Carnegie Mellon University
Language Technologies Institute

Vision, Robotics, Hardware…

Yonatan Bisk and Emma Strubell

Some slides stolen from 11-777

Benchmarking on Pi Zero

	Delta	Remark
USB Ethernet dongle	+104 mA	No network connection, only a USB Ethernet dongle inserted into the USB OTG port.
USB Ethernet dongle + link	+180 mA	After inserting an Ethernet cable.
USB Ethernet dongle + iperf	+244 mA	Average. See chart below for details.
2.4 GHz WiFi	+11 mA	Connected to 2.4 GHz access point.
2.4 GHz WiFi + iperf	+187 mA	Average. See chart below for details.
Logitech RF dongle for keyboard/mouse	+29 mA	
Seagate USB HDD (idle)	+255 mA	Stable current after initial insertion that peaked at 1.06A. See chart below for details.
Seagate USB HDD (iozone)	+554 mA	Average. See chart below for details
HDMI cable	+7 mA	This is only after inserting the cable, not about enabling/disabling HDMI. More on that below.

Processor clocks can also be modified!

https://www.cnx-software.com/2021/12/09/raspberry-pi-zero-2-w-power-consumption/

Operating Systems and Libraries

Ubuntu 18.04 and 20.04

Python 3.6+

PyTorch 1.09+

Torchvision 0.9.0

Huggingface transformers 4.9.2+

These are not required.
You might find several others very useful:
1. TFLite
2. ONNX
3. Detectron2
4. ROS

But they are not officially supported by the course so you will have limited TA support with debugging.

