Florence: A New Foundation Model for Computer Vision

(Lu Yuan 1 Dongdong Chen[1] Yi-Ling Chen[1] Noel Codella[1] Xiyang Dai[1] Jianfeng Gao[2] Houdong Hu[1] Xuedong Huang[1] Boxin Li[1] Chunyuan Li[2] Ce Liu[1] Mengchen Liu[1] Zicheng Liu[1] Yumao Lu[1] Yu Shi[1] Lijuan Wang[1] Jianfeng Wang[1] Bin Xiao[1] Zhen Xiao[1] Jianwei Yang[2] Michael Zeng[1] Luowei Zhou[1] Pengchuan Zhang[2])

Presenter : Kyungjin Cho

Mail : kjcho.amc@gmail.com

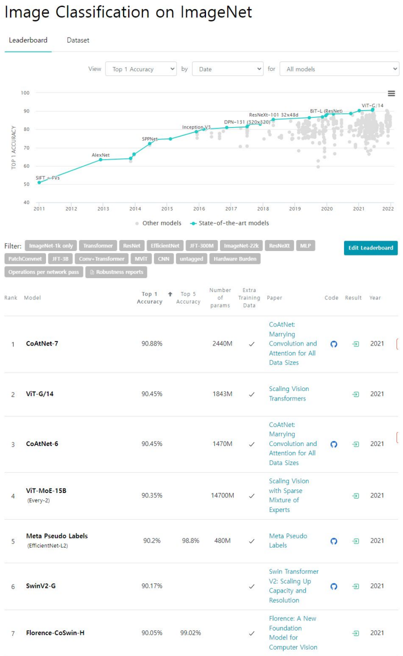
2022.01.12

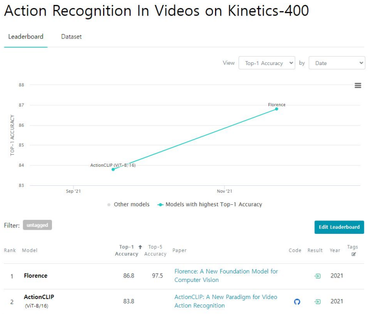
https://arxiv.org/abs/2111.11432

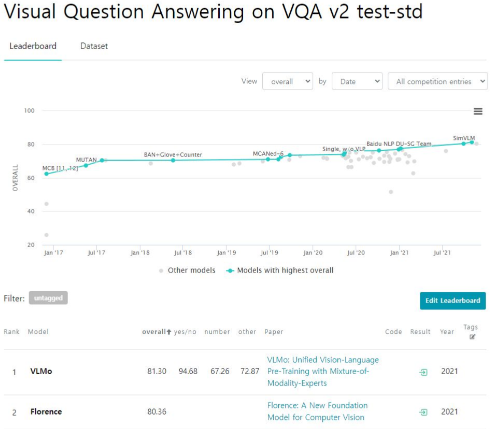
Contents

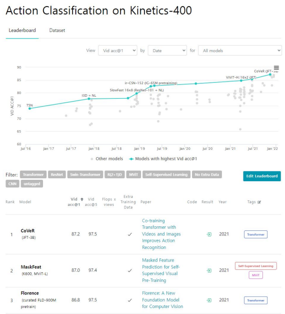
❖ Papers with code

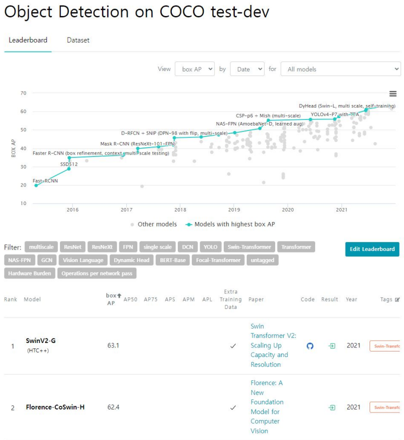
Florence achieves new state-of-the-art results in majority of 44 representative benchmarks, ~







