Appendix

This appendix provides further details for the main paper:

§A contains further *results* for COCO object detection (§A.1) AVA action detection (§A.2) and ImageNet classification (§A.3), as well as *ablations* for ImageNet classification and COCO object detection (§A.4) and Kinetics action classification (§A.5).

§B contains additional MViTv2 *upgrade details* (§B.1), and additional *implementation details* for: ImageNet classification (§B.2), COCO object detection (§B.3), Kinetics action classification (§B.4), SSv2 action classification (§B.5), and AVA action detection (§B.6).

A. Additional Results

A.1. Results: COCO Object Detection

System-level comparsion on COCO. Table A.1 shows the system-level comparisons on COCO data. We compare our results with previous state-of-the-art models. We adopt Soft-NMS [4] during inference, following [55]. MViTv2-L* achieves 58.7 AP^{box} with multi-scale testing, which is already +0.7 AP better than the best results of Swin-L* that relies on the improved HTC++ detector [55].

model	framework	AP^{box}	AP^{mask}	Flops	Param
Copy-Paste [26]	Cascade, NAS-FPN	55.9	47.2	1440	185
Swin-L [55]	HTC++	57.1	49.5	1470	284
Swin-L [55]*	HTC++	58.0	50.4	-	284
MViTv2-L	Cascade	56.9	48.6	1519	270
MViTv2-L*	Cascade	**58.7**	**50.5**	-	270

Table A.1. **System-level comparison on COCO object detection and segmentation**. The detection frameworks include Cascade Mask R-CNN [6] (Cascade), the improved Hybrid Task Cascade (HTC++) [55] and Cascade Mask R-CNN with NAS-FPN [27]. * indicates multi-scale testing. FLOPs and Params are in Giga (10^9) and Mega (10^6).

A.2. Results: AVA Action Detection

Results on AVA. Table A.2 shows the results of our MViTv2 models compared with prior state-of-the-art works on the AVA dataset [32] which is a dataset for spatiotemporal-localization of human actions.

We observe that MViT consistently achieves better results compared to MViTv1 [21] counterparts. For example, MViTv2-S 16×4 (26.8 mAP) improves +2.3 over MViTv1-B 16×4 (24.5 mAP) with fewer flops and parameters (both with the same recipe and default K400 pre-training). For K600 pre-training, MViTv2-B 32×3 (29.9 mAP) improves +1.2 over MViTv1-B-24 32×3. This again validates the effectiveness of the proposed MViTv2 improvements in §4.1 of the main paper. Using *full-resolution* testing (without cropping) can further improve MViTv2-B by +0.6 to achieve 30.5 mAP. Finally, the larger MViTv2-L 40×3 achieves the state-of-the-art results at 34.4 mAP using IN-21K and K700 pre-training.

model	pretrain	val mAP center	val mAP full	FLOPs	Param
SlowFast, 4×16, R50 [23]	K400	21.9	-	52.6	33.7
SlowFast, 8×8, R101 [23]	K400	23.8	-	137.7	53.0
MViTv1-B, 16×4 [21]	K400	24.5	-	70.5	36.4
MViTv1-B, 64×3 [21]	K400	27.3	-	454.7	36.4
MViTv2-S, 16×4	K400	26.8	27.6	64.5	34.3
MViTv2-B, 32×3	K400	**28.1**	**29.0**	225.2	51.0
SlowFast, 8×8 R101+NL [23]	K600	27.1	-	146.6	59.2
SlowFast, 16×8 R101+NL [23]	K600	27.5	-	296.3	59.2
X3D-XL [22]	K600	27.4	-	48.4	11.0
Object Transformer [80]	K600	**31.0**	-	243.8	86.2
ACAR 8×8, R101-NL [60]	K600	-	**31.4**	N/A	N/A
MViTv1-B, 16×4 [21]	K600	26.1	-	70.4	36.3
MViTv1-B-24, 32×3 [21]	K600	28.7		236.0	52.9
MViTv2-B, 32×3	K600	29.9	30.5	225.2	51.0
ACAR 8×8, R101-NL [60]	K700	-	**33.3**	N/A	N/A
MViTv2-B, 32×3	K700	**31.3**	32.3	225.2	51.0
MViTv2-L↑ 312^2, 40×3	IN21K+K700	**33.5**	**34.4**	2828	213.0

Table A.2. **Comparison with previvous work on AVA v2.2**. We adopt two test strategies: 1) *center* (*single center crop*): we resize the shorter spatial side to 224 pixels and takes a 224^2 center crop for inference. 2) *full* (*full-resolution*): we resize the shorter spatial side to 224 pixels and take the full image for inference. We report inference cost with the *center* testing strategy (*i.e.* 224^2 input). Magnitudes are Giga (10^9) for FLOPs and Mega (10^6) for Param.

A.3. Results: ImageNet Classification

Results of ImageNet-1K. Table A.3 shows the comparison of our MViTv2 with *more* prior work (without external data or distillation models) on ImageNet-1K. As shown in the Table, our MViTv2 achieves better results than any previously published methods for a variety of model complexities. We note that our improvements to pooling attention bring significant gains over the MViTv1 [21] counterparts which use exactly the same training recipes (for all datasets we compare on); therefore the gains over MViTv1 stem solely from our technical improvements in §4.1 of the main paper.

