	ABR	CRF	Bit Error	Packet Loss
Mini Kinetics-C															
3D ResNet-18	**66.2**	**53.3**	**80.5**	47.7	**45.8**	40.5	**43.0**	**58.9**	**53.5**	**53.4**	**65.1**	**60.1**	**56.4**	**55.8**	**59.3**
3D ResNet-18(std=0.2)	57.8	43.8	75.8	**60.7**	37.2	**46.6**	14.0	16.9	51.2	41.4	56.4	51.4	48.4	49.3	51.9
Mini SSV2-C															
3D ResNet-18	**53.0**	**42.6**	**80.3**	34.1	21.9	**38.0**	**42.9**	**48.0**	42.5	34.9	**42.9**	**49.1**	**47.8**	**40.3**	**46.9**
3D ResNet-18(std=0.2)	45.7	35.7	78.1	**43.9**	**25.4**	28.5	25.5	26.0	**43.5**	**40.2**	34.8	42.4	43.2	34.5	40.4

Table 5: Corruption robustness of 3D ResNet-18 on the corrupted Mini Kinetics.

				Spatial						Temporal					
Model	Clean	mPC	rPC	Shot	Rain	Fog	Contrast	Brightness	Saturate	Motion	Frame Rate	ABR	CRF	Bit Error	Packet Loss
Shot Noise	71.5	61.4	85.9	**73.6**	69.2	34.8	44.2	61.2	57.0	65.0	70.1	68.3	65.6	62.2	65.2
Rain	73.7	63.1	85.6	57.6	**74.9**	36.8	55.3	66.5	59.9	68.2	72.3	69.6	67.0	63.1	66.4
Fog	41.6	41.8	**100.5**	34.3	46.3	**55.7**	52.5	34.8	28.2	49.8	46.4	42.1	39.5	35.3	36.6
Contrast	49.6	47.5	95.8	36.2	56.8	38.9	**67.1**	46.6	38.0	55.5	51.8	48.7	46.1	41.5	42.8
Brightness	71.2	59.8	84.0	51.4	63.2	33.2	48.7	**74.3**	60.7	62.9	68.6	66.8	64.0	60.4	62.9
Saturate	75.4	61.9	82.1	56.8	66.0	28.3	47.6	69.3	**68.5**	67.6	72.7	69.6	65.7	63.7	67.3
Motion Blur	71.3	60.6	85.0	50.9	61.4	39.7	53.5	61.9	57.4	**75.7**	70.4	68.2	65.8	60.0	62.6
Frame Rate	**80.3**	65.5	81.6	53.1	65.9	41.2	54.4	70.1	65.3	71.6	77.2	75.4	71.4	68.5	71.8
ABR	79.4	**66.6**	83.9	56.4	68.2	44.1	57.2	69.5	65.7	72.3	**77.3**	75.9	73.4	67.9	71.8
CRF	78.8	65.8	83.5	55.4	67.3	40.1	53.2	67.5	65.8	72.0	76.8	**76.8**	**74.4**	68.4	71.5
Bit Error	77.8	66.5	85.5	55.1	68.2	42.0	57.0	70.9	65.3	68.5	76.6	74.4	70.9	**73.4**	**75.1**
Packet Loss	77.3	64.6	83.6	56.5	67.4	36.9	52.0	69.9	64.6	67.4	74.7	72.0	68.5	71.7	73.0
Clean	78.9	65.8	83.4	57.4	70.0	39.0	54.0	70.0	66.5	71.8	76.8	74.2	70.4	67.3	72.0

8 Training with Corruption

Beyond evaluating the models trained on clean data with our proposed benchmark, we also train the network with our proposed corruptions directly. We use the backbone of 3D ResNet-18. Similar to [2], we use a subset of the standard Kinetics to train the network. It contains 40 classes comparing to 400 classes of the original dataset. The results in Table 5 show that most of the models achieve the best performance on the corruption they are trained on. However, our proposed benchmark aims to evaluate the robust generalization ability of models, under the condition that we may not know the corruption exactly in advance. Hence, we test the models on other types of corruption as well. We find that the models do not generalize to other types of corruption well. 9 out of 12 models trained on single corruption obtain lower mPC than the model trained on clean data.