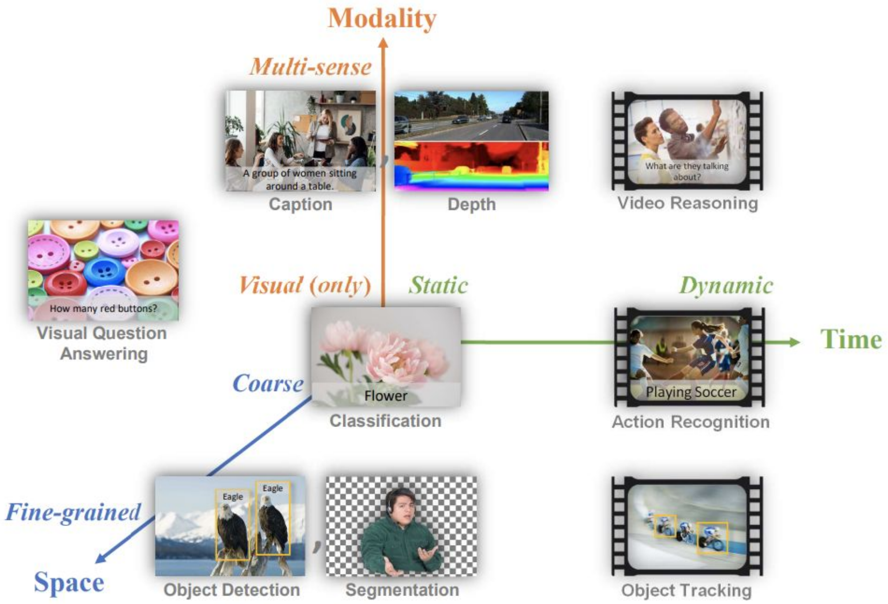




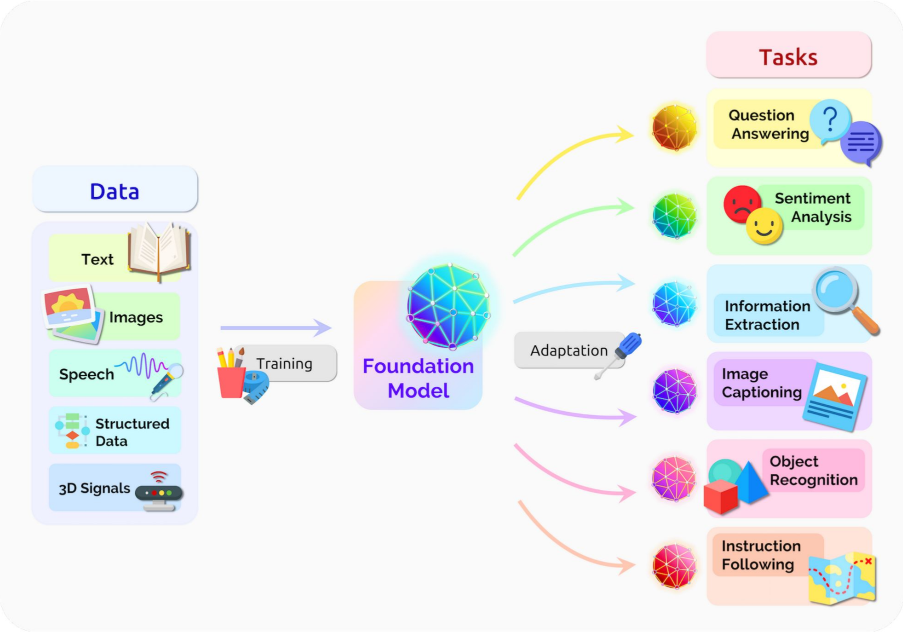
https://paperswithcode.com/

❖ Preliminary brief

- New CV foundation model
- Expand the representations from
  - coarse(scene) to fine (object)
  - static (images) to dynamic (videos)
  - RGB to multiple modalities (caption, depth, chest X-ray)
- Incorporates universal visual-language representations from Web-scale image-text data
- Can be easily adapted for various computer vision tasks
  - CLS, OD, VQA, Retrieval,,



❖ Visual foundation model



The term of foundation model was first introduced in (Bommasani et al., 2021) to refer to any model that is trained from **broad data at scale** that is capable of being **adapted** (e.g. fine-tuned) to **a wide range of downstream tasks**.