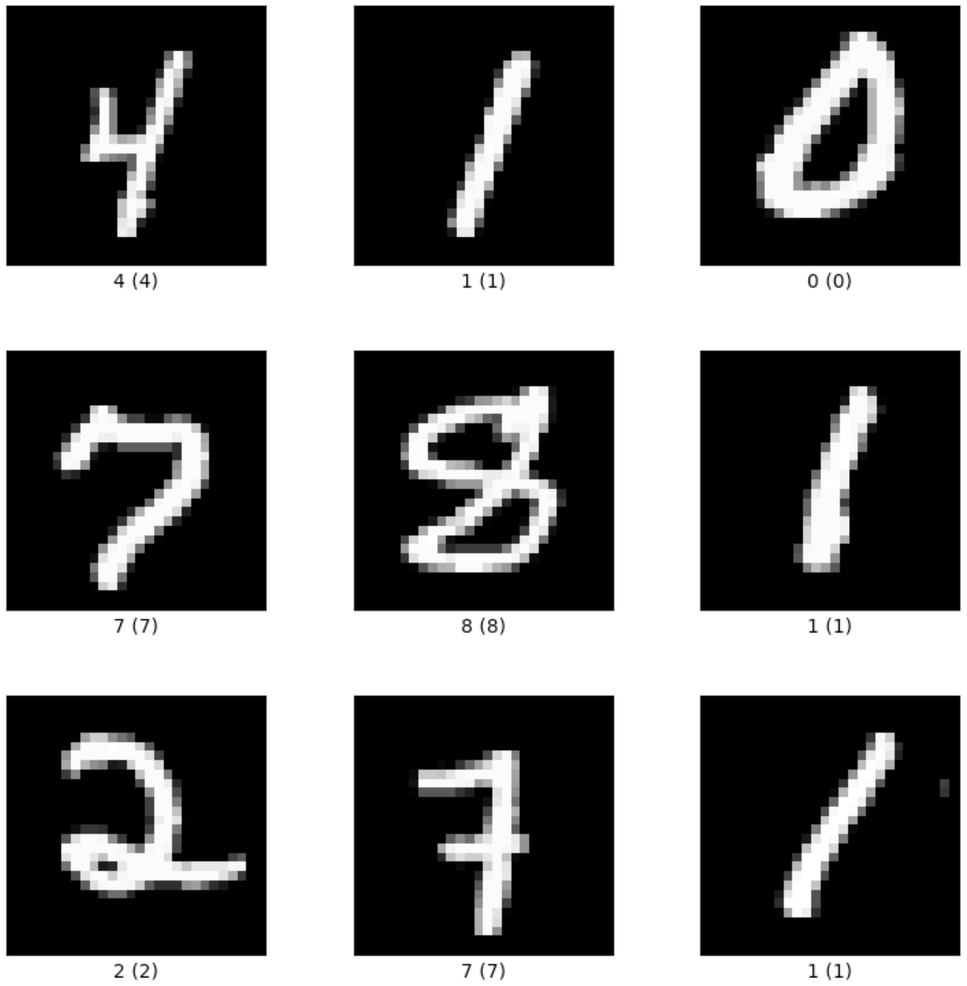
Keywords

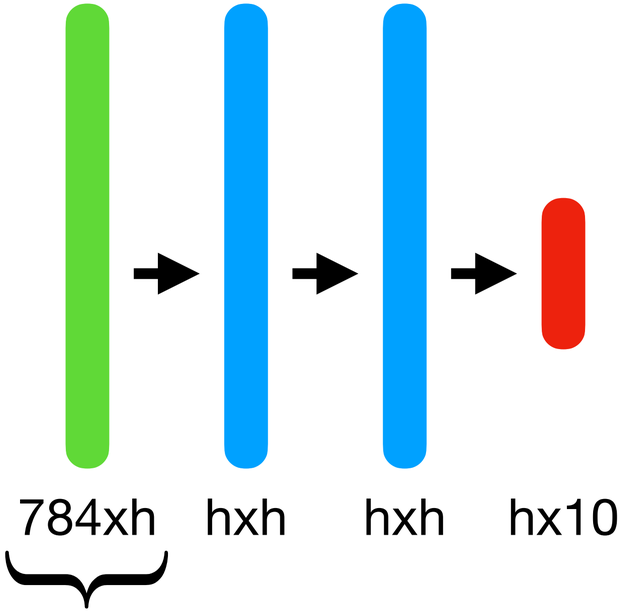
- Feature Representations
- Input/Output dimensionality
- CNNs: Layers (norms, pooling, etc)
  - 1D, 2D, 3D
- Transformers: (Self-Attention Blocks, Auxiliary Loss, Masking, Auto-regressive)
- Mapping, real valued control, …

Questions

- How much information does a pixel, a spectrogram, a word contain?
- How much do I care about parameter size vs model size?
- Does my task require a large vocabulary (why?), high resolution (why?), …
- What interactions are required in the input?
  (Which pixels or words need to talk to each other?)

Lab 1: Computer Vision

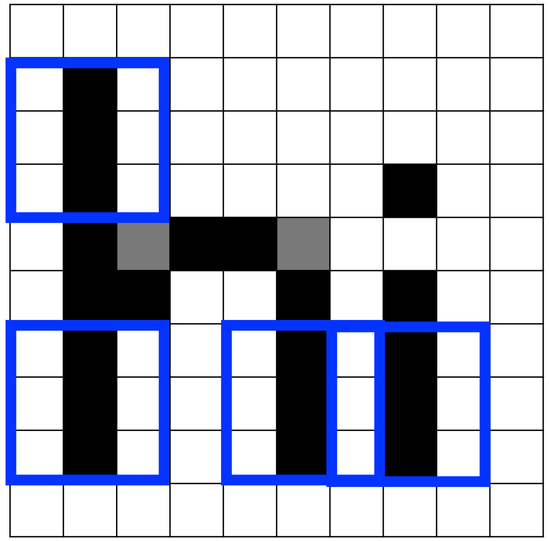




Each hidden dimension is a function of ***all*** input pixels

Let's make some assumptions:

