Technical Report of Challenge on VideoPipe Track on Video Defect Classification

Jiawei Dong[1], Shuo Wang[1]
[1] Shanghai Paidao Intelligent Technology Co., Ltd.,
Shanghai 200240, China
Email: {Jiawei.dong, gavin.wang}@ai-prime.ai

***Abstract*—Video anomaly analysis is important for industrial applications in the real world. In particular, the urban pipe system is one of the most important infrastructures in a city. In order to ensure its normal operation, we need to inspect pipe defects smartly. This is a technical report on the video defect classification track of the ICPR VideoPipe Challenge. The report focuses on the team's data preparation, task definition, model selection, training process and inference process during the competition. The solution provides feasible adjustment strategies, and we will also provide complete training and inference code for others to reproduce.**

I. Introduction

The video classification of urban pipelines has always been a relatively complex classification task. Compared with the previous urban pipeline datasets, this task has great differences in data types, data label granularity, and dataset size. In view of the above characteristics, we propose three distinctive solutions: the frame-based method, the video-based method and the super-image-based method. Among the three methods, both video-based and super-image-based can achieve the first place on the leaderboard, and the score obtained by the super-image-based method was ten points higher than the second place. This report will focus on the descriptions of these three methods, while explaining the thinking ideas of the solution, and propose feasible strategies to further improve the score.

II. Data Preparing

The first is data preparation, including frame extraction, data distribution observation, data multi-fold splitting and data augmentation. In terms of dataset composition, none of the three solutions mentioned in this report use any extra datasets during training.

A. Frame Extracting

Here we use CV2 to extract all frames in 9601 videos, the extracted frame images retain the original resolution, and all frame images are assigned to each folder according to the video they belong to.

B. Folder Spliting

Usually we need to divide the data into multiple folders so that all samples can be learned in the model. Since this task is a multi-label classification task, and the data has a very significant long-tailed distribution, in order to divide the data set into 5 different folders, and try to ensure that the distribution of training and validation samples in the folder basically conforms to the distribution of dataset, we used the iterative stratification from scikit-multilearn library. Taking folder 0 as an example, the distribution of the split training and validation sets is shown in the figure.

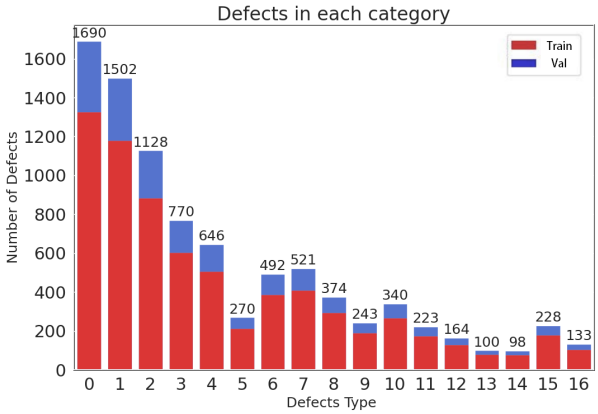

Fig. 1. Folder 0 Train and Val Set

C. Data Augmentation

In data augmentation, we mainly use horizontal flipping, we have also experimented with other data augmentation methods such as RandAug, AutoAug, and some common data augmentation methods such as vertical flipping, random cropping, rotation, color shift, etc. In the experiments, these methods will reduce our score, so we do not use these data augmentation methods.

III. Task Definition and Solutions

According to the thinking of the solution, we position the task into the following three categories.

A. Frame-Based Task

This method refers to Sewel-ML [1], which introduces a multi-label classification strategy based on urban pipe images, and provides a pipeline for reference. Next I will describe the training and validation process of this method in detail.