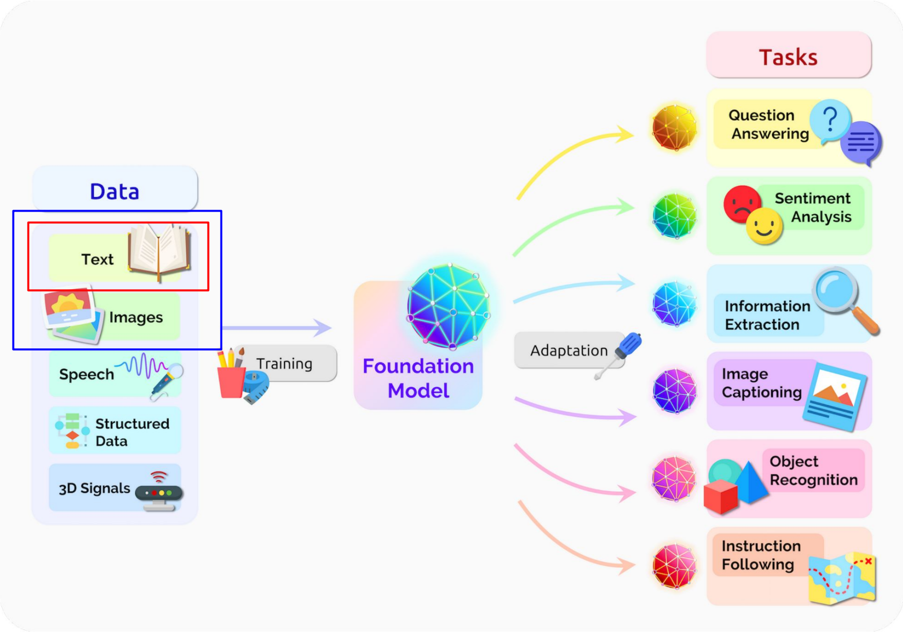
https://arxiv.org/pdf/2108.07258.pdf

❖ Visual foundation model

BERT, GPT3

CLIP, ALIGN, Wu Dao 2.0

- Efficient transfer learning
- Zero-shot capability.



https://arxiv.org/abs/1810.04805, https://arxiv.org/abs/2005.14165, https://arxiv.org/abs/2103.00020, https://arxiv.org/abs/2102.05918, https://gpt3demo.com/apps/wu-dao-20

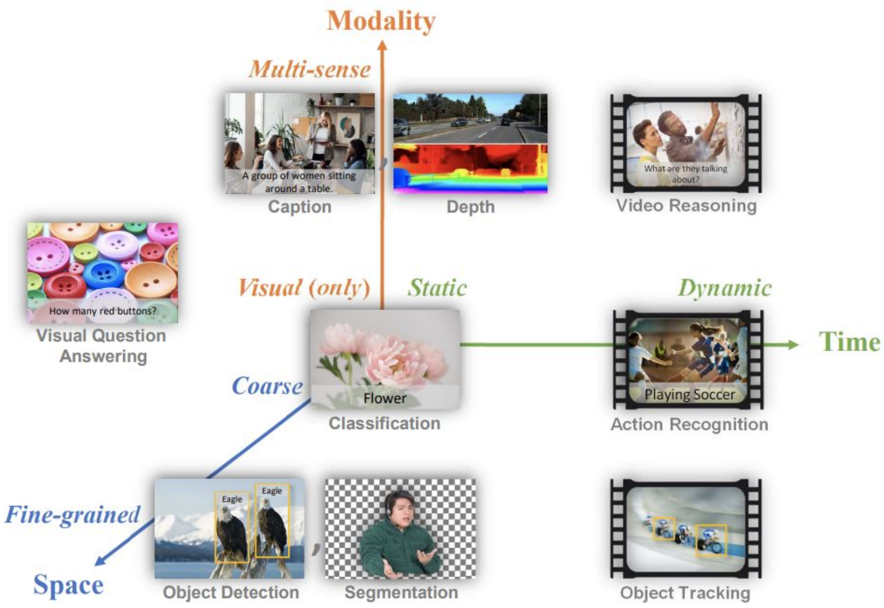
❖ Visual foundation model

"What is the foundation model for computer vision?"

1) Space : from coarse (e.g. scene-level classification) to fine-grained (e.g. object detection)

2) Time: from static (e.g. images) to dynamic (e.g. videos)

3) Modality: from RGB only to multiple senses (e.g. captioning and depth)



Foundation models for computer vision to be a pre-trained model and its adapters for solving all vision tasks in this ***Space-Time Modality*** *space.*

❖ Ecosystem of constructing Visual foundation models

- Data curation
  - Diverse, large-scale data (million) is the lifeblood of foundation models