A.4. Ablations: ImageNet and COCO

Decomposed relative position embeddings. As introduced in Sec. 4.1, our Relative position embedding is only applied for Q_i by default. We could further extend it to all Q, K and V terms for attention layers:

$$\mathrm{Attn}(Q, K, V) = AV + E^{(\mathrm{rel_v})},$$
$$\text{where} \quad A = \mathrm{Softmax}\left((QK^\top + E^{(\mathrm{rel_q})} + E^{(\mathrm{rel_k})})/\sqrt{d}\right).$$

model	Acc center	Acc resize	FLOPs (G)	Param (M)
RegNetY-4GF [62]	80.0		4.0	21
RegNetZ-4GF [15]	**83.1**		4.0	28
EfficientNet-B4 ↑ 380^2 [71]	82.9		4.2	19
DeiT-S [72]	79.8		4.6	22
PVT-S [78]	79.8		3.8	25
TNT-S [33]	81.5		5.2	24
T2T-ViT$_t$-14 [85]	81.7		6.1	22
CvT-13 [81]	81.6		4.5	20
Twins-S [11]	81.7		2.9	24
ViL-S-RPB [89]	82.4		4.9	25
PVTv2-V2 [77]	82.0		4.0	25
CrossViT$_c$-15 [9]	82.3		6.1	28
XCiT-S12 [18]	82.0		4.8	26
Swin-T [55]	81.3		4.5	29
CSWin-T [16]	82.7		4.3	23
MViTv2-T	82.3		4.7	24
RegNetY-8GF [62]	81.7		8.0	39
EfficientNet-B5 ↑ 456^2 [71]	**83.6**		9.9	30
PVT-M [78]	81.2		6.7	44
T2T-ViT$_t$-19 [85]	82.4		9.8	39
CvT-21 [81]	82.5		7.1	32
Twins-B [11]	83.2		8.6	56
ViL-M-RPB [89]	83.5		8.7	40
PVTv2-V2-B3 [77]	83.2		6.9	45
CrossViT$_c$-18 [9]	82.8		9.5	44
XCiT-S24 [18]	82.6		9.1	48
Swin-S [55]	83.0		8.7	50
CSWin-S [16]	**83.6**		6.9	35
MViT-v1-B-16 [21]	83.0		7.8	37
MViTv2-S	**83.6**		7.0	35
RegNetY-16GF [62]	82.9		15.9	84
RegNetZ-16GF [15]	84.1		15.9	95
EfficientNet-B6 ↑ 528^2 [71]	84.2		19	43
NFNet-F0 ↑ 256^2 [5]	83.6		12.4	72
DeiT-B [72]	81.8		17.6	87
PVT-L [78]	81.7		9.8	61
T2T-ViT$_t$-21 [85]	82.6		15.0	64
TNT-B [33]	82.9		14.1	66
Twins-L [11]	83.7		15.1	99
ViL-B-RPB [89]	83.7		13.4	56
PVTv2-V2-B5 [77]	83.8		11.8	82
CaiT-S36 [74]	83.3		13.9	68
XCiT-M24 [18]	82.7		16.2	84
Swin-B [55]	83.3		15.4	88
CSWin-B [16]	84.2		15.0	78
MViTv1-B-24 [21]	83.4		10.9	54
MViTv2-B	**84.4**		10.2	52
EfficientNet-B7 ↑ 600^2 [71]	84.3		37.0	66
NFNet-F1 ↑ 320^2 [5]	84.7		35.5	133
DeiT-B ↑ 384^2 [72]	83.1		55.5	87
TNT-B ↑ 384^2 [33]	83.9		N/A	66
CvT-32 ↑ 384^2 [81]		83.3	24.9	32
CaiT-S36↑ 384^2 [74]		85.0	48	68
Swin-B ↑ 384^2 [55]		84.2	47.0	88
MViT-v1-B-24 ↑ 320^2 [21]	84.8		32.7	73
MViTv2-B ↑ 384^2	**85.2**	**85.6**	36.7	52
NFNet-F2 ↑ 352^2 [5]	85.1		62.6	194
XCiT-M24 [18]	82.9		36.1	189
CoAtNet-3 [13]	84.5		34.7	168
MViTv2-L	**85.3**		42.1	218
NFNet-F4 ↑ 512^2 [5]	85.9		215.3	316
CoAtNet-3 [13] ↑ 384^2		85.8	107.4	168
MViTv2-L ↑ 384^2	**86.0**	**86.3**	140.2	218

Table A.3. **Comparison to previous work on ImageNet-1K**. Input images are 224×224 by default and ↑ denotes using different sizes. MViT is trained for 300 epochs without any external data or models. We report our ↑ 384^2 models tested using a *center* crop or a *resized full* crop of the original image, to compare to prior work.

rel pos rel_q	rel pos rel_k	rel pos rel_v	IN-1K Acc	IN-1K Mem(G)	IN-1K Test (im/s)	COCO AP^{box}	COCO AP^{mask}
✓	×	×	83.6	6.2	316	49.9	45.0
×	✓	×	83.4	6.2	321	49.7	44.8
✓	✓	×	83.6	6.4	300	50.0	45.0
×	×	✓	83.6	30.8	109	OOM	OOM
✓	×	✓	83.7	30.9	104	OOM	OOM
✓	✓	✓	83.6	30.9	103	OOM	OOM

Table A.4. **Ablation of rel pos embeddings** on ImageNet-1K and COCO with MViT-S.