Besides single corruption, different types of corruption may happen simultaneously in nature. To mimic the random process that occurs in nature, we also evaluate the performance of models on each corruption with additive white Gaussian noise. The Gaussian noise has a standard deviation of 0.1. Table 6 shows that Gaussian noise can degrade the performance of models significantly. The model trained with shot noise obtains the best performance on all types of corruptions due to its similarity to Gaussian noise.

9 Impact of Batch Size and Training Epochs

Besides 2D-3D module, input sampling strategy, and input length, we study the impact of more general hyper-parameters including batch size (16 and 32) and training epochs (25 and 50). We show the results in Table 7 and Table 8. We find these two general hyper-parameters have less impact on the performance comparing to the hyper-parameters which are specific for video classification. The clean accuracy and mPC of models are similar while the hyper-parameters differ by large.

10 Number of Clips in Video Level Evaluation

The evaluation on Mini Kinetics-C and Mini SSV2-C requires 60 times of computation cost on original validation datasets. It is necessary to find the proper number of clips at the video level in practice, which minimizes the trade-off between performance and efficiency. To find the number of clips, we train the models on Mini Kinetics using 8 frames input and evaluate them on Mini

Table 6: Corruption robustness of 3D ResNet-18 on Mini Kinetics-C with additive Gaussian noise.

				Spatial						Temporal					
Model	Clean	mPC	rPC	Shot	Rain	Fog	Contrast	Brightness	Saturate	Motion	Frame Rate	ABR	CRF	Bit Error	Packet Loss
Shot Noise	71.5	**62.3**	**87.1**	**74.0**	**68.5**	**33.2**	**45.0**	**64.2**	**59.4**	**66.1**	**71.3**	**69.3**	**66.7**	**63.3**	**66.6**
Rain	73.7	48.2	65.4	47.3	55.0	15.7	26.0	53.3	48.9	53.4	62.2	58.6	55.4	49.3	53.7
Fog	41.6	27.9	67.1	27.4	32.1	22.3	18.2	26.7	20.6	35.8	35.1	31.6	30.8	26.4	27.9
Contrast	49.6	31.3	63.1	26.3	39.5	20.5	25.4	31.2	26.3	39.7	36.7	35.5	33.8	29.4	30.7
Brightness	71.2	45.1	63.3	41.6	44.5	14.0	19.4	61.5	49.0	49.0	58.7	54.6	51.6	46.9	50.7
Saturate	75.4	47.1	62.5	44.3	47.5	12.2	18.7	58.4	54.3	51.0	63.1	58.0	54.5	49.8	53.9
Motion Blur	71.3	41.7	58.5	38.4	38.8	12.5	16.6	49.5	42.4	56.0	55.3	52.2	49.3	43.3	46.4
Frame Rate	**80.3**	41.7	51.9	38.3	35.9	10.7	14.4	53.8	42.8	43.6	59.6	52.8	48.0	48.3	52.0
ABR	79.4	44.6	56.2	43.2	41.5	10.8	15.0	53.9	48.3	49.9	60.0	56.9	53.2	48.8	53.2
CRF	78.8	43.6	55.3	40.0	37.7	8.7	15.0	55.5	47.5	48.1	60.5	55.7	51.8	48.5	54.1
Bit Error	77.8	43.2	55.5	39.1	38.2	8.0	12.7	56.6	49.0	43.2	60.8	53.6	49.4	52.2	55.9
Packet Loss	77.3	44.5	57.6	44.9	41.4	11.4	14.9	55.5	48.7	45.8	61.4	54.0	49.4	51.1	55.6
Clean	78.9	46.0	58.3	44.9	47.2	12.9	18.3	54.1	46.9	49.3	63.9	57.3	52.2	49.9	54.6

Table 7: Corruption robustness of 3D ResNet-18 trained with different batch size. The bs stands for batch size.