Parameter Efficient, Local Processing



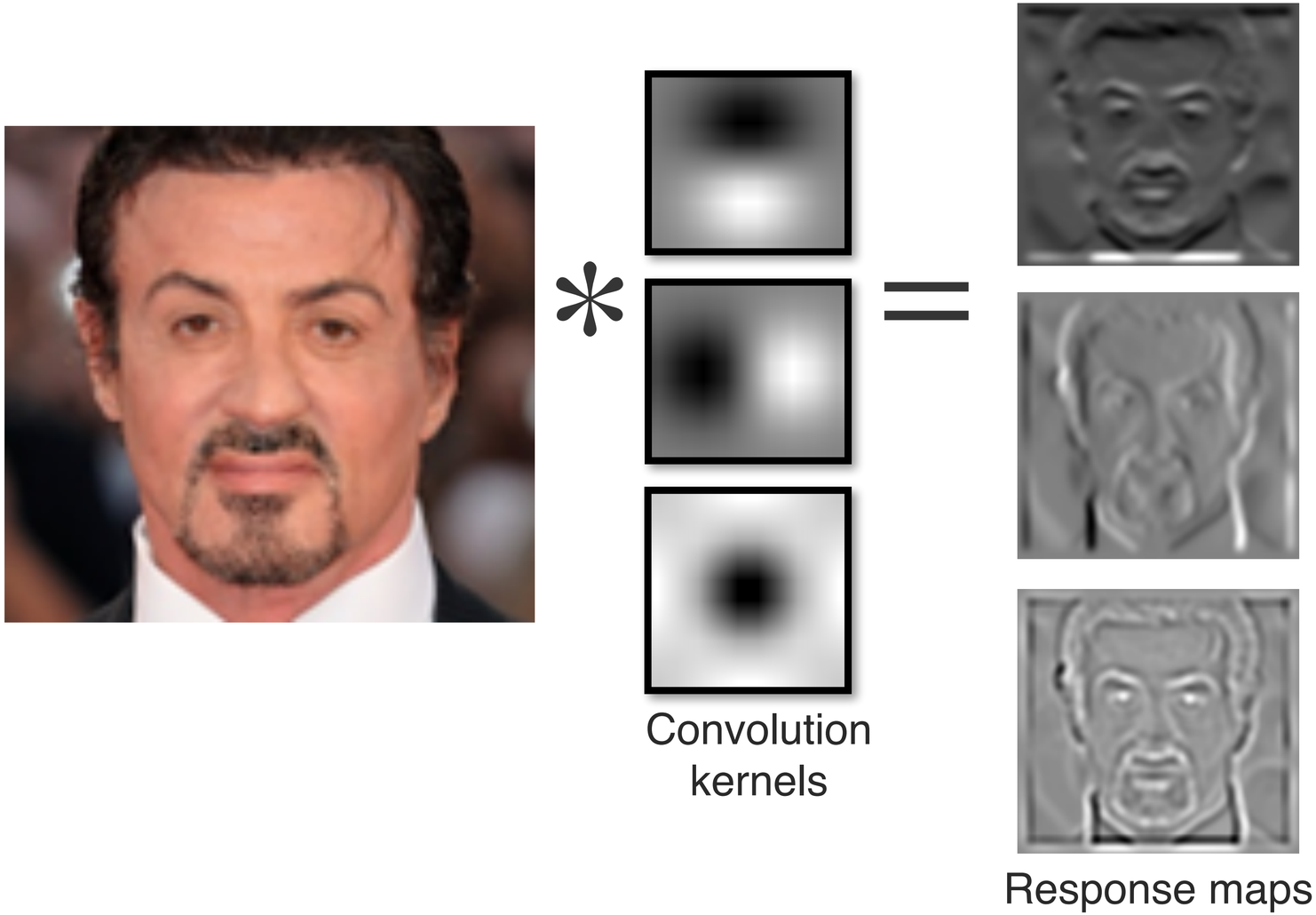
Learning a code book

Small, inexpensive, and reusable parameters

FF	10x10xh	Run once
CNN	3x3xh	Run 81x? 🤔
		Run 16x?

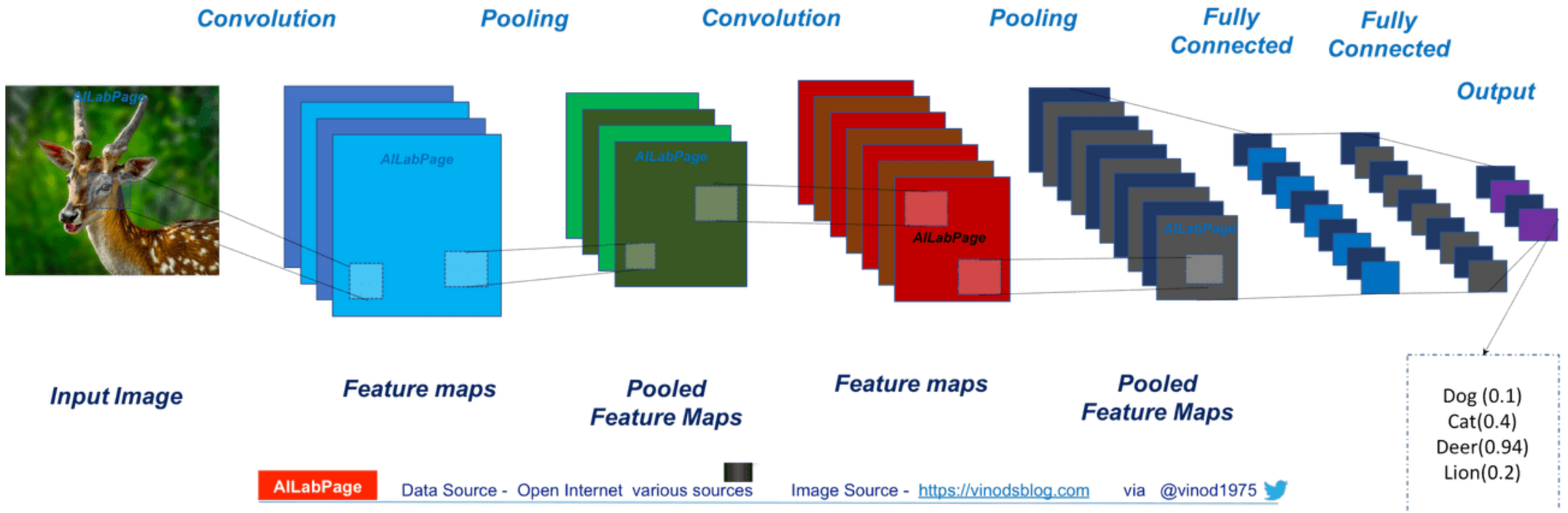
Learning efficiency — “lots of small images”