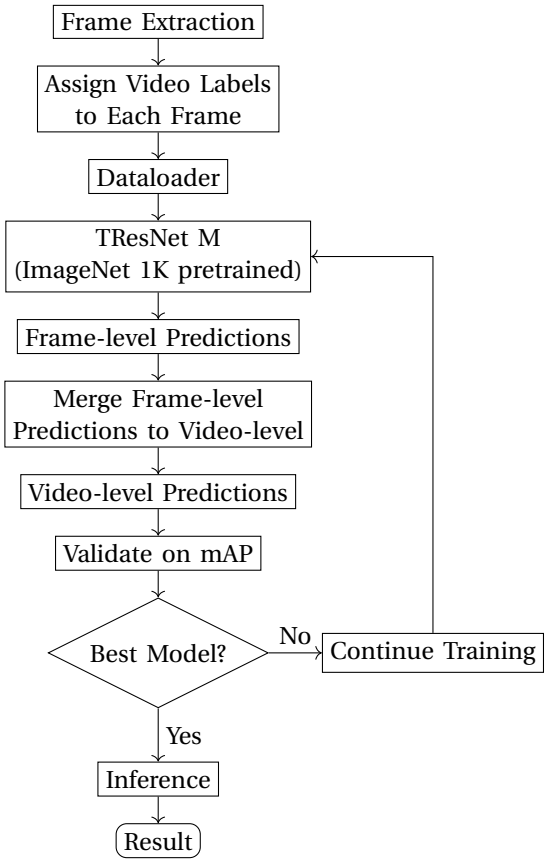

Fig. 2. Flow Chart of Frame-Based Method

1) Model: We used TResNet [2] for training model. On the leaderboard of the multi-label classification dataset, TResNet has always been in a top 3 position, mainly due to the ASL [3] used in the network. TResNet also has a very high speed on training and inference. While ensuring the accuracy of the model, the number of parameters of the model is also moderate. So in order to use it as a baseline for fast implementation, we use TResNet.

2) Training and Validation Pipeline: The overall flow of training and validation is shown in Figure 2. First, we get all the previously extracted frame images, for all the frame images of each video, assign the video labels to the frame images. Then we split the dataset into 5 folders, generate the corresponding file list and send it to the dataloader. Second, after the data preparation is completed, the TResNet image classification network is used for training. After the model is trained for several epochs, the frame-level predictions of the model for the validation dataset are collected, and post-processing methods are used to convert the frame-level predictions into video-level predictions for evaluation. After multiple rounds of training and evaluation operations, the model with the best evaluation result is selected for prediction on the test dataset.

3) Post-Process: The post-processing stage here mainly refers to the process of converting the frame-level predictions of TResNet into video-level predictions. Our post-processing method is very simple, which is to average(or maximum and median) the predictions of all frames in a video , and output it as the predictions of this video. This method is proved to be simple and effective, but it lacks rationality, and the performance is poor for some long videos.

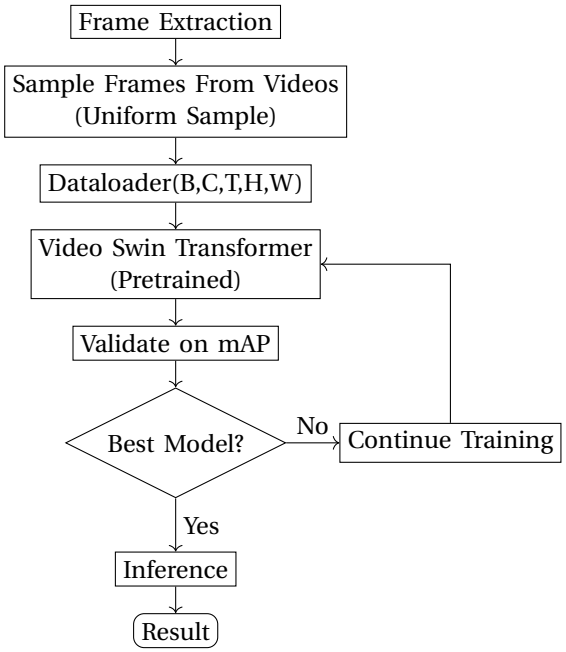

Fig. 3. Flow Chart of Video-Based Method

4) Result: Using this simple method, we achieved a score of 54.71% on leaderboard, which is a good start for us.

B. Video-Based Task

Video-based methods refer to taking the task as a regular video action classification task and using a video action classification network to train and predict it.

1) Model: We used Video Swin Transformer [4] as a model for video classification. This model is a pure-transformer architecture for video recognition that is based on spatiotemporal locality inductive bias. This model is adapted from the Swin Transformer for image recognition, and thus it could leverage the power of the strong pre-trained image models. The proposed approach achieves state-of-the-art performance on three widely-used benchmarks, Kinetics-400, Kinetics-600 and Something-Something v2.