				Spatial						Temporal					
Batch Size	Clean	mPC	rPC	Shot	Rain	Fog	Contrast	Brightness	Saturate	Motion	Frame Rate	ABR	CRF	Bit Error	Packet Loss
bs=16	65.2	54.2	83.1	52.4	52.1	49.2	44.7	45.0	58.4	53.3	64.9	59.8	56.1	56.0	59.0
bs=32	66.2	53.3	80.5	47.7	45.8	40.5	43.0	58.9	53.5	53.4	65.1	60.1	56.4	55.8	59.3

Kinetics-C at the video level. Figure 2 shows that the models using uniform sampling saturate faster than dense sampling. The performance improvement is subtle after 4 clips. Dense sampling requires more number of clips at the video level. The model performance saturates at 8 clips. As a result, we use 4 clips for uniform sampling and 8 clips for dense sampling for all the video level evaluations.

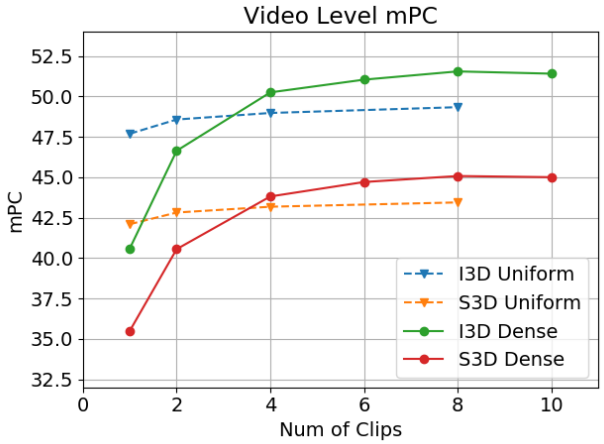

Figure 2: mPC of I3D and S3D using different number of clips at video level evaluation.

11 Full Corruption Robustness Results

The classification accuracy on each corruption with respect to each severity level in Mini Kinetics-C is in Figure 3, the accuracy on corruptions in Mini SSV2-C is in Figure 4. Both the figures shows the degrading performance when we increase the severity level of corruptions. When we compare the performance of models on Mini Kinetic-C and Mini SSV2-C horizontally, they present obvious difference in degradation on motion blur and frame rate conversion. The model trained on Mini SSV2 are more sensitive to these two temporal corruptions. Especially for frame rate conversion, the classification accuracy of models drops around 40% on Mini SSV2-C, while the performance of models on Mini Kinetics-C maintains the same.

Table 8: Corruption robustness of 3D ResNet-18 trained with different training epochs.

				Spatial						Temporal					
Epoch	Clean	mPC	rPC	Shot	Rain	Fog	Contrast	Brightness	Saturate	Motion	Frame Rate	ABR	CRF	Bit Error	Packet Loss
Epochs=25	65.2	54.6	83.7	50.1	54.6	48.1	43.2	44.7	59.2	54.5	65.8	60.7	57.3	56.9	59.8
Epochs=50	66.2	53.3	80.5	47.7	45.8	40.5	43.0	58.9	53.5	53.4	65.1	60.1	56.4	55.8	59.3

