Supplementary Material for paper *“Bandit based Attention Mechanism in Vision Transformers”*

1. Experimental setting

We report the detailed experimental settings used while pre-training and fine-tuning the models, in Tables 1, 3, and 4. In the setting of self-supervised learning, we have used the same hyperparameters as discussed in the original paper for both DINO [2] and EsViT [8].

Table 1. Summary of training method used to pre-train our proposed approach on Imagenet-1k and fine-tune on three other datasets namely Imagenet-200 [5], CIFAR-10 [7] and CIFAR-100 [7]

Procedure →	UCB	Downstream task
	Pretrain.	Finetune.
Batch size	256	256
Optimizer	Adam	SGD
LR	3.10^{-3}	3.10^{-4}
LR decay	cosine	cosine
Weight decay	0.02	0.02
Warmup epochs	5	5
Label smoothing ε	0.1	0.1
% Dropout	0.1	0.1
Stoch. Depth	✓	✓
Repeated Aug	✓	✓
Gradient Clip.	1.0	1.0
H. flip	✓	✓
RRC	✓	✓
Rand Augment	✓	✓
LayerScale	✓	✓
Mixup alpha	✓	✓
Cutmix alpha	1.0	1.0
Erasing prob.	✓	✓
ColorJitter	0.3	0.3
Test crop ratio	1.0	1.0
Loss	CE	CE

2. Additional Experiments

2.1. Image experiments

In the main submission, we have finetuned three datasets Imagenet-200 [5], Cifar-10 [7], and CIFAR-100 [7]. In this supplementary, we report the results of the additional experiments conducted on several other datasets as shown in Table 3. A detailed description of the dataset along with the Train, Test split is given in Table 2

Table 2. Detailed list of the datasets along with Train-Test size used for finetuning

Dataset	Classes	Train size	Test size
Image Classification			
Imagenet-200 [5]	200	1,00,000	10,000
CIFAR-10 [7]	10	50,000	10,000
CIFAR-100 [7]	100	50,000	10,000
Describable Textures [3]	47	3,760	1,880
Oxford-IIIT Pets [10]	37	3,680	3,669
Oxford Flowers 102 [9]	102	2,040	6,149
STL10 [4]	10	5,000	8,000
Audio Classification			
DCASE19 [6]	10	9700	4157
ESC-50 [11]	50	1600	400
FSC22 [1]	27	1420	606

Table 3. Comparison between SOTA models pre-trained models on Imagenet-1k. We finetune these models on various small datasets. With our method, we were able to beat the baseline results. The baseline results were trained using the training techniques in DeiT-III.

Model name	Image classificaion			
	Describable Textures [3]	Oxford IIIT Pets [10]	Oxford Flowers 102 [9]	STL10 [4]
ViT-B-16	99.5	99.1	98.6	97.8
ViT-B-32	98.1	98.0	96.4	96.7
ViT-L-16	99.1	99.0	98.9	97.9
ViT-L-32	97.9	98.3	97.3	97.1
ViT-H-14	97.7	97.6	96.7	95.6
ViT-B-16-UCB	99.8	100	100	100
ViT-B-32-UCB	97.1	97.6	97.8	98.1
ViT-L-16-UCB	99.2	98.8	99.0	99.1
ViT-L-32-UCB	97.5	97.2	98.6	98.9
ViT-H-14-UCB	98.8	99.1	99.3	99.4

2.2. Audio Experiments

We have also conducted an audio classification experiment on three datasets. We have extracted the audio spectrograms from the audio before feeding them in our model. The results are highlighted in Table 4.

3. GPU Memory cost of the proposed approach

We note that our proposed approach utilizes additional GPU memory compared to standard ViTs. To demonstrate

Table 4. Comparison between various approaches for Audio classification

Model name	Audio classification		
	FSC22 [1]	ESC50 [11]	DCASE19 [6]
ViT-B-16	84.1	81.5	85.1
ViT-B-32	82.4	80.4	82.2
ViT-L-16	83.6	81.8	83.9
ViT-L-32	82.7	81.2	83.1
ViT-H-14	83.8	82.9	84.6
ViT-B-16-UCB	86.4	83.4	86.7
ViT-B-32-UCB	83.2	82.7	83.6
ViT-L-16-UCB	84.8	83.9	86
ViT-L-32-UCB	83.8	82.9	82.9
ViT-H-14-UCB	85.2	83.7	86.6

this, we report [Batch size / GPU Memory Usage (MB)]. From Table 5 we see that with increasing batch size we are seeing an increase in GPU memory. The memory cost increases with higher p values in the $top - p$ parameter as can be seen in Table 6.

Table 5. GPU memory requirement for different batch size while keeping the $top - p$ parameter fixed at 5. OOM denotes of out-of-memory problem.