2) Training and Validation Pipeline: The overall flow of training and validation is shown in Figure 3. The process is based on the standard video action classification network training process. First, collect all the previously extracted frame images, sample the image using uniform sample method, and send them to the dataloader. Second, it goes directly to the training and validation process of the model, so this is a complete end-to-end network. Except

TABLE I
TRAINING DETAILS AND RESULT OF VIDEO SWIN TRANSFORMER

Model	Bacbone	Params	Lr Schd	Pretrain	mAP(%)
Video Swin Transformer	Swin-B	88M	Onecycle 30E	Kinectic 600	64.512
				Kinectic 400	64.798
				SS V2	64.714

for the data preparation process, there is no other data preprocessing and postprocessing, the training process is clean, easy to understand and reproducible.

3) Result: Using video classification network based on the Video Swin Transformer, and using different Backbone for training, the mAP score on the Leaderboard reached 64.79%. Compared with the method based on single-frame prediction, video-based method boosted score by nearly 10%. Training details are shown in the Table I.

C. Super-image Task

We refer to a paper for the super-image method [5]. They propose a new perspective on video understanding by casting the video recognition problem as an image recognition task. Showing that an image classifier alone can suffice for video understanding without temporal modeling. The approach is simple and universal. It composes input frames into a super image to train an image classifier to fulfill the task of action recognition, in exactly the same way as classifying an image. They proves the viability of such an idea by demonstrating strong and promising performance on four public datasets including Kinetics400, Something-to-something (V2), MiT and Jester, using a recently developed vision transformer. They also experiment with the prevalent ResNet image classifiers in computer vision to further validate our idea. The results on Kinetics400 are comparable to some of the best-performed CNN approaches based on spatiotemporal modeling.

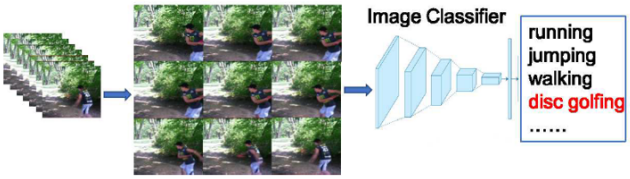

Fig. 4. Overview of SIFAR(Super-Image For Action Recognization)

In the original paper, they used a technique similar to sliding window to slide on the super-image, thereby preserving the temporal information, but in our opinion, the temporal information in this task may not be so important, so based on the original paper, we simplify the process and directly use super-image as the object of training and evaluation.

1) Pre-Process: The pre-processing here refers to the process of converting several frame images into a super-image. First, we obtain N samples from the video using uniform sampling (the size of N is determined by the number of rows and columns of the super-image. The N we use is 9, which means the super-image is a 3×3 grid). Second, data augmentation is performed on each sampled image. Third, the augmented images are collaged into a super-image according to the rows and columns of super-image, and the input processing flow is completed.