Convolution Kernels



Object Descriptors

Convolutional Neural Network (CNN)



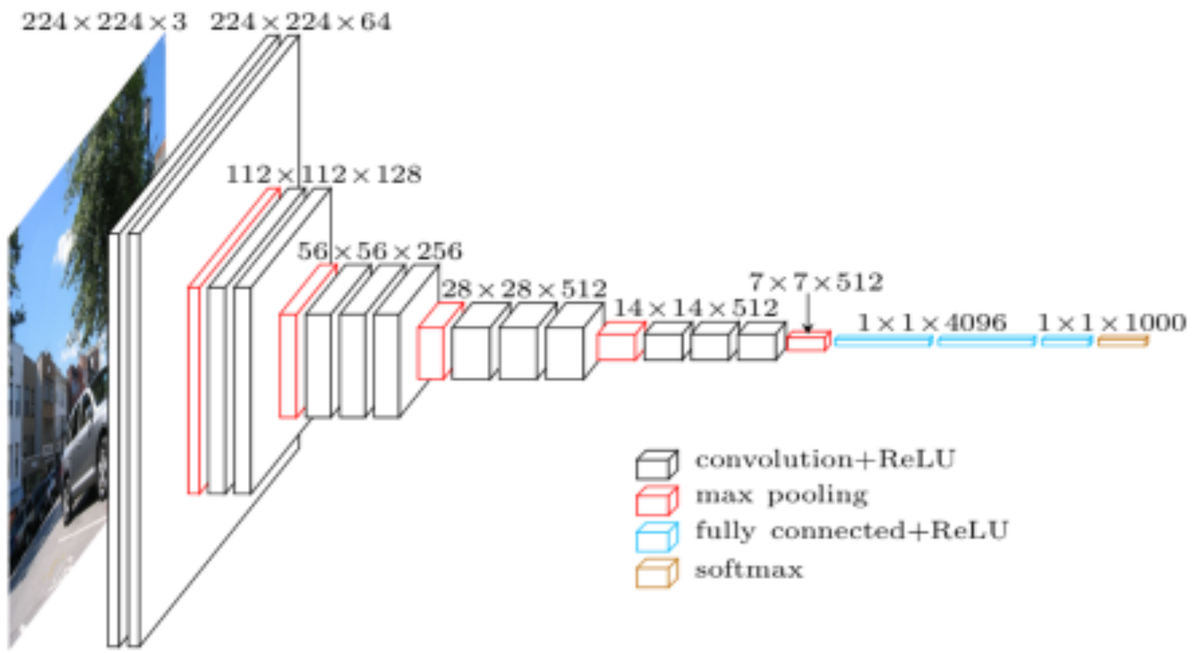
One representation, lots of tasks

https://github.com/facebookresearch/detectron2

VGGNet model

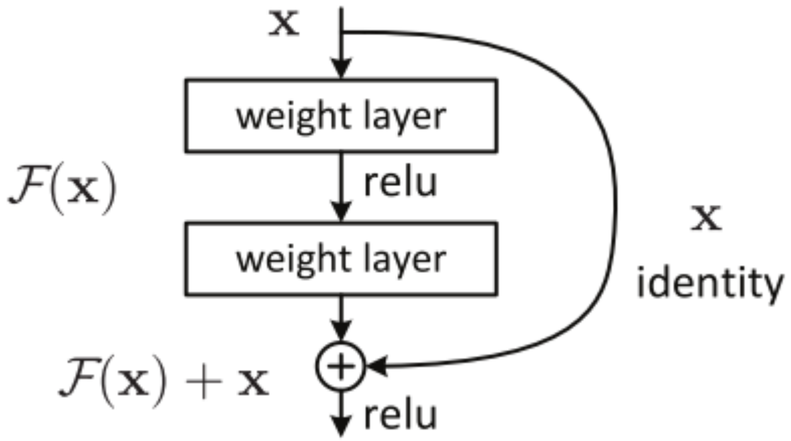