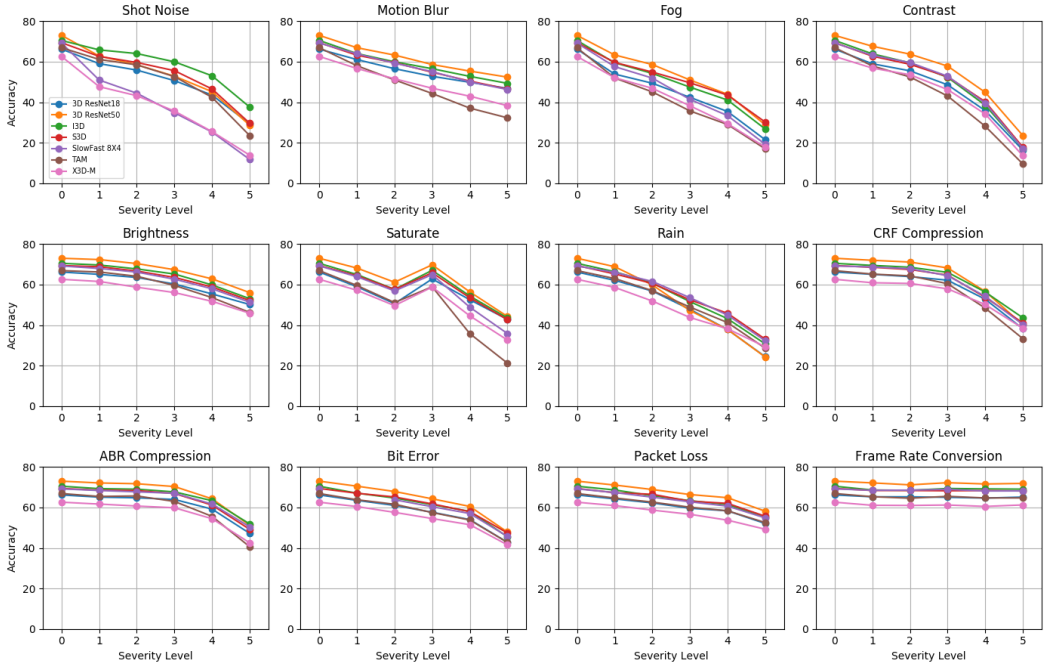

Figure 3: Model accuracy for SOTA approaches with respect to the video corruptions in Mini Kinetics-C. We evaluate the models at 5 level of severity. Severity level 0 indicates the accuracy on clean data.

12 Sample of Corruption Severity

In Figure 5, Figure 6 and Figure 7, we show different types of corruptions with five level of severity. When the severity level increase, the corruptions vary from subtle to obvious. The parameters for synthesizing the corruption are crucial for the benchmark, and they are listed in the code on `https://github.com/Newbeeyoung/Video-Corruption-Robustness`. With this range, the benchmark can represent each corruption type comprehensively.

References

[1] C.-F. Chen, R. Panda, K. Ramakrishnan, R. Feris, J. Cohn, A. Oliva, and Q. Fan. Deep analysis of cnn-based spatio-temporal representations for action recognition. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021.

[2] R. Geirhos, C. R. M. Temme, J. Rauber, H. H. Schütt, M. Bethge, and F. A. Wichmann. Generalisation in humans and deep neural networks. In *Proceedings of the Neural Information Processing Systems (NIPS)*, 2018.

[3] J. Gilmer, N. Ford, N. Carlini, and E. Cubuk. Adversarial examples are a natural consequence of test error in noise. In *International Conference on Machine Learning*, pages 2280–2289, 2019.