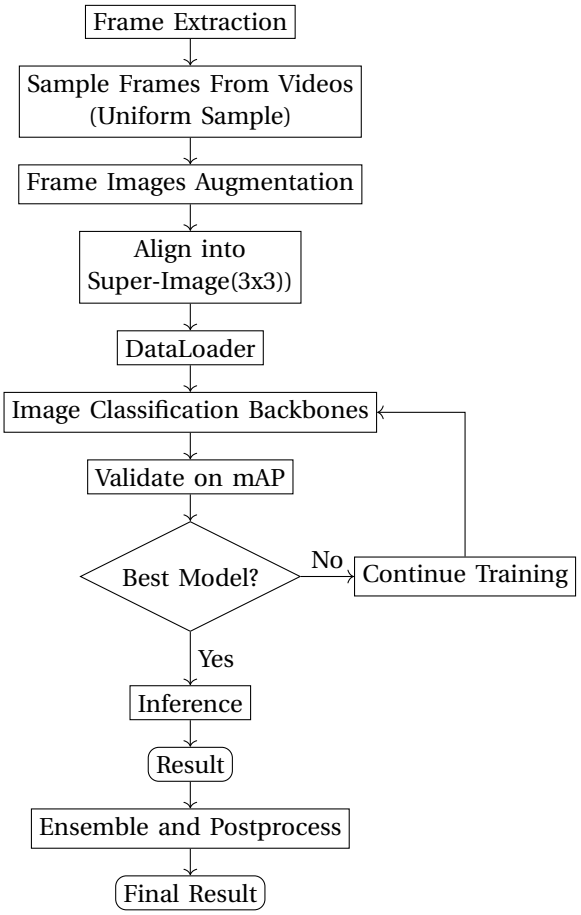

Fig. 5. Flow Chart of Super-Image Method

2) Model: Since we transform the video classification task into a normal image classification task, there are more models to choose from. In the experiment, we selected dozens of top-ranked models on ImageNet for testing. Among the models we tested, there are several models with outstanding performance, such as: Convnext Base [6], MLDecoder (TResNet XL) [7], NFNet F3, NFNet F6 [8], EfficientNet L2 [9].

In the actual model selection, the model we choose usually has the following characteristics: pre-trained on a large data set, excellent score on ImageNet, larger input size, moderate amount of parameters, good generalization ability and fitting speed.

3) Training and Validation Pipeline: The training and inference process of the super-image method is shown in the Figure 5. First, we need to extract frames from videos and obtain several frame images using uniform

TABLE II
TRAINING DETAILS AND RESULT OF SUPER-IMAGE METHOD

Model	Pretrain	Params	Input Size	Super-Img Grid	Data Aug	Optim	Lr Schd	CV mAP (%)	LB mAP (%)
Tresnet XL + MLDecoder	ImageNet 21K (Input Size 640)	78M	1334 (448*3)	3*3	Horizonal Flip + Tiles Shuffle	AdamW	OneCycle 30e	67.19	68.299
ConvNeXt Base	IN22Kft1K (Input Size 384)	88M	1334 (448*3)	3*3	Horizonal Flip + Tiles Shuffle	AdamW	OneCycle 30e	69.89	70.287
NFNET F3	ImageNet 1K (Input Size 416)	254M	1334 (448*3)	3*3	Horizonal Flip + Tiles Shuffle	AdamW	OneCycle 30e	71.41	71.109
NFNET F6	ImageNet 1K (Input Size 576)	438M	1152 (384*3)	3*3	Horizonal Flip + Tiles Shuffle	AdamW	OneCycle 30e	70.45	70.769
EfficientNet L2	ImageNet 1K (Input Size 800)	480M	1334 (448*3)	3*3	Horizonal Flip + Tiles Shuffle	AdamW	OneCycle 30e	70.95	70.853

sampling. Second, we will perform data augmentation on the sampled frame images. Third, for each video, we collage the augmented frame images to a super-image. Fourth, we send the super-images to the image classification network through dataloader for training and validation. After getting the model with the highest validation mAP, inference is performed on the test dataset. Finally, we ensemble and post-process the predictions of multiple image classification models to get our final score.

4) Result: As shown in the Table II are the five best models we tested, as well as the pre-trained models of each network, the parameters of the model, the input size, the super-image parameters, the data augmentation method, the optimizer, and the local validation score and leaderboard score, In order to ensure the effect of the later ensemble, we try to choose more heterogeneous networks. At the same time, based on these types of networks, we have conducted several ablation experiments on training strategies. Finally we fine-tuned the parameters of each network to ensure the best performance.

IV. ENSEMBLE AND POST-PROCESSING

After all the models achieved their best results, we started to ensemble the predictions. We experimented with a variety of ensemble methods, and finally found that the average ensemble and weight ensemble were the most effective. The process is, for the 5 folder predictions from same model, we use average ensemble method. After getting the predictions whose number is the same as the number of models, for predictions from different models, we use weight ensemble method, the size of the weight is LB score related. The ensemble detail are shown in the Figure III.

After we get the final ensemble prediction, we need to post-process it. The post-processing here refers to, for each prediction, if prob of 'ZC' above 0.9, set prob of 'ZC' to 1, set other prob of classes to 0. After post-processing, our score reaches 72.181%, which is also our final submission score.

V. BOOSTING TRICKS

We summarize some tricks used in the competition, but it should be noted that boosted value only represents the boosted score of some models, not all of models. Because we use a variety of models, we cannot guarantee the boosted score is completely accurate with the table on every model, and part of the boosted score is an estimate.