And the rel pos terms are defined as:

$$
\begin{aligned}
E_{ij}^{(\text{rel}_q)} &= Q_i \cdot R^q_{p(i),p(j)}, \\
E_{ij}^{(\text{rel}_k)} &= R^k_{p(i),p(j)} \cdot K_i, \\
E_i^{(\text{rel}_v)} &= \sum_j A_{ij} * R^v_{p(i),p(j)}.
\end{aligned}
$$

Table A.4 shows the ablation experiments: different variants achieve similar accuracy on ImageNet and COCO. However rel_v requires more GPU memory (*e.g.* 30.8G vs 6.2G on ImageNet and out-of-memory (OOM) on COCO) and has a ~2.9×lower test throughput on ImageNet. For simplicity and efficiency, we use only rel_q by default.

Effect of pre-training datasets for detection. In §6.2 of the main paper we observe that ImageNet pre-training can have very different effects for different model sizes for video classification. Here, we are interested in the impact of pre-training on the larger IN-21K *vs*. IN-1K for COCO *object detection tasks*. Table A.5 shows our ablation: The large-scale IN-21K pre-training is more helpful for larger models, *e.g.* MViT-B and MViT-L have +0.5 and +0.9 gains in AP^{box}.

variant	AP^{box} IN-1k	AP^{box} IN-21k	AP^{mask} IN-1k	AP^{mask} IN-21k
MViTv2-S	49.9	50.2	45.1	45.1
MViTv2-B	51.0	51.5	45.7	46.4
MViTv2-L	51.8	52.7	46.2	46.8

Table A.5. **Effect of pre-training datasets for COCO.** Detection methods are initialized from IN-1K or IN-21K pre-trained weights.

A.5. Ablations: Kinetics Action Classification

In §5.3 of the main paper we ablated the impact of our improvements to pooling attention, *i.e.* decomposed relative positional embeddings & residual pooling connections, for image classification and object detection. Here, we ablate the effect of our improvements for video classification.

Positional embeddings for video. Table A.6 compares different positional embeddings for MViTv2 on K400. Similar to image classification and object detection (Table 6 of the main paper), relative positional embeddings surpass absolute

	rel. pos. space	rel. pos. time	abs. pos.	Top-1 (%)	Train (clip/s)	Param (M)
(*1*) no pos.				80.1	**91.5**	**34.4**
(*2*) abs. pos.			✓	80.4	91.0	34.7
(*3*) time-only rel.		✓		80.8	80.5	**34.4**
(*4*) space-only rel.	dec.			80.6	76.2	34.5
(*5*) dec. space rel. + time rel.	dec.	✓		81.0	66.6	34.5
(*6*) joint space rel. + time rel.	joint	✓		**81.1**	33.6	37.1
(*7*) joint space/time rel.	joint			-	8.4	73.7

Table A.6. **Ablation of positional embeddings** on K400 with MViTv2-S 16×4. Training throughput is measured by average clips per-second with 8 V100 GPUs. Our (*5*) *decomposed space/time rel.* positional embeddings are accurate and significantly faster than other joint versions. Note that we do not finish the full training for (*7*) *joint space/time rel.* as the training speed is too slow (∼8× slower than ours) and (*6*) *joint space rel.* already shows large drawbacks (∼2× slower) of joint rel. positional embeddings.

positional embeddings by ∼0.6% comparing (*2*) and (*5*, *6*). Comparing (*5*) to (*6*), our *decomposed space/time rel.* positional embeddings achieve nearly the same accuracy as the *joint space rel.* embeddings while being ∼**2**× *faster* in training. For *joint space/time rel.* (*5 vs. 7*), our *decomposed space/time rel.* is even ∼**8**×faster with ∼**2**×fewer parameters. This demonstrates the effectiveness of our decomposed design for relative positional embeddings.

Residual pooling connection for video. Table A.7 studies the effect of residual pooling connections on K400. We observe similar results as for image classification and object detection (Table 7 of the main paper), that: both Q pooling blocks and residual paths are *essential* in our improved MViTv2 and combining them together leads to **+1.7**% accuracy on K400 while using them separately only improves slightly (+0.4%).

residual pooling	Top-1	FLOPs
(*1*) w/o	79.3	64
(*2*) full Q pooling	79.7	65
(*3*) residual	79.7	64
(*4*) full Q pooling + residual	**81.0**	65

Table A.7. **Ablation of residual pooling connections** on K400 with MViTv2-S 16×4 architecture.

B. Additional Implementation Details

B.1. Other Upgrades in MViT

Besides the technical improvements introduced in §4.1 of the main paper, MViT entails two further changes: (i) We conduct the channel dimension expansion in the *attention computation* of the first transformer block of each stage, instead of performing it in the last MLP block of the prior stage as in MViTv1 [21]. This change has similar accuracy (±0.1%) to the original version, while reducing parameters and FLOPs. (ii) We remove the class token in MViT by default as this has no advantage for image classification tasks. Instead, we average the output tokens from the last transformer block and apply the final classification head upon it. In practice, we find this modification could reduce the training time by ∼8%.

B.2. Details: ImageNet Classification

IN-1K training. We follow the training recipe of MViTv1 [21,72] for IN-1K training. We train for 300 epochs with 64 GPUs. The batch size is 32 per GPU by default. We use truncated normal distribution initialization [35] and adopt synchronized AdamW [58] optimization with a base learning rate of 2×10^{-3} for batch size of 2048. We use a linear warm-up strategy in the first 70 epochs and a decayed half-period cosine schedule [72].