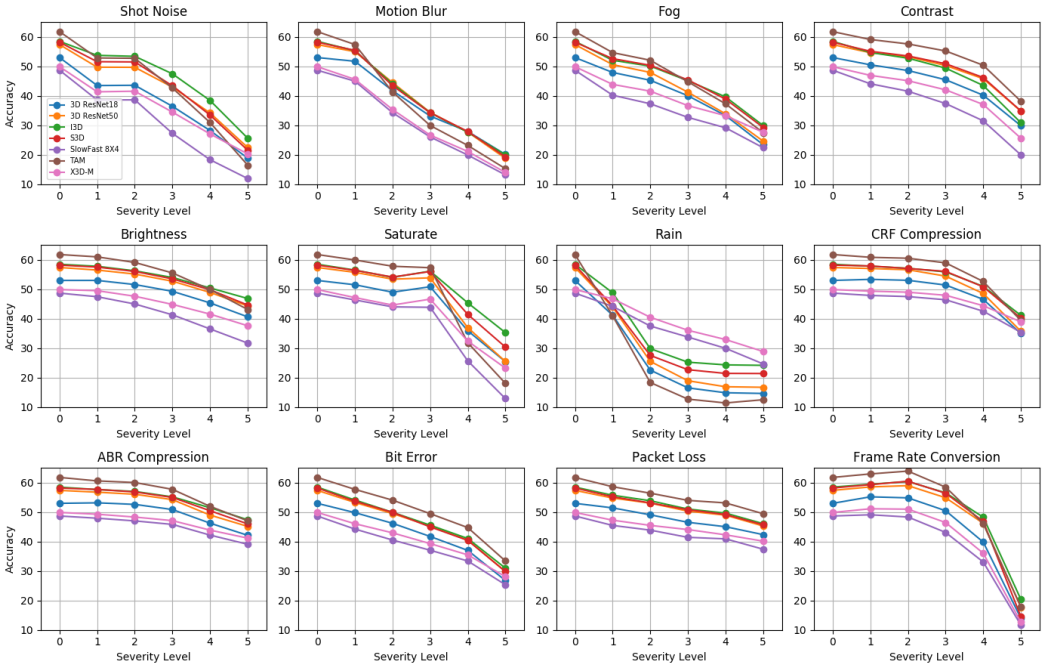

Figure 4: Model accuracy for SOTA approaches with respect to the video corruptions in Mini SSV2-C.

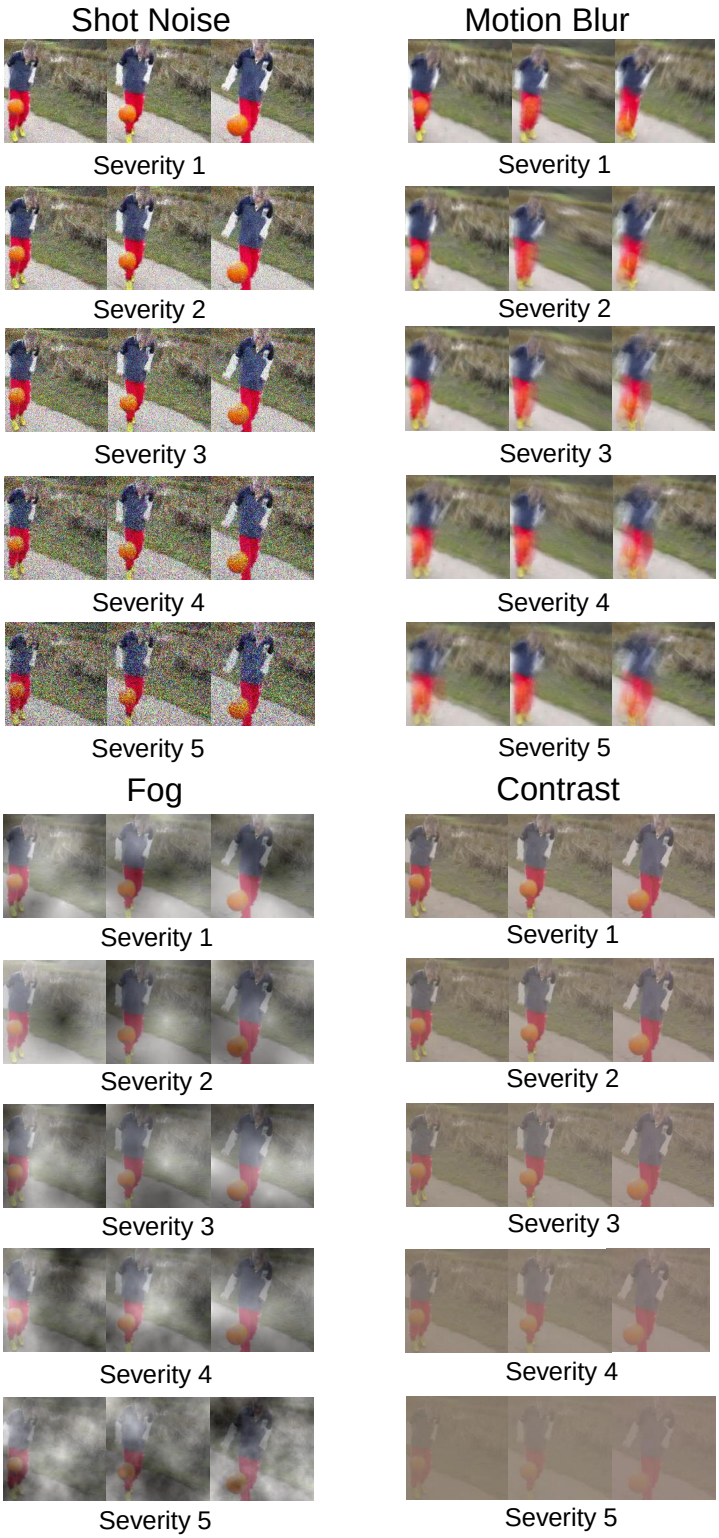

Figure 5: The visualization samples of shot noise, motion blur, fog and contrast with varying severities in Mini Kinetics-C