First is the tricks that boosted the score. At the data level, a larger input size is more effective. At the data augmentation level, only horizontal flipping and super-image tiles disruption can improve the score. In terms of sampling strategy, the most effective method is uniform sampling. At the model level, we have proved that longer warmup epoch, larger learning rate, and the use of onecycle scheduler are all effective. Due to the large input size of the super-image, in order to increase the actual batch size, we also used some tricks like gradient accumulation, mixed precision, gradient checkpoint. We also used exponential moving average on model weights, which can greatly boost the score of our model. On the ensemble strategy, only the average ensemble and the weighted ensemble are proved to be stable and effective. Finally the post-processing trick mentioned above can also slightly improve the score.

TABLE III
MODEL ENSEMBLE RESULT

Model	LB mAP(%)	Ensemble Weight	Ensembled LB mAP(%) (post-processed)
Tresnet XL + MLDecoder	68.299	0.1	72.181
ConvNeXt Base	70.287	0.15	
NFNET F3	71.109	0.25	
NFNET F6	70.769	0.2	
EfficientNet L2	70.853	0.2	
Video Swin Transformer(ema)	68.251	0.1	

We also found some strategies that will lower our score, such as heavier data augmentation, using sequence sampling, using super-images with larger rows and columns, using TTA at inference, etc.

We also use some weakly supervised learning methods,

TABLE IV
BOOSTING TRICKS

Level	Type	Description	Boosted(%)
Data	Size	Large input size(448)	1
	Augment	Horizonal flip	0.6
		Tiles shuffle	1
	Sample	Uniform sample	2.2
Model	Learning Strategy	Long warmup epoch	0.9
		Big learning rate	1.8
		Onecycle scheduler	0.5
	Batch Strategy	Accumulate gradients	2
		Mixed precision	
		Gradient checkpoint	
	Other	Ema models	5
Ensemble	5 folders ensemble, mix folder ensemble		1.8
Postprocess	For each prediction, if prob of 'ZC' above 0.9, set prob of 'ZC' to 1, set other prob of classes to 0.		0.12

TABLE V
LOWERING TRICKS

Level	Type	Description	Boosted(%)
Data	Augment	Randaug	-1
		Autoaug	-0.6
		Rotate, vertical flip, color jitter	-1.6
	Sample	Sequence sample	-2.2
		Larger super-image grid(4x4, 5x5)	-1
Model	Weakly Supervised Model	SimCLR + TransMIL	-19.4 (Local Val)
		SimCLR + MLDecoder	-19.7 (Local Val)
		MAE + TransMIL	-22 (Local Val)
		MAE + MLDecoder	-25 (Local Val)
TTA	Horizonal flip, Vertical flip		-3
	Resample video		-0.3
	Grid shuffle		-0.5
Ensemble	Ensemble by max mAP of each class		-1.6
Postprocess	Set threshold for each class		-1.3

trying to convert this task into a weakly supervised multi-instance learning problem. The main process is as follows: First, use MAE, SimCLR and other self-supervised networks to pre-train on the dataset. Second, use pre-trained self-supervised networks to extract features from images. Third, use the extracted features as the input of TransMIL and MLdecoder for training and validation. After such process the final score can only reach our frame-based methods.

VI. CONCLUSIONS

- Predictions based on frame images are not very effective and interpretable. Due to the characteristic of the pipeline, it cannot validate actual mAP score during training, but only after post-processing.
- The prediction results based on video classification are relatively general, and the training takes a lot of time. Considering that the model may have learned some unimportant temporal information. In fact, the network does not need to identify various defects as actions, and only needs to learn features in single frame image.
- It is also possible to transform this task into a weakly supervised multi-instance learning task, but pre-training of feature extractors such as MAE and SimCLR is a critical step, and they are also time-consuming. If the feature extractor can be pre-trained well on the dataset of similar domain, the score can definitely be improved a lot.
- The super-image-based method is the most interpretable and effective in this task. The video is used as a large image for training and prediction, network only need to pay attention to the defects of the tile in the large image. At the same time, we also found that random disruption of super-image tiles can even improve our score, which also proves that temporal information is not very important in this task.