Used for object classification task

- 1000-way classification task
- 138 million parameters



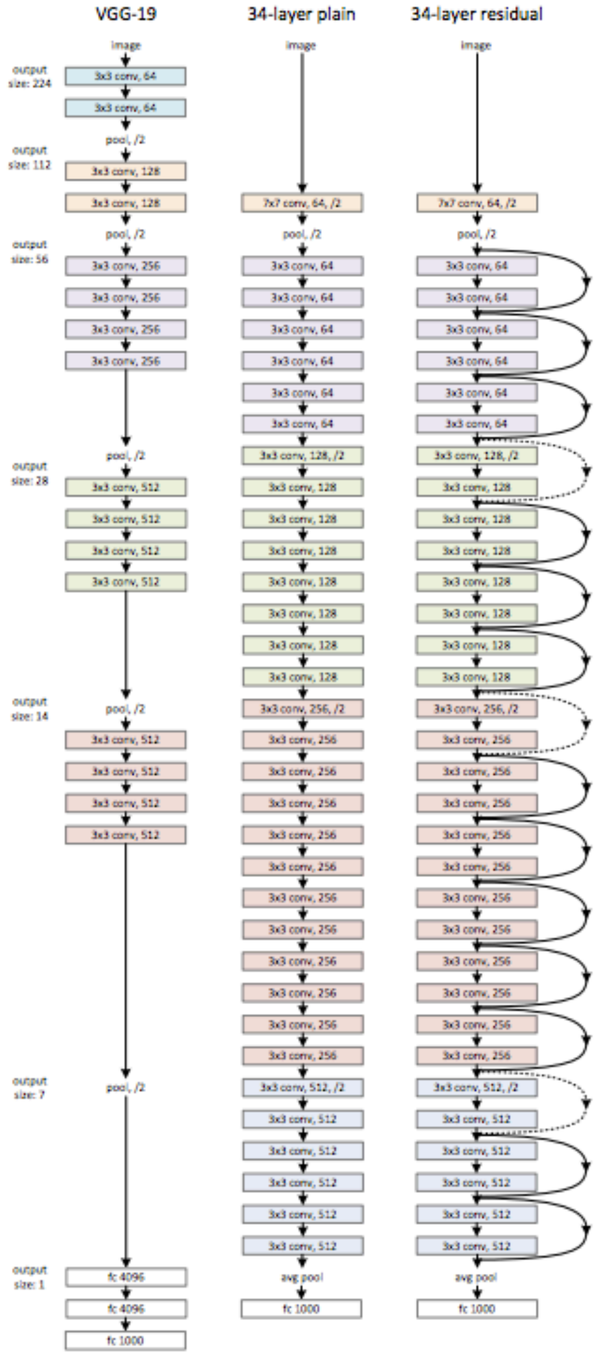
Residual Networks

Adding residual connections

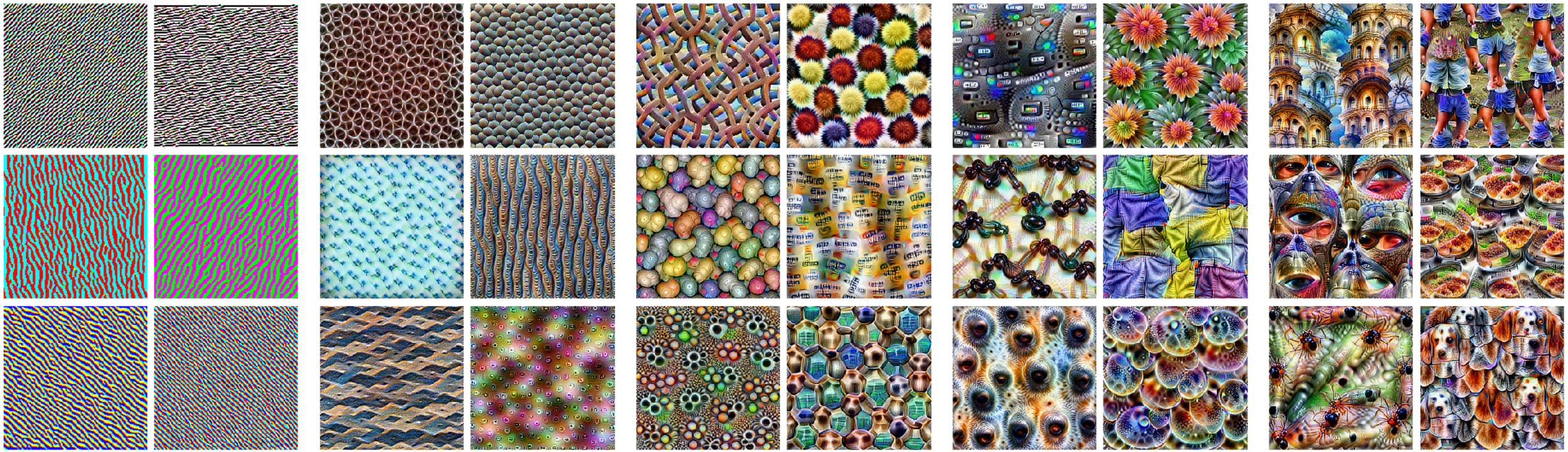


ResNet (He et al., 2015)

- Up to 152 layers!