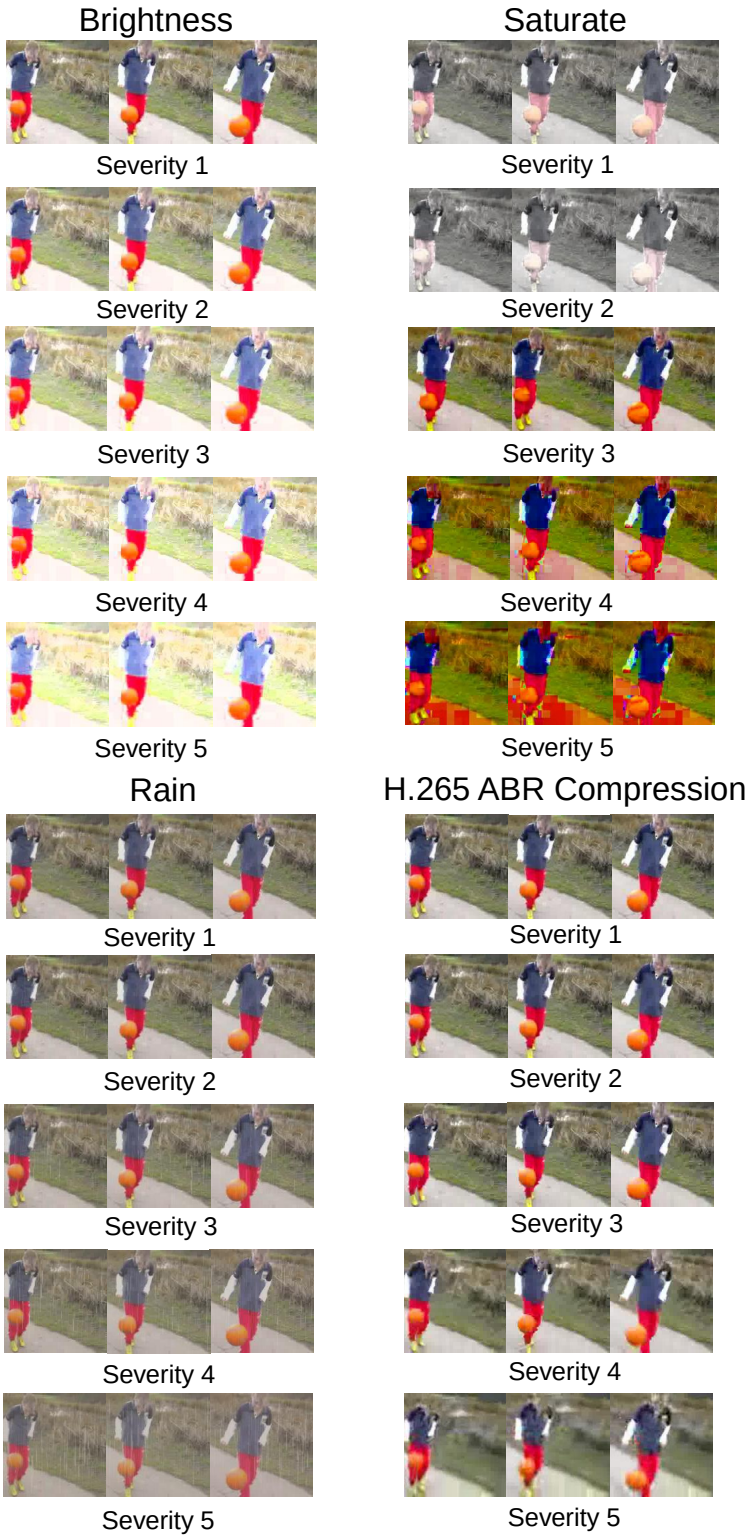

Figure 6: The visualization samples of brightness, saturate, rain and h.265 abr compression with varying severities in Mini Kinetics-C