Batch size	32	64	128	256	512
GPU Memory Usage	10867	18209	32379	65301	OOM

Table 6. GPU memory requirement for different p values in the $top - p$ parameter. The batch size has been fixed at 32

Top-p	1	4	5	6	10
GPU Memory Usage	9583	10823	10967	12041	13431

References

[1] Meelan Bandara, Roshinie Jayasundara, Isuru Ariyarathne, Dulani Meedeniya, and Charith Perera. Fsc22 dataset, 2022. 1, 2

[2] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the International Conference on Computer Vision (ICCV)*, 2021. 1

[3] M. Cimpoi, S. Maji, I. Kokkinos, S. Mohamed, , and A. Vedaldi. Describing textures in the wild. In *Proceedings of the IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2014. 1

[4] Adam Coates, Andrew Ng, and Honglak Lee. An analysis of single-layer networks in unsupervised feature learning. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 215–223. JMLR Workshop and Conference Proceedings, 2011. 1

[5] Le et al. Imagenet-200. 1

[6] Le et al. Imagenet-200. 1, 2

[7] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 1

[8] Chunyuan Li, Jianwei Yang, Pengchuan Zhang, Mei Gao, Bin Xiao, Xiyang Dai, Lu Yuan, and Jianfeng Gao. Efficient self-supervised vision transformers for representation learning. *International Conference on Learning Representations (ICLR)*, 2022. 1

[9] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian conference on computer vision, graphics & image processing*, pages 722–729. IEEE, 2008. 1

[10] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *2012 IEEE conference on computer vision and pattern recognition*, pages 3498–3505. IEEE, 2012. 1

[11] Karol J Piczak. Esc: Dataset for environmental sound classification. In *Proceedings of the 23rd ACM international conference on Multimedia*, pages 1015–1018, 2015. 1, 2

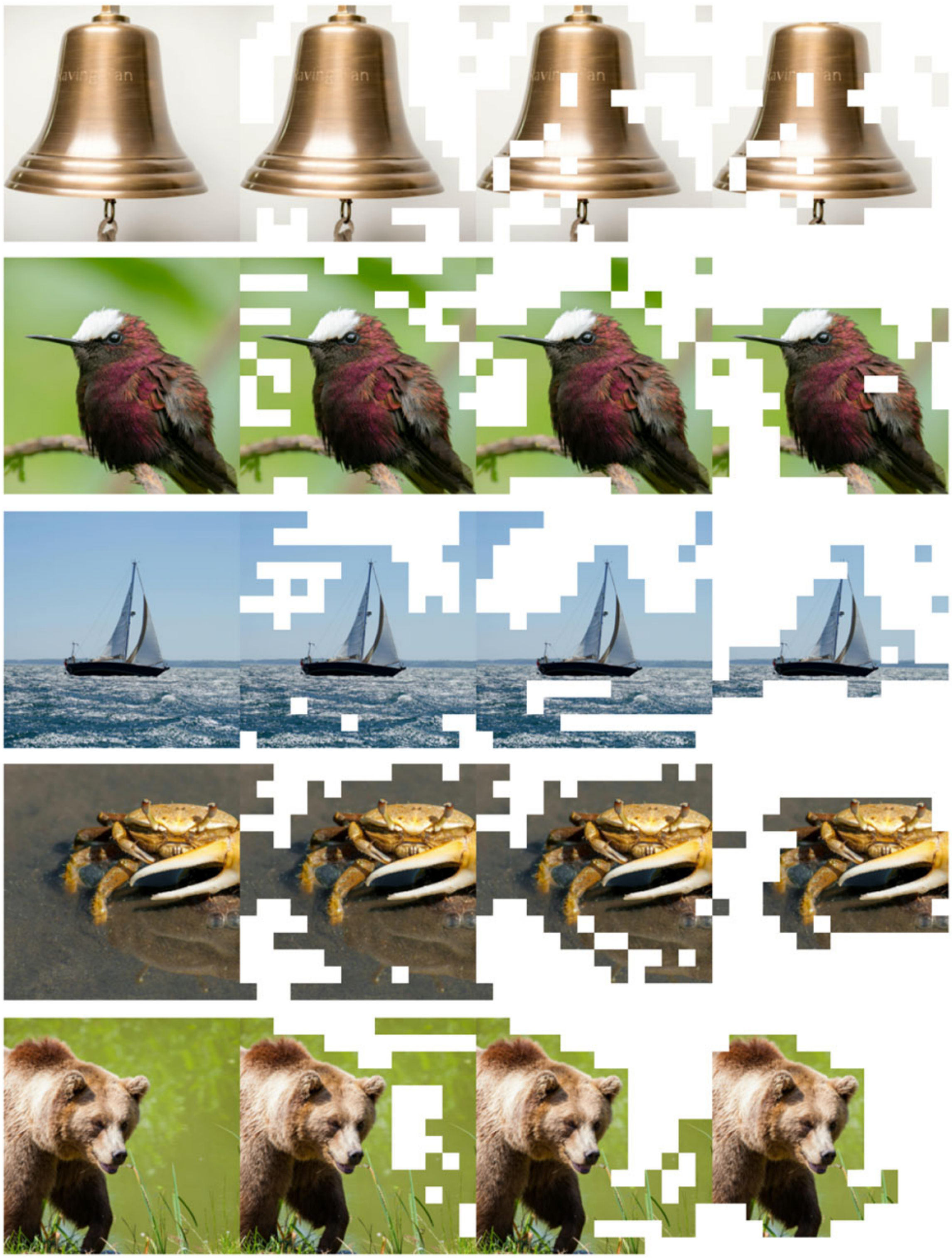

Figure 1. Visualization of Images after dropping redundant patches