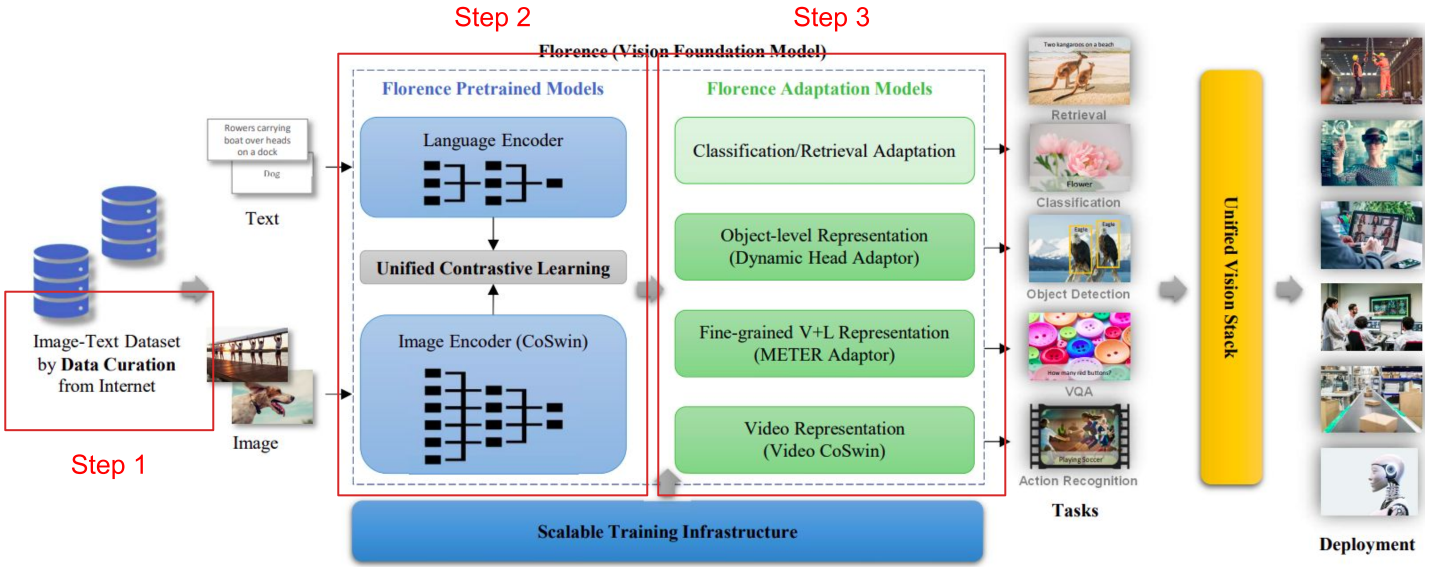
- Model pretraining
  - two-tower architecture including an image encoder and a language encoder.
    ex) CLIP, ALIGN, (Contrastive learning)
    image encoder: Swin, CvT, Vision Longformer, Focal Transformer, and CSwin (Vision transformer)

- Task adaptations
  - *extensible* and *transferable*
    1. space (from scene to objects) using the dynamic head adapter (Self attention method)
    2. time (from static image to videos) via proposed video CoSwin adapter (Vision transformer)
    3. modality (from images to language) via METER adapter (Masked image modeling)

- Training infrastructure
  - ZeRO, activation checkpointing, mixed-precision training, gradient cache (Training method)

https://arxiv.org/abs/2103.00020, https://arxiv.org/abs/2102.05918, https://arxiv.org/abs/2103.14030, https://arxiv.org/abs/2103.15808, https://arxiv.org/abs/2103.15358, https://arxiv.org/abs/2107.00641, https://arxiv.org/abs/2107.00652, https://arxiv.org/pdf/2106.08322, https://arxiv.org/abs/2111.02387, https://arxiv.org/abs/1910.02054, https://arxiv.org/abs/2101.06983,

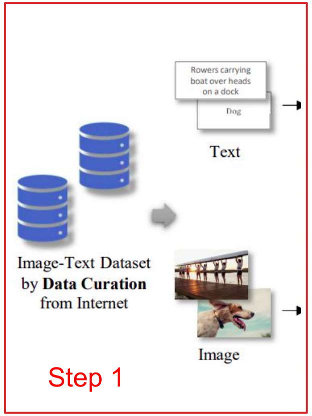
❖ Florence



Step 2
Step 3
Florence (Vision Foundation Model)
Florence Pretrained Models
Florence Adaptation Models
Rowers carrying boat over heads on a dock
Dog
Text
Language Encoder
Unified Contrastive Learning
Image Encoder (CoSwin)
Image-Text Dataset by Data Curation from Internet
Image
Step 1
Classification/Retrieval Adaptation
Object-level Representation (Dynamic Head Adaptor)
Fine-grained V+L Representation (METER Adaptor)
Video Representation (Video CoSwin)
Scalable Training Infrastructure
Two kangaroos on a beach
Retrieval
Flower
Classification
Eagle
Object Detection
VQA
Playing Soccer
Action Recognition
Tasks
Unified Vision Stack
Deployment

❖ Data curation

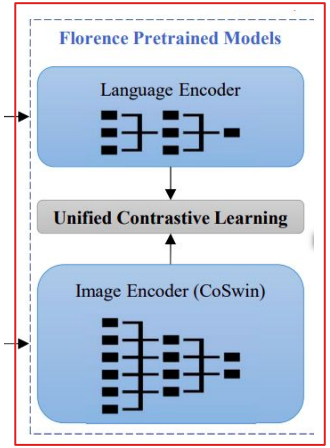
- Construct a FLD-900M from 3 billion Internet images and their descriptions.
- Respecting legal and ethical constraints
- Short caption, non-english, overlapped image, similar image, redundant image, redundant text, etc,,
- 9.7M unique queries(1 caption, many corresponding images)



❖ Unified Image-Text Contrastive learning

- CLIP implicitly assumes that each image-text pair has its unique caption, which allows other captions to be considered negative examples.
- However, in web-scale data, multiple images can be associated with identical captions.
- Image-text triplet (x, t, y) x : image t : language description, y : language label
- All image-text pairs mapped to the same label y are regarded as positive in our universal image-text contrastive learning.
- Our empirical experiments indicate that **long language descriptions** with rich content would be more beneficial for image-text representation learning than **short descriptions** (e.g. , one or two words). “A photo of the [WORD]”, “A cropped photo of [WORD]”