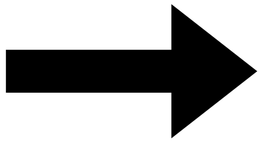
Hierarchical Concepts



Edges (layer conv2d0) **Textures** (layer mixed3a) **Patterns** (layer mixed4a) **Parts** (layers mixed4b & mixed4c) **Objects** (layers mixed4d & mixed4e)

https://distill.pub/2017/feature-visualization/

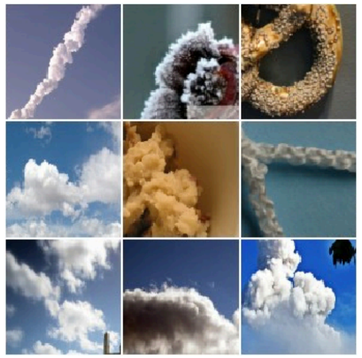
Discriminative Labels Select Features



“Dog”

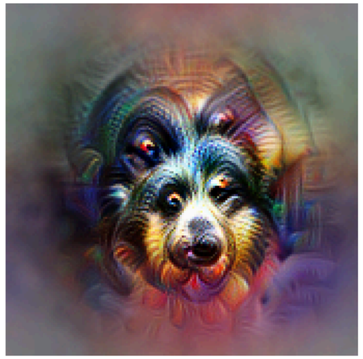
What will the model learn is the meaning of dog?

Dataset Examples show us what neurons respond to in practice

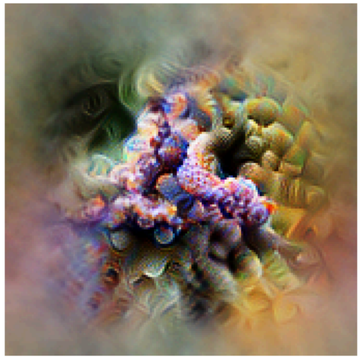


Optimization isolates the causes of behavior from mere correlations. A neuron may not be detecting what you initially thought.

Baseball—or stripes?
mixed4a, Unit 6



Animal faces—or snouts?
mixed4a, Unit 240



Clouds—or fluffiness?
mixed4a, Unit 453

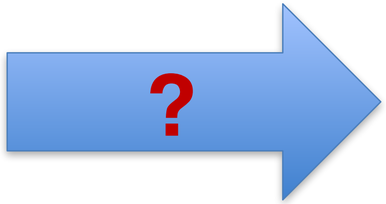
Buildings—or sky?
mixed4a, Unit 492

Object recognition

Input Image

Object Detection (and Segmentation)

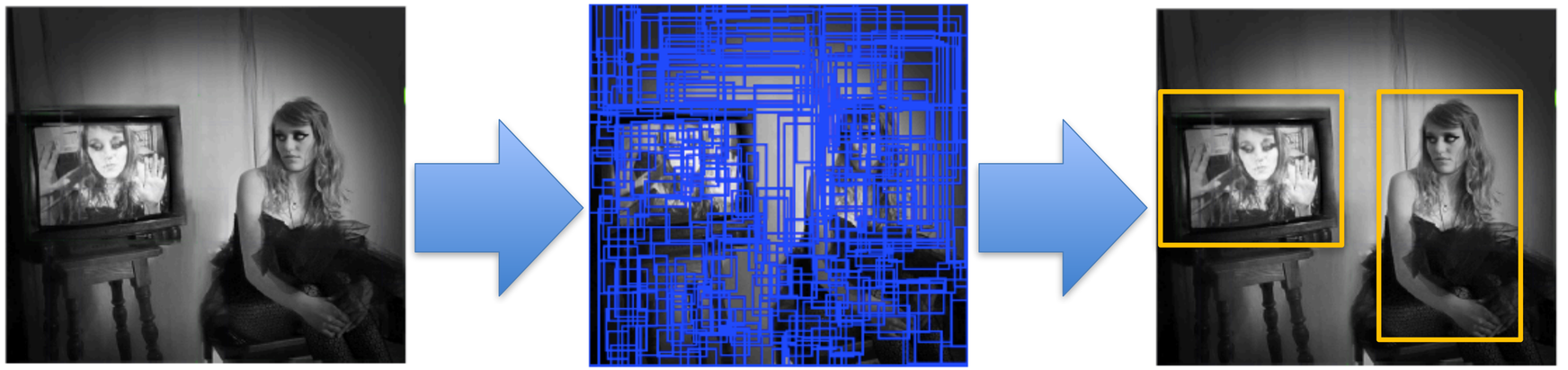
Input image



Detected Objects

One option: Sliding window

Object Detection (and Segmentation)