REFERENCES

[1] J. B. Haurum and T. B. Moeslund, "Sewer-ml: A multi-label sewer defect classification dataset and benchmark," 2021. [Online]. Available: https://arxiv.org/abs/2103.10895

[2] T. Ridnik, H. Lawen, A. Noy, E. B. Baruch, G. Sharir, and I. Friedman, "Tresnet: High performance gpu-dedicated architecture," 2020. [Online]. Available: https://arxiv.org/abs/2003.13630

[3] E. Ben-Baruch, T. Ridnik, N. Zamir, A. Noy, I. Friedman, M. Protter, and L. Zelnik-Manor, "Asymmetric loss for multi-label classification," 2020. [Online]. Available: https://arxiv.org/abs/2009.14119

[4] Z. Liu, J. Ning, Y. Cao, Y. Wei, Z. Zhang, S. Lin, and H. Hu, "Video swin transformer," 2021. [Online]. Available: https://arxiv.org/abs/2106.13230

[5] Q. Fan, Chun-Fu, Chen, and R. Panda, "Can an image classifier suffice for action recognition?" 2021. [Online]. Available: https://arxiv.org/abs/2106.14104

[6] Z. Liu, H. Mao, C.-Y. Wu, C. Feichtenhofer, T. Darrell, and S. Xie, "A convnet for the 2020s," 2022. [Online]. Available: https://arxiv.org/abs/2201.03545

[7] T. Ridnik, G. Sharir, A. Ben-Cohen, E. Ben-Baruch, and A. Noy, "Ml-decoder: Scalable and versatile classification head," 2021. [Online]. Available: https://arxiv.org/abs/2111.12933

[8] A. Brock, S. De, S. L. Smith, and K. Simonyan, "High-performance large-scale image recognition without normalization," 2021. [Online]. Available: https://arxiv.org/abs/2102.06171

[9] Q. Xie, M.-T. Luong, E. Hovy, and Q. V. Le, "Self-training with noisy student improves imagenet classification," 2019. [Online]. Available: https://arxiv.org/abs/1911.04252