Step 2



❖ Unified Image-Text Contrastive learning

$$\boxed{u = \frac{f_\theta(x)}{\|f_\theta(x)\|} \quad v = \frac{f_\phi(x)}{\|f_\phi(x)\|}}$$

$$\mathcal{L} = \mathcal{L}_{i2t} + \mathcal{L}_{t2i}.$$

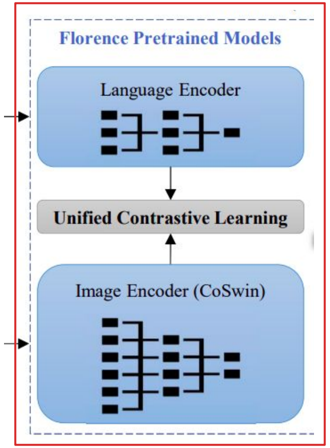
dot product

$$\mathcal{L}_{i2t} = -\sum_{i \in \mathcal{B}} \frac{1}{|\mathcal{P}(i)|} \sum_{k \in \mathcal{P}(i)} \log \frac{\exp(\tau \boldsymbol{u}_i \boldsymbol{v}_k)}{\sum_{j \in \mathcal{B}} \exp(\tau \boldsymbol{u}_i \boldsymbol{v}_j)}, \text{where } k \in \mathcal{P}(i) = \{k | k \in \mathcal{B}, y_k = y_i\}$$

$$\mathcal{L}_{t2i} = -\sum_{j \in \mathcal{B}} \frac{1}{|\mathcal{Q}(j)|} \sum_{k \in \mathcal{Q}(j)} \log \frac{\exp(\tau \boldsymbol{u}_k \boldsymbol{v}_j)}{\sum_{i \in \mathcal{B}} \exp(\tau \boldsymbol{u}_i \boldsymbol{v}_j)}, \text{where } k \in \mathcal{Q}(j) = \{k | k \in \mathcal{B}, y_k = y_j\}$$

- image encoder : f_θ , text encoder : f_ϕ
  normalized visual feature vector: u , normalized language feature vector: v
- Adam optimizer, weight decay, model parameter (893M)
- image size : 224 × 224 (80K iterations, fine-tuning 384 × 384),
  maximum language description length : 76
- Batch size 24,576 (takes 10 days to train on 512 NVIDIA-A100 GPUs.)
- CoSwin transformer was used.
- Two linear projection layers are added the dimensions of image and language features.

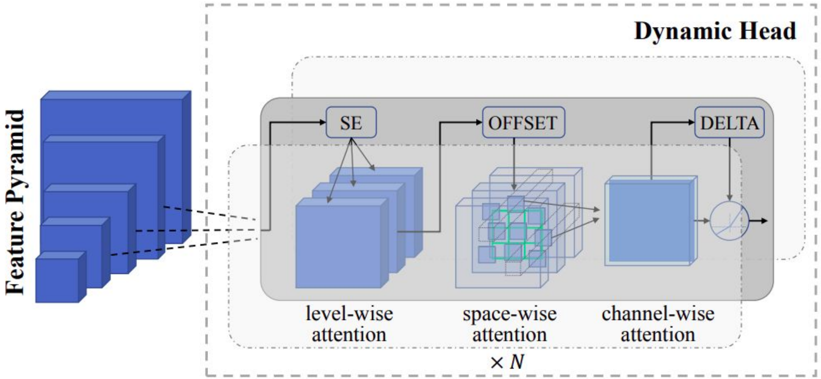
Step 2



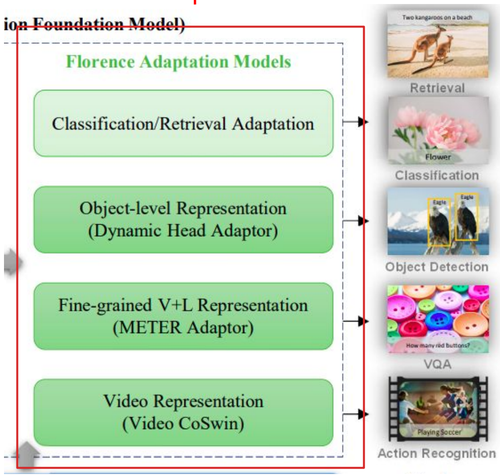
❖ Adaptation models (Object-level Visual Representation Learning)

Dynamic Head

- Coarse(scene) to fine (object) *level* × *space* × *channel*
- FLOD-9M (for FLorence Object detection Dataset)
  - LVIS, OpenImages, Object365, ImageNet-22K (with pseudo-labeling)
- Batch size 128 (takes 7 days)