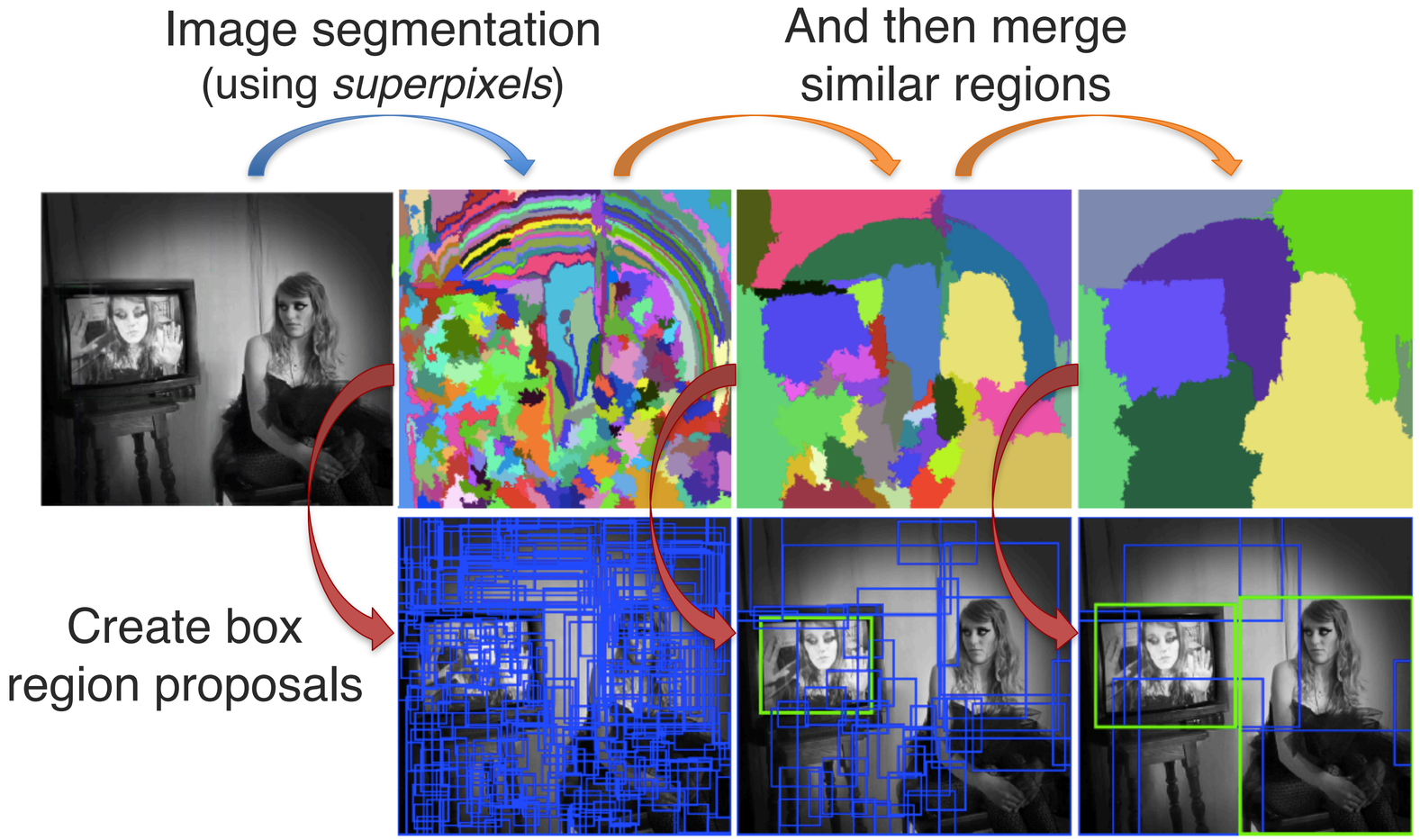


Input image Region Proposals Detected Objects

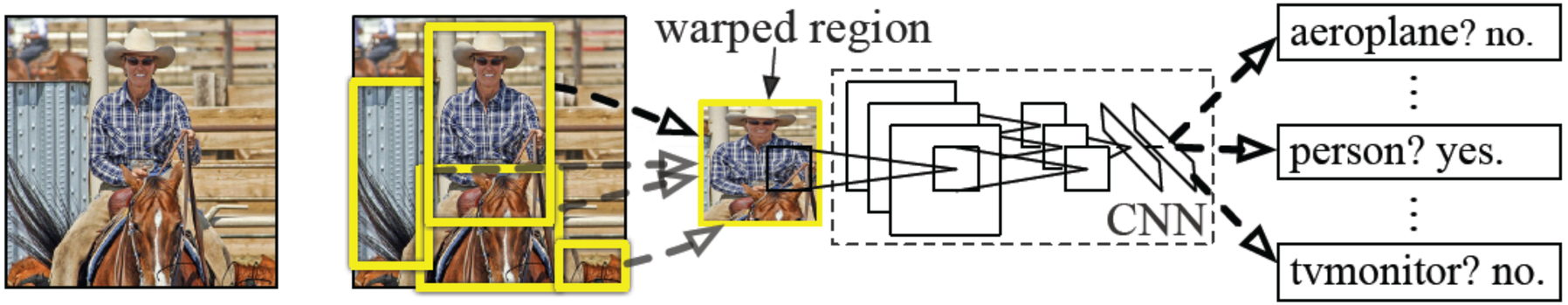
A better option: Start by Identifying hundreds of region proposals and then apply our CNN object detector

How to efficiently identify region proposals?