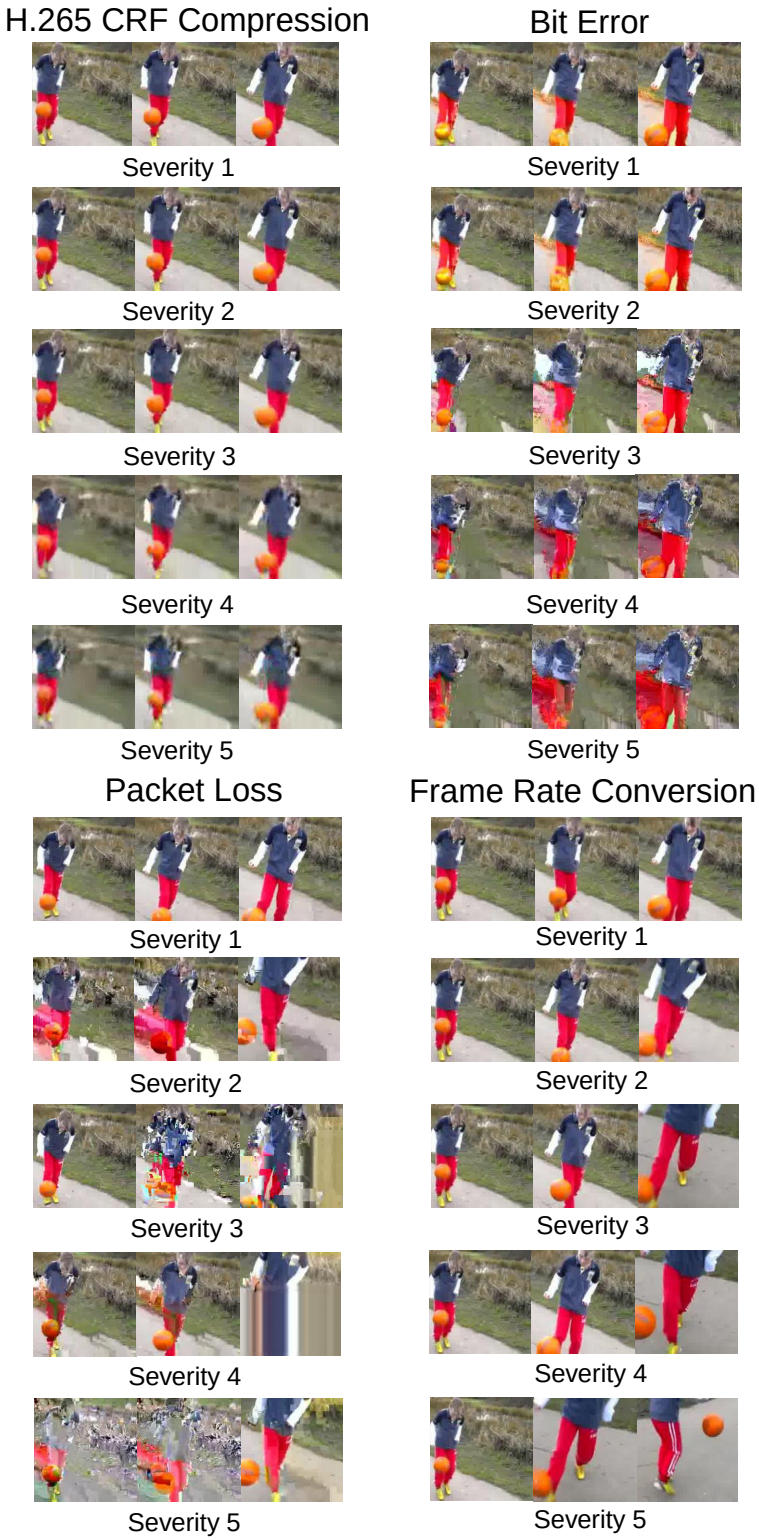

Figure 7: The visualization samples of h.265 crf compression, bit error, packet loss and frame rate conversion with varying severities in Mini Kinetics-C