APPENDIX A
MODELS TRAINING AND TESTING RESULT

TABLE VI
MODEL PERFORMANCE ON FOLDER, CATEGORIES AND LEADERBOARD

MODELS	Folder	Best mAP(%)	Categories AP(%)																LB mAP(%)
			0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
MLDecoder (TResNet-XL, ImageNet 21K)	0	66.98	100.00	82.19	70.56	60.60	83.80	81.48	56.83	71.71	63.60	67.72	75.89	76.84	54.70	12.53	53.91	48.25	78.00	68.299
	1	65.02	99.86	79.76	64.84	57.31	83.27	80.74	52.01	81.73	63.81	58.11	76.57	70.22	60.47	35.90	36.58	43.55	60.66	
	2	69.84	99.99	80.09	74.37	60.37	84.59	86.27	57.78	74.92	61.29	71.01	67.13	76.07	65.94	51.20	53.15	38.87	84.31	
	3	67.97	100.00	78.37	69.58	57.40	82.08	77.91	57.03	75.88	59.23	73.23	64.80	85.37	66.79	48.60	42.83	46.45	69.95	
	4	66.16	100.00	81.16	71.38	64.12	78.85	78.74	65.89	83.32	68.56	51.00	65.07	62.07	85.46	34.39	40.43	30.27	64.07	
ConvNeXt-Base (IN22Kft1K)	0	69.48	99.98	81.58	72.17	61.15	81.00	78.06	62.89	78.53	72.74	69.68	68.55	89.75	63.06	18.59	57.64	44.06	81.63	70.287
	1	67.80	99.97	80.18	64.82	63.03	80.54	91.67	58.94	84.57	76.43	55.97	70.96	75.13	69.38	38.25	42.36	37.28	63.15	
	2	72.19	99.99	81.72	74.12	65.18	84.93	85.40	65.41	82.68	70.19	70.50	63.63	73.64	77.24	55.13	51.32	40.02	86.11	
	3	71.58	99.98	80.87	70.23	61.50	80.65	82.88	64.85	80.64	67.61	74.56	61.99	84.81	68.83	43.29	61.24	53.88	78.98	
	4	68.42	99.99	80.33	69.10	71.20	80.64	82.80	63.25	83.70	74.91	47.17	64.37	61.92	89.03	41.18	46.42	39.70	67.36	
NFNet F3 (ImageNet 1K)	0	71.09	99.95	83.59	74.61	62.32	80.77	81.48	71.11	80.44	78.35	72.78	71.80	90.28	71.13	20.86	38.48	49.11	81.56	71.109
	1	69.93	99.99	81.96	66.74	66.77	83.40	92.06	67.11	85.27	74.33	58.04	77.93	75.75	72.32	33.27	48.39	43.24	62.17	
	2	74.95	99.99	84.53	75.69	68.22	86.92	88.12	65.78	81.15	75.54	73.33	68.82	83.07	85.82	47.26	63.17	41.16	85.50	
	3	71.79	100.00	80.17	71.19	59.30	82.53	83.57	65.33	83.31	70.29	79.16	61.84	82.09	69.25	53.03	59.81	46.46	73.11	
	4	69.27	100.00	83.18	72.83	69.08	81.72	82.53	70.45	82.34	73.44	50.36	63.52	64.19	87.71	46.65	42.03	40.41	67.17	
NFNet F6 (ImageNet 1K)	0	69.51	99.98	82.65	74.47	63.57	81.89	78.18	65.60	79.79	77.31	71.72	67.78	90.10	65.23	16.38	40.80	47.57	78.68	70.769
	1	69.00	99.91	81.59	65.47	65.41	82.82	91.10	59.38	85.78	76.25	58.62	77.92	77.47	71.32	36.71	39.78	42.57	60.92	
	2	72.06	99.95	81.38	75.64	63.10	86.91	86.92	66.76	82.23	72.13	71.93	67.27	76.99	71.86	46.52	51.55	36.21	87.65	
	3	71.55	99.94	81.42	71.30	62.41	83.17	82.59	65.12	83.86	72.37	73.77	59.99	84.25	75.55	46.46	52.07	48.77	73.33	
	4	70.15	100.00	84.67	73.75	69.08	83.76	84.89	66.99	84.80	74.60	56.90	63.30	67.29	90.16	44.90	45.24	37.67	64.48	
Video Swin Transformer (ema)	0	68.82	100.00	83.24	69.63	60.53	82.50	82.44	63.93	78.46	76.07	71.36	69.94	87.09	62.63	19.90	35.57	43.69	83.02	68.251
	1	67.55	99.99	80.72	61.48	63.10	79.14	92.24	52.02	83.34	79.24	61.37	73.64	70.41	66.29	40.42	40.08	39.59	65.23	
	2	71.31	99.94	82.32	73.99	63.30	84.15	88.66	59.03	79.66	72.40	70.19	68.78	74.00	73.34	40.29	54.63	39.83	87.83	
	3	70.57	99.99	79.45	69.49	58.39	83.61	81.61	59.95	79.31	68.00	76.70	69.48	80.30	69.74	48.07	49.51	55.80	70.21	
	4	68.87	99.99	81.60	69.50	71.32	80.64	81.85	67.82	79.26	73.95	52.53	65.22	65.09	84.62	52.68	36.00	40.14	68.60	
EfficientNet L2	0	70.62	99.92	83.05	72.25	61.17	79.90	80.23	61.92	77.21	77.87	76.09	69.36	93.55	64.55	32.06	41.52	51.03	78.82	70.853
	1	69.76	100.00	81.07	67.45	63.45	80.53	89.51	65.84	87.65	79.04	61.33	76.40	74.15	72.48	39.68	39.28	45.76	62.30	
	2	73.68	99.98	82.84	76.72	62.30	84.87	90.52	67.48	83.07	73.12	70.73	69.06	80.81	76.16	49.23	58.23	39.12	88.37	
	3	70.50	99.88	79.56	70.76	58.05	79.71	79.86	66.21	82.50	67.72	74.80	59.68	84.82	65.90	55.81	46.60	53.07	73.58	
	4	70.17	100.00	83.28	70.83	70.93	81.85	81.85	68.98	85.05	74.90	59.53	60.85	69.75	86.44	49.71	39.49	44.80	64.59	