Step 3

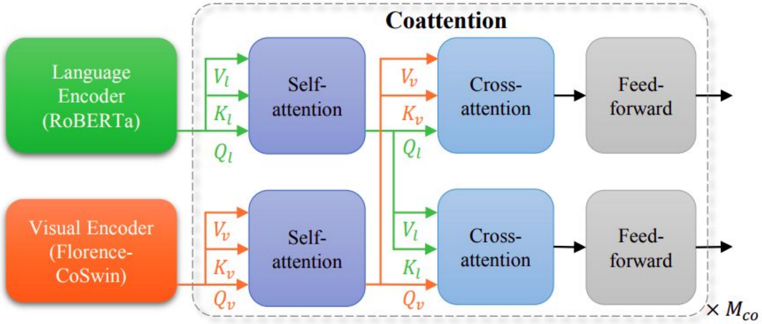


https://arxiv.org/pdf/2106.08322.pdf

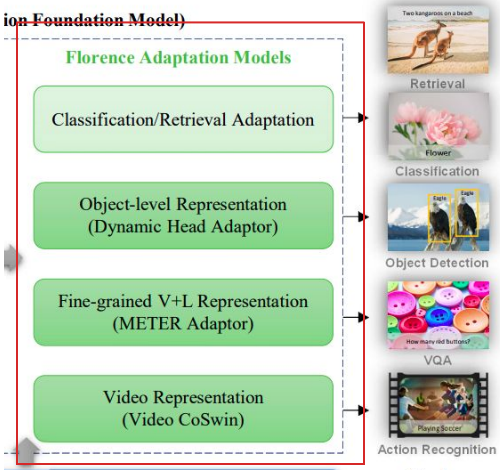
❖ Adaptation models (Fine-Grained V+L Representation Learning)

METER

- Visual question answering, image captioning, fine-grained representation
- Language Encoder (RoBERTa)
- Image-text matching loss and masked-language modeling loss



Step 3

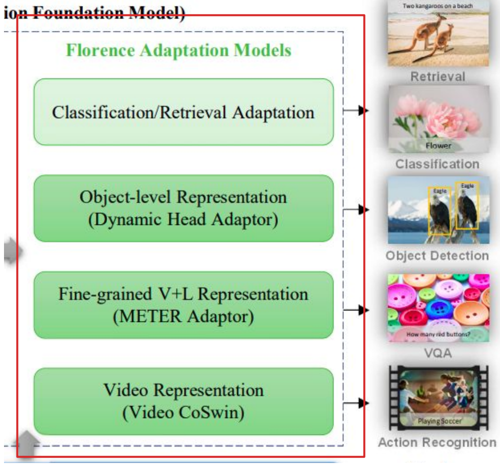


❖ Adaptation models (Adaption to Video Recognition)

Video CoSwin

- 2D conv → 3D conv
- 3D convolution-based patch merging operator
- 2D shifted window design with 3D shifted local windows in self-attention layers

Step 3



❖ Image classification

Florence: A New Foundation Model for Computer Vision

	Food101	CIFAR10	CIFAR100	SUN397	Stanford Cars	FGVC Aircraft	VOC2007	DTD	Oxford Pets	Caltech101	Flowers102	ImageNet
CLIP-ResNet-50x64	91.8	86.8	61.3	48.9	76.0	35.6	83.8	53.4	93.4	90.6	77.3	73.6
CLIP-ViT-L/14 (@336pix)	93.8	**95.7**	77.5	68.4	78.8	37.2	84.3	55.7	93.5	92.8	78.3	76.2
FLIP-ViT-L/14	92.2	**95.7**	75.3	73.1	70.8	**60.2**	-	60.7	92.0	93.0	**90.1**	78.3
Florence-CoSwin-H (@384pix)	**95.1**	94.6	**77.6**	**77.0**	**93.2**	55.5	**85.5**	**66.4**	**95.9**	**94.7**	86.2	**83.7**

Table 1. Zero-shot transfer of image classification comparisons on 12 datasets: CLIP-ResNet-50x64 (Radford et al., 2021), FLIP-ViT-L/14 (Yao et al., 2021).

Florence: A New Foundation Model for Computer Vision

	Food101	CIFAR10	CIFAR100	SUN397	Stanford Cars	FGVC Aircraft	VOC2007	DTD	Oxford Pets	Caltech101	Flowers102
SimCLRv2-ResNet-152x3	83.6	96.8	84.5	69.1	68.5	63.1	86.7	80.5	92.6	94.9	96.3
ViT-L/16 (@384pix)	87.4	97.9	89.0	74.9	62.5	52.2	86.1	75.0	92.9	94.7	99.3
EfficientNet-L2 (@800pix)	92.0	**98.7**	**89.0**	75.7	75.5	68.4	89.4	82.5	95.6	94.7	97.9
CLIP-ResNet-50x64	94.8	94.1	78.6	81.1	90.5	67.7	88.9	82.0	94.5	95.4	98.9
CLIP-ViT-L/14 (@336pix)	95.9	97.9	87.4	82.2	91.5	71.6	89.9	83.0	95.1	96.0	99.2
Florence-CoSwin-H (@384pix)	**96.2**	97.6	87.1	**84.2**	**95.7**	**83.9**	**90.5**	**86.0**	**96.4**	**96.6**	**99.7**