For regularization, we set weight decay to 0.05 for MViTv2-T/S/B and 0.1 for MViTv2-L/H and label-smoothing [70] to 0.1. Stochastic depth [41] (*i.e.* drop-path or drop-connect) is also used with rate 0.1 for MViTv2-T & MViTv2-S, rate 0.3 for MViTv2-B, rate 0.5 for MViTv2-L and rate 0.8 for MViTv2-H. Other data augmentations have the same (default) hyperparameters as in [21, 73], including mixup [88], cutmix [87], random erasing [91] and rand augment [12].

For 384×384 input resolution, we fine-tune the models trained on 224×224 resolution. We decrease the batch size to 8 per GPU and fine-tune 30 epochs with a base learning rate of 4×10^{-5} per 256 batch-size samples. For MViTv2-L and MViTv2-H, we disable mixup and fine-tune with a learning rate of 5×10^{-4} per batch of 64. We linearly scale learning rates with the number of overall GPUs (*i.e.* the overall batch-size).

IN-21K pre-training and fine-tuning on IN-1K. We download the latest winter-2021 version of IN-21K from the official website. The training recipe follows the IN-1K training introduced above except for some differences described next. We train the IN-21K models on the joint set of IN-21K and 1K for 90 epochs (60 epochs for MViTv2-H) with a 6.75×10^{-5} base learning rate for MViTv2-S and MViTv2-B, and 10^{-4} for MViTv2-L and MViTv2-H, per batch-size of 256. The weight decay is set as 0.01 for MViTv2-S and MViTv2-B, and 0.1 for MViTv2-L and MViTv2-H.

When fine-tuning IN-21K MViTv2 models on IN-1K for MViTv2-L and MViTv2-H, we disable mixup and fine-tune for 30 epochs with a learning rate of 7×10^{-5} per batch of 64. We use a weight decay of 5×10^{-2}. The MViTv2-H ↑ 512^2 model is initialized from the 384^2 variant and trained for 3 epochs with mixup enabled and weight decay of 10^{-8}.

B.3. Details: COCO Object Detection

For object detection experiments, we adopt two typical object detection framework: Mask R-CNN [36] and Cascade

Mask R-CNN [6] in Detectron2 [82]. We follow the same training settings from [55]: multi-scale training (scale the shorter side in $[480, 800]$ while longer side is smaller than 1333), AdamW optimizer [58] ($\beta_1, \beta_2 = 0.9, 0.999$, base learning rate 1.6×10^{-4} for base size of 64, and weight decay of 0.1), and 3×schedule (36 epochs). The drop path rate is set as $0.1, 0.3, 0.4$, 0.5 and 0.6 for MViTv2-T, MViTv2-S, MViTv2-B, MViTv2-L and MViTv2-H, respectively. We use PyTorch's automatic mixed precision during training.

For the stronger recipe for MViTv2-L and MViTv2-H in Table. 5 of the main paper, we use *large-scale jittering* (1024×1024 resolution) as the training augmentation [26] and a longer schedule (50 epochs) with IN-21K pre-training.

B.4. Details: Kinetics Action Classification

Training from scratch. We follow the training recipe and augmentations from [19, 21] when training from scratch for Kinetics datasets. We adopt synchronized AdamW [58] and train for 200 epochs with 2 repeated augmentation [40] on 128 GPUs. The mini-batch size is 4 clips per GPU. We adopt a half-period cosine schedule [57] of learning rate decaying. The base learning rate is set as 1.6×10^{-3} for 512 batch-size. We use weight decay of 0.05 and set drop path rate as 0.2 and 0.3 for MViTv2-S and MViTv2-B.

For the input clip, we randomly sample a clip (T frames with a temporal stride of τ; denoted as $T \times \tau$ [23]) from the full-length video during training. For the spatial domain, we use Inception-style [69] cropping (randomly resize the input *area* between a [min, max], scale of [0.08, 1.00], and jitter aspect ratio between 3/4 to 4/3). Then we take an $H \times W =$ 224×224 crop as the network input.

During inference, we apply two testing strategies following [21, 23]: *(i)* Temporally, uniformly samples K clips (*e.g.* K=5) from a video. *(ii)* in spatial axis, scales the shorter spatial side to 256 pixels and takes a 224×224 center crop or 3 crops of 224×224 to cover the longer spatial axis. The final score is averaged over all predictions.

For the input clips, we perform the same data augmentations across all frames, including random horizontal flip, mixup [88] and cutmix [87], random erasing [91], and rand augment [12].

For Kinetics-600 and Kinetics-700, all hyper-parameters are *identical* to K400.

Fine-tuning from ImageNet. When using IN-1K or IN-21K as pre-training, we adopt the initialization scheme introduced in §4.3 of the main paper and shorter training schedules. For example, we train 100 epochs with base learning rate as 4.8×10^{-4} for 512 batch-size when fine-tuning from IN-1K for MViTv2-S and MViTv2-B, and 75 epochs with base learning as 1.6×10^{-4} when fine-tuning from IN-21K. For long-term models with 40×3 sampling, we initialize from the 16×4 counterparts, disable mixup, train for 30 epochs with learning rate of 1.6×10^{-5} at batch-size of 128, and use a weight decay of 10^{-8}.