Table 2. Comparisons of image classification linear probing on 11 datasets with existing state-of-the-art models, including SimCLRv2 (Chen et al., 2020c), ViT (Dosovitskiy et al., 2021a), EfficientNet (Xie et al., 2020), and CLIP (Radford et al., 2021).

Model	Params	Data	Accuracy	
			Top-1	Top-5
BiT-L-ResNet152x4	928M	300M	87.54	98.46
ALIGN-Efficient-L2	480M	1800M	88.64	98.67
ViT-G/14	1843M	3000M	90.45	-
CoAtNet-7	2440M	3000M	**90.88**	-
Florence-CoSwin-H	637M	900M	90.05	**99.02**

Table 3. Classification fine tuning on ImageNet-1K. Florence is compared with: BiT-L-ResNet152x4 (Kolesnikov et al., 2020), ALIGN-Efficient-L2 (Jia et al., 2021), ViT-G/14 (Zhai et al., 2021), CoAtNet-7 (Dai et al., 2021c) in terms of model scale, data scale and Top-1/Top-5 accuracy.

❖ Few-shot Cross-domain Classification

	Method	Flickr30K (1K test set)				MSCOCO (5K test set)			
		Image → Text		Text → Image		Image → Text		Text → Image	
		R@1	R@5	R@1	R@5	R@1	R@5	R@1	R@5
Zero-shot	ImageBERT (Qi et al., 2020)	70.7	90.2	54.3	79.6	44.0	71.2	32.3	59.0
	UNITER (Chen et al., 2020d)	83.6	95.7	68.7	89.2	-	-	-	-
	CLIP (Radford et al., 2021)	88.0	98.7	68.7	90.6	58.4	81.5	37.8	62.4
	ALIGN (Jia et al., 2021)	88.6	98.7	75.7	**93.8**	58.6	83.0	45.6	69.8
	FLIP (Yao et al., 2021)	89.8	**99.2**	75.0	93.4	61.3	84.3	45.9	70.6
	Florence	**90.9**	**99.1**	**76.7**	**93.6**	**64.7**	**85.9**	**47.2**	**71.4**
Fine-tuned	GPO (Chen et al., 2020a)	88.7	98.9	76.1	94.5	68.1	90.2	52.7	80.2
	UNITER (Chen et al., 2020d)	87.3	98.0	75.6	94.1	65.7	88.6	52.9	79.9
	ERNIE-ViL (Yu et al., 2020)	88.1	98.0	76.7	93.6	-	-	-	-
	VILLA (Gan et al., 2020)	87.9	97.5	76.3	94.2	-	-	-	-
	Oscar (Li et al., 2020)	-	-	-	-	73.5	92.2	57.5	82.8
	ALIGN (Jia et al., 2021)	95.3	99.8	84.9	97.4	77.0	93.5	59.9	83.3
	FLIP (Yao et al., 2021)	96.6	**100.0**	87.1	97.7	78.9	94.4	61.2	84.3
	Florence	**97.2**	**99.9**	**87.9**	**98.1**	**81.8**	**95.2**	**63.2**	**85.7**

Table 5. Image-text retrieval comparisons on Flickr30K and MSCOCO datasets (zero-shot and fine-tuned).

ISIC EuroSAT CropDisease ChestX

	Model	ISIC	EuroSAT	CropD	ChestX	mean
5-shot	CW	57.4	88.1	96.6	29.7	68.0
	Florence	57.1	90.0	97.7	29.3	**68.5**
20-shot	CW	68.1	94.7	99.2	38.3	75.1
	Florence	72.9	95.8	99.3	37.5	**76.4**
50-shot	CW	74.1	96.9	99.7	44.4	78.8
	Florence	78.3	97.1	99.6	42.8	**79.5**

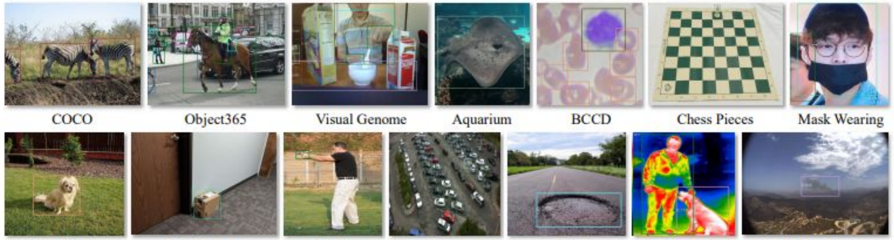
Table 4. Comparison with CW (Liu et al., 2020) (CD-FSL Challenge 2020 Winner) on CD-FSL benchmark. The average result comparison is 74.8 (Florence) vs. 73.9 (CW).