B.5. Details: Something-Something V2 (SSv2)

The SSv2 dataset [31] contains 169k training, and 25k validation videos with 174 human-object interaction classes. We fine-tune the pre-trained Kinetics models and take the same recipe as in [21]. Specifically, we train for 100 epochs (40 epochs for MViTv2-L) using 64 or 128 GPUs with 8 clips per GPU and a base learning rate of 0.02 (for batch size of 512) with half-period cosine decay [57]. We adopt synchronized SGD and use weight decay of 10^{-4} and drop path rate of 0.4. The training augmentation is the same as Kinetics in §B.4, except we disable random flipping and repeated augmentations in training.

We use the segment-based input frame sampling [21, 52] (split each video into segments, and sample one frame from each segment to form a clip). During inference, we take a single clip with 3 spatial crops to form predictions over a single video.

B.6. Details: AVA Action Detection

The AVA action detection dataset [32] assesses the spatiotemporal localization of human actions in videos. It has 211k training and 57k validation video segments. We evaluate methods on AVA v2.2 and use mean Average Precision (mAP) metric on 60 classes as is standard in prior work [23].

We use MViTv2 as the backbone and follow the same detection architecture in [21, 23] that adapts Faster R-CNN [64] for video action detection. Specifically, we extract region-of-interest (RoI) features [29] by frame-wise RoIAlign [36] on the spatiotemporal feature maps from the last MViTv2 layer. The RoI features are then max-pooled and fed to a per-class, sigmoid classifier for action prediction.

The training recipe is identical to [21] and summarized next. We pre-train our MViTv2 models on Kinetics. The region proposals are identical to the ones used in [21, 23]. We use proposals that have overlaps with ground-truth boxes by IoU > 0.9 for training. The models are trained with synchronized SGD training on 64 GPUs (8 clips per GPU). The base learning rate is set as 0.6 with a half-period cosine schedule of learning rate decaying. We train for 30 epochs with linear warm-up [30] for the first 5 epochs and use a weight decay of 1×10^{-8} and drop-path rate of 0.4.

C. Additional Discussions

Societal impact. Our MViTv2 is a general vision backbone for various vision tasks, including image recognition, object detection, instance segmentation, video classification and video detection. Though we are not providing any direct applications, it could potentially apply to a wide range of vision-related applications, which then might have a wide

range of societal impacts. On the positive side, the better vision backbone could potentially improve the performance of many different computer vision applications, *e.g.* visual inspection and quality management in manufacturing, cancer and tumor detection in healthcare, and vehicle re-identification and pedestrian detection in transportation.

On the other hand, the advanced vision recognition technologies could also have potential negative societal impact if they are adopted by harmful or mismanaged applications, *e.g.* usage in surveillance systems that violate privacy. It is important to be aware when vision technologies are deployed in practical applications.

Limitations. Our MViTv2 is a general vision backbone and we demonstrate its effectiveness on various recognition tasks. To reduce the full hyperparameter tuning space for MViTv2 on different datasets and tasks, we mainly follow the existing standard recipe for each task from the community (*e.g.* [21, 55, 73]) with lightweight tuning (*e.g.* learning rate, weight decay). Therefore, the choice of hyperparameters for different MViTv2 variants may be suboptimal.

In addition, MViTv2 provides five different variants from tiny to huge models with different complexity as a general backbone. In the future, we think there are two potential interesting research directions: scaling down MViTv2 to even smaller models for mobile applications, and scaling up MViTv2 to even larger models for large-scale data scenarios.

References

[1] Anurag Arnab, Mostafa Dehghani, Georg Heigold, Chen Sun, Mario Lučić, and Cordelia Schmid. Vivit: A video vision transformer. *arXiv preprint arXiv:2103.15691*, 2021. 1, 2, 8

[2] Josh Beal, Eric Kim, Eric Tzeng, Dong Huk Park, Andrew Zhai, and Dmitry Kislyuk. Toward transformer-based object detection. *arXiv preprint arXiv:2012.09958*, 2020. 1

[3] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? *arXiv preprint arXiv:2102.05095*, 2021. 2, 8

[4] Navaneeth Bodla, Bharat Singh, Rama Chellappa, and Larry S Davis. Soft-nms–improving object detection with one line of code. In *Proc. ICCV*, 2017. 6, 9

[5] Andrew Brock, Soham De, Samuel L Smith, and Karen Simonyan. High-performance large-scale image recognition without normalization. *arXiv preprint arXiv:2102.06171*, 2021. 5, 10

[6] Zhaowei Cai and Nuno Vasconcelos. Cascade r-cnn: Delving into high quality object detection. In *Proc. CVPR*, 2018. 2, 5, 6, 9, 12

[7] João Carreira, Eric Noland, Chloe Hillier, and Andrew Zisserman. A short note on the kinetics-700 human action dataset. *arXiv preprint arXiv:1907.06987*, 2019. 7

[8] Joao Carreira and Andrew Zisserman. Quo vadis, action recognition? a new model and the kinetics dataset. In *Proc. CVPR*, 2017. 2, 4, 7

[9] Chun-Fu Chen, Quanfu Fan, and Rameswar Panda. Crossvit: Cross-attention multi-scale vision transformer for image classification. In *Proc. ICCV*, 2021. 4, 10

[10] Yunpeng Chen, Haoqi Fang, Bing Xu, Zhicheng Yan, Yannis Kalantidis, Marcus Rohrbach, Shuicheng Yan, and Jiashi Feng. Drop an octave: Reducing spatial redundancy in convolutional neural networks with octave convolution. *arXiv preprint arXiv:1904.05049*, 2019. 2

[11] Xiangxiang Chu, Zhi Tian, Yuqing Wang, Bo Zhang, Haibing Ren, Xiaolin Wei, Huaxia Xia, and Chunhua Shen. Twins: Revisiting the design of spatial attention in vision transformers. In *NIPS*, 2021. 2, 5, 10

[12] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proc. CVPR*, 2020. 11, 12

[13] Zihang Dai, Hanxiao Liu, Quoc V Le, and Mingxing Tan. Coatnet: Marrying convolution and attention for all data sizes. *arXiv preprint arXiv:2106.04803*, 2021. 5, 10

[14] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proc. CVPR*, pages 248–255. Ieee, 2009. 2, 4

[15] Piotr Dollár, Mannat Singh, and Ross Girshick. Fast and accurate model scaling. In *Proc. CVPR*, 2021. 2, 5, 10

[16] Xiaoyi Dong, Jianmin Bao, Dongdong Chen, Weiming Zhang, Nenghai Yu, Lu Yuan, Dong Chen, and Baining Guo. Cswin transformer: A general vision transformer backbone with cross-shaped windows. *arXiv preprint arXiv:2107.00652*, 2021. 5, 10

[17] 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. 1, 2, 5

[18] Alaaeldin El-Nouby, Hugo Touvron, Mathilde Caron, Piotr Bojanowski, Matthijs Douze, Armand Joulin, Ivan Laptev, Natalia Neverova, Gabriel Synnaeve, Jakob Verbeek, et al. Xcit: Cross-covariance image transformers. *arXiv preprint arXiv:2106.09681*, 2021. 5, 10

[19] Haoqi Fan, Yanghao Li, Bo Xiong, Wan-Yen Lo, and Christoph Feichtenhofer. PySlowFast. https://github.com/facebookresearch/slowfast, 2020. 2, 7, 12

[20] Haoqi Fan, Tullie Murrell, Heng Wang, Kalyan Vasudev Alwala, Yanghao Li, Yilei Li, Bo Xiong, Nikhila Ravi, Meng Li, Haichuan Yang, Jitendra Malik, Ross Girshick, Matt Feiszli, Aaron Adcock, Wan-Yen Lo, and Christoph Feichtenhofer. PyTorchVideo: A deep learning library for video understanding. In *Proceedings of the 29th ACM International Conference on Multimedia*, 2021. https://pytorchvideo.org/. 2

[21] Haoqi Fan, Bo Xiong, Karttikeya Mangalam, Yanghao Li, Zhicheng Yan, Jitendra Malik, and Christoph Feichtenhofer. Multiscale vision transformers. In *Proc. ICCV*, 2021. 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13

[22] Christoph Feichtenhofer. X3D: Expanding architectures for efficient video recognition. In *Proc. CVPR*, pages 203–213, 2020. 2, 8, 9

[23] Christoph Feichtenhofer, Haoqi Fan, Jitendra Malik, and Kaiming He. SlowFast networks for video recognition. In *Proc. ICCV*, 2019. 2, 7, 8, 9, 12

[24] Christoph Feichtenhofer, Axel Pinz, and Richard Wildes. Spatiotemporal residual networks for video action recognition. In *NIPS*, 2016. 4

[25] Christoph Feichtenhofer, Axel Pinz, and Andrew Zisserman. Convolutional two-stream network fusion for video action recognition. In *Proc. CVPR*, 2016. 2

[26] Golnaz Ghiasi, Yin Cui, Aravind Srinivas, Rui Qian, Tsung-Yi Lin, Ekin D Cubuk, Quoc V Le, and Barret Zoph. Simple copy-paste is a strong data augmentation method for instance segmentation. In *Proc. CVPR*, 2021. 6, 9, 12

[27] Golnaz Ghiasi, Tsung-Yi Lin, and Quoc V Le. Nas-fpn: Learning scalable feature pyramid architecture for object detection. In *Proc. CVPR*, 2019. 9

[28] Rohit Girdhar, Joao Carreira, Carl Doersch, and Andrew Zisserman. Video action transformer network. In *Proc. CVPR*, 2019. 2

[29] Ross Girshick. Fast R-CNN. In *Proc. ICCV*, 2015. 2, 12

[30] Priya Goyal, Piotr Dollár, Ross Girshick, Pieter Noordhuis, Lukasz Wesolowski, Aapo Kyrola, Andrew Tulloch, Yangqing Jia, and Kaiming He. Accurate, large minibatch SGD: training ImageNet in 1 hour. *arXiv:1706.02677*, 2017. 12

[31] Raghav Goyal, Samira Ebrahimi Kahou, Vincent Michalski, Joanna Materzynska, Susanne Westphal, Heuna Kim, Valentin Haenel, Ingo Fruend, Peter Yianilos, Moritz Mueller-Freitag, et al. The "Something Something" video database for learning and evaluating visual common sense. In *ICCV*, 2017. 7, 12

[32] Chunhui Gu, Chen Sun, David A. Ross, Carl Vondrick, Caroline Pantofaru, Yeqing Li, Sudheendra Vijayanarasimhan, George Toderici, Susanna Ricco, Rahul Sukthankar, Cordelia Schmid, and Jitendra Malik. AVA: A video dataset of spatio-temporally localized atomic visual actions. In *Proc. CVPR*, 2018. 9, 12

[33] Kai Han, An Xiao, Enhua Wu, Jianyuan Guo, Chunjing Xu, and Yunhe Wang. Transformer in transformer. In *NIPS*, 2021. 5, 10