For zero-shot retrieval, we feed the input text (or image) to the language (or image) encoder of Florence to get the feature embeddings, and also compute the feature embeddings of the set of possible images (or texts) by the image (or language) encoder.

❖ Object Detection and Zero-shot Transfer

Benchmark	Model	AP
COCO miniVal	DyHead	60.3
	Soft Teacher	60.7
	Florence	<u>**62.0**</u>
COCO test-Dev	DyHead	60.6
	Soft Teacher	61.3
	Florence	<u>**62.4**</u>
Object365	Multi-dataset Detection	33.7
	Florence	<u>**39.3**</u>
Visual Genome	VinVL	13.8
	Florence	<u>**16.2**</u>

Table 6. Object detection fine tuning comparisons with state-of-the-art methods, including DyHead (Dai et al., 2021a), Soft Teacher (Xu et al., 2021b), Multi-dataset Detection (Zhou et al., 2021), VinVL (Zhang et al., 2021b).



Florence: A New Foundation Model for Computer Vision

		Aquarium	BCCD	Chess Pieces	Mask Wearing	Oxford Pets	Packages	Pistols	PKLot	Pothole	Thermal	Wildfire Smoke
	Images	638	364	292	149	3680	26	2986	12416	665	203	737
	Categories	7	3	12	2	37	1	1	2	1	2	1
Fine-tuned	DyHead-Swin-L (full)	53.1	62.6	80.7	52.0	85.9	52.0	74.4	98.0	61.8	75.9	58.7
	DyHead-Swin-L (5-shot)	39.0	40.6	57.3	26.8	47.5	32.8	20.0	22.1	10.8	54.9	14.2
Zero-shot	ZSD	16.0	1.2	0.1	0.6	0.3	58.3	31.5	0.2	2.4	37.4	0.002
	Florence	43.1	15.3	13.4	15.0	68.9	79.6	41.4	31.4	53.3	46.9	48.7

Table 7. Zero-shot transfer in object detection, in comparison with previous state-of-the-art model DyHead (Dai et al., 2021a) (on COCO) fine tuning results on full-set or 5-shot respectively and zero-shot detection baseline model ZSD (Bansal et al., 2018).

❖ Image-text retrieval comparisons

	Method	Flickr30K (1K test set)				MSCOCO (5K test set)			
		Image → Text		Text → Image		Image → Text		Text → Image	
		R@1	R@5	R@1	R@5	R@1	R@5	R@1	R@5
Zero-shot	ImageBERT (Qi et al., 2020)	70.7	90.2	54.3	79.6	44.0	71.2	32.3	59.0
	UNITER (Chen et al., 2020d)	83.6	95.7	68.7	89.2	-	-	-	-
	CLIP (Radford et al., 2021)	88.0	98.7	68.7	90.6	58.4	81.5	37.8	62.4
	ALIGN (Jia et al., 2021)	88.6	98.7	75.7	**93.8**	58.6	83.0	45.6	69.8
	FLIP (Yao et al., 2021)	89.8	**99.2**	75.0	93.4	61.3	84.3	45.9	70.6
	Florence	**90.9**	**99.1**	**76.7**	**93.6**	**64.7**	**85.9**	**47.2**	**71.4**
Fine-tuned	GPO (Chen et al., 2020a)	88.7	98.9	76.1	94.5	68.1	90.2	52.7	80.2
	UNITER (Chen et al., 2020d)	87.3	98.0	75.6	94.1	65.7	88.6	52.9	79.9
	ERNIE-ViL (Yu et al., 2020)	88.1	98.0	76.7	93.6	-	-	-	-
	VILLA (Gan et al., 2020)	87.9	97.5	76.3	94.2	-	-	-	-
	Oscar (Li et al., 2020)	-	-	-	-	73.5	92.2	57.5	82.8
	ALIGN (Jia et al., 2021)	95.3	99.8	84.9	97.4	77.0	93.5	59.9	83.3
	FLIP (Yao et al., 2021)	96.6	**100.0**	87.1	97.7	78.9	94.4	61.2	84.3
	Florence	**97.2**	**99.9**	**87.9**	**98.1**	**81.8**	**95.2**	**63.2**	**85.7**

Table 5. Image-text retrieval comparisons on Flickr30K and MSCOCO datasets (zero-shot and fine-tuned).

For zero-shot retrieval, we feed the input text (or image) to the language (or image) encoder of Florence to get the feature embeddings, and also compute the feature embeddings of the set of possible images (or texts) by the image (or language) encoder.

❖ Take home message