[34] Zhang Hang, Chongruo Wu, Zhongyue Zhang, Yi Zhu, Zhi Zhang, Haibin Lin, and Yue Sun. Resnest: Split-attention networks. 2020. 2

[35] Boris Hanin and David Rolnick. How to start training: The effect of initialization and architecture. *arXiv preprint arXiv:1803.01719*, 2018. 11

[36] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask R-CNN. In *Proc. ICCV*, 2017. 1, 3, 5, 6, 11, 12

[37] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proc. CVPR*, 2015. 1

[38] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proc. CVPR*, 2016. 2, 6

[39] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *Proc. ECCV*, 2016. 2

[40] Elad Hoffer, Tal Ben-Nun, Itay Hubara, Niv Giladi, Torsten Hoefler, and Daniel Soudry. Augment your batch: Improving generalization through instance repetition. In *Proc. CVPR*, pages 8129–8138, 2020. 12

[41] Gao Huang, Yu Sun, Zhuang Liu, Daniel Sedra, and Kilian Q Weinberger. Deep networks with stochastic depth. In *Proc. ECCV*, 2016. 11

[42] Boyuan Jiang, MengMeng Wang, Weihao Gan, Wei Wu, and Junjie Yan. Stm: Spatiotemporal and motion encoding for action recognition. In *Proc. CVPR*, pages 2000–2009, 2019. 2

[43] Zihang Jiang, Qibin Hou, Li Yuan, Daquan Zhou, Xiaojie Jin, Anran Wang, and Jiashi Feng. Token labeling: Training a 85.5% top-1 accuracy vision transformer with 56m parameters on imagenet. *arXiv preprint arXiv:2104.10858*, 2021. 5

[44] 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:1705.06950*, 2017. 2, 7

[45] Dan Kondratyuk, Liangzhe Yuan, Yandong Li, Li Zhang, Mingxing Tan, Matthew Brown, and Boqing Gong. MoViNets: Mobile video networks for efficient video recognition. In *Proc. CVPR*, 2021. 8

[46] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet classification with deep convolutional neural networks. In *NIPS*, 2012. 2

[47] Yann LeCun, Bernhard Boser, John Denker, Donnie Henderson, Richard Howard, Wayne Hubbard, and Lawrence Jackel. Handwritten digit recognition with a back-propagation network. In *NIPS*, 1989. 3

[48] Yann LeCun, Bernhard Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne Hubbard, and Lawrence D Jackel. Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989. 2

[49] Yan Li, Bin Ji, Xintian Shi, Jianguo Zhang, Bin Kang, and Limin Wang. Tea: Temporal excitation and aggregation for action recognition. In *Proc. CVPR*, pages 909–918, 2020. 8

[50] Yanghao Li, Saining Xie, Xinlei Chen, Piotr Dollar, Kaiming He, and Ross Girshick. Benchmarking detection transfer learning with vision transformers. *arXiv preprint arXiv:2111.11429*, 2021. 7

[51] Zhenyang Li, Kirill Gavrilyuk, Efstratios Gavves, Mihir Jain, and Cees GM Snoek. VideoLSTM convolves, attends and flows for action recognition. *Computer Vision and Image Understanding*, 166:41–50, 2018. 2

[52] Ji Lin, Chuang Gan, and Song Han. Temporal shift module for efficient video understanding. In *Proc. ICCV*, 2019. 12

[53] Tsung-Yi Lin, Piotr Dollár, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *Proc. CVPR*, 2017. 1, 2, 3

[54] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. Microsoft COCO: Common objects in context. In *Proc. ECCV*, 2014. 4, 5

[55] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer:

Hierarchical vision transformer using shifted windows. *arXiv preprint arXiv:2103.14030*, 2021. 1, 2, 3, 4, 5, 6, 7, 9, 10, 12, 13

[56] Ze Liu, Jia Ning, Yue Cao, Yixuan Wei, Zheng Zhang, Stephen Lin, and Han Hu. Video swin transformer. *arXiv preprint arXiv:2106.13230*, 2021. 2, 8

[57] Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. *arXiv:1608.03983*, 2016. 12

[58] Ilya Loshchilov and Frank Hutter. Fixing weight decay regularization in adam. 2018. 11, 12

[59] Daniel Neimark, Omri Bar, Maya Zohar, and Dotan Asselmann. Video transformer network. *arXiv preprint arXiv:2102.00719*, 2021. 2, 8

[60] Junting Pan, Siyu Chen, Mike Zheng Shou, Yu Liu, Jing Shao, and Hongsheng Li. Actor-context-actor relation network for spatio-temporal action localization. In *Proc. CVPR*, 2021. 9

[61] Zhaofan Qiu, Ting Yao, and Tao Mei. Learning spatio-temporal representation with pseudo-3d residual networks. In *Proc. ICCV*, 2017. 2

[62] Ilija Radosavovic, Raj Prateek Kosaraju, Ross Girshick, Kaiming He, and Piotr Dollár. Designing network design spaces. In *Proc. CVPR*, June 2020. 2, 5, 10

[63] Joseph Redmon, Santosh Divvala, Ross Girshick, and Ali Farhadi. You only look once: Unified, real-time object detection. In *Proc. CVPR*, 2016. 2

[64] Shaoqing Ren, Kaiming He, Ross Girshick, and Jian Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. In *NIPS*, 2015. 12

[65] Peter Shaw, Jakob Uszkoreit, and Ashish Vaswani. Self-attention with relative position representations. *arXiv preprint arXiv:1803.02155*, 2018. 3

[66] Karen Simonyan and Andrew Zisserman. Two-stream convolutional networks for action recognition in videos. In *NIPS*, 2014. 2

[67] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proc. ICLR*, 2015. 1, 2

[68] Robin Strudel, Ricardo Garcia, Ivan Laptev, and Cordelia Schmid. Segmenter: Transformer for semantic segmentation. *arXiv preprint arXiv:2105.05633*, 2021. 1

[69] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proc. CVPR*, 2015. 2, 12

[70] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. *arXiv:1512.00567*, 2015. 11

[71] Mingxing Tan and Quoc V Le. Efficientnet: Rethinking model scaling for convolutional neural networks. *arXiv preprint arXiv:1905.11946*, 2019. 2, 5, 10

[72] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. Training data-efficient image transformers & distillation through attention. *arXiv preprint arXiv:2012.12877*, 2020. 4, 5, 10, 11

[73] Hugo Touvron, Matthieu Cord, Matthijs Douze, Francisco Massa, Alexandre Sablayrolles, and Hervé Jégou. DeiT: Data-efficient image transformers. *arXiv preprint arXiv:2012.12877*, 2020. 1, 2, 11, 13

[74] Hugo Touvron, Matthieu Cord, Alexandre Sablayrolles, Gabriel Synnaeve, and Hervé Jégou. Going deeper with image transformers. *arXiv preprint arXiv:2103.17239*, 2021. 5, 10

[75] Du Tran, Heng Wang, Lorenzo Torresani, and Matt Feiszli. Video classification with channel-separated convolutional networks. In *Proc. ICCV*, 2019. 2

[76] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. *arXiv preprint arXiv:1706.03762*, 2017. 1, 2

[77] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pvtv2: Improved baselines with pyramid vision transformer. *arXiv preprint arXiv:2106.13797*, 2021. 5, 10

[78] Wenhai Wang, Enze Xie, Xiang Li, Deng-Ping Fan, Kaitao Song, Ding Liang, Tong Lu, Ping Luo, and Ling Shao. Pyramid vision transformer: A versatile backbone for dense prediction without convolutions. In *IEEE ICCV*, 2021. 1, 2, 6, 10

[79] Chao-Yuan Wu, Christoph Feichtenhofer, Haoqi Fan, Kaiming He, Philipp Krähenbühl, and Ross Girshick. Long-term feature banks for detailed video understanding. In *Proc. CVPR*, 2019. 2

[80] Chao-Yuan Wu and Philipp Krahenbuhl. Towards long-form video understanding. In *Proc. CVPR*, 2021. 9

[81] Haiping Wu, Bin Xiao, Noel Codella, Mengchen Liu, Xiyang Dai, Lu Yuan, and Lei Zhang. Cvt: Introducing convolutions to vision transformers. *arXiv preprint arXiv:2103.15808*, 2021. 4, 5, 10

[82] Yuxin Wu, Alexander Kirillov, Francisco Massa, Wan-Yen Lo, and Ross Girshick. Detectron2. https://github.com/facebookresearch/detectron2, 2019. 5, 6, 12

[83] Saining Xie, Ross Girshick, Piotr Dollár, Zhuowen Tu, and Kaiming He. Aggregated residual transformations for deep neural networks. In *Proc. CVPR*, 2017. 6

[84] Saining Xie, Chen Sun, Jonathan Huang, Zhuowen Tu, and Kevin Murphy. Rethinking spatiotemporal feature learning for video understanding. *arXiv:1712.04851*, 2017. 2

[85] Li Yuan, Yunpeng Chen, Tao Wang, Weihao Yu, Yujun Shi, Zi-Hang Jiang, Francis E.H. Tay, Jiashi Feng, and Shuicheng Yan. Tokens-to-token vit: Training vision transformers from scratch on imagenet. In *Proc. ICCV*, 2021. 10

[86] Li Yuan, Qibin Hou, Zihang Jiang, Jiashi Feng, and Shuicheng Yan. Volo: Vision outlooker for visual recognition, 2021. 5

[87] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proc. ICCV*, 2019. 11, 12

[88] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. Mixup: Beyond empirical risk minimization. In *Proc. ICLR*, 2018. 11, 12

[89] Pengchuan Zhang, Xiyang Dai, Jianwei Yang, Bin Xiao, Lu Yuan, Lei Zhang, and Jianfeng Gao. Multi-scale vision longformer: A new vision transformer for high-resolution image encoding. In *Proc. ICCV*, 2021. 2, 5, 6, 10

[90] Sixiao Zheng, Jiachen Lu, Hengshuang Zhao, Xiatian Zhu, Zekun Luo, Yabiao Wang, Yanwei Fu, Jianfeng Feng, Tao Xiang, Philip HS Torr, et al. Rethinking semantic segmentation from a sequence-to-sequence perspective with transformers. In *Proc. CVPR*, 2021. 1

[91] Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. Random erasing data augmentation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 13001–13008, 2020. 11, 12

[92] Bolei Zhou, Alex Andonian, Aude Oliva, and Antonio Torralba. Temporal relational reasoning in videos. In *ECCV*, 2018. 2

[93] Xingyi Zhou, Dequan Wang, and Philipp Krähenbühl. Objects as points. *arXiv preprint arXiv:1904.07850*, 2019. 2