Selective Search [Uijlings et al., IJCV 2013]



R-CNN [Girshick et al., CVPR 2014]

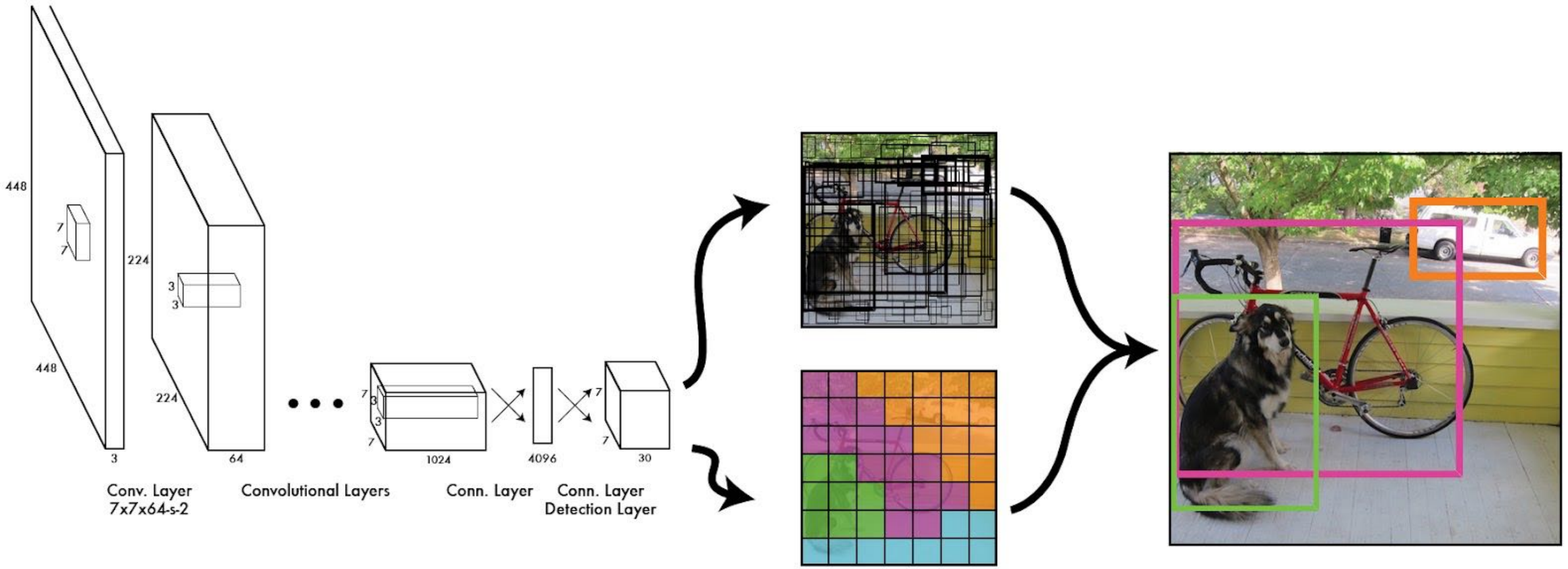


- Select ~2000 region proposals → Time consuming!
- Warp each region
- Apply CNN to each region → Time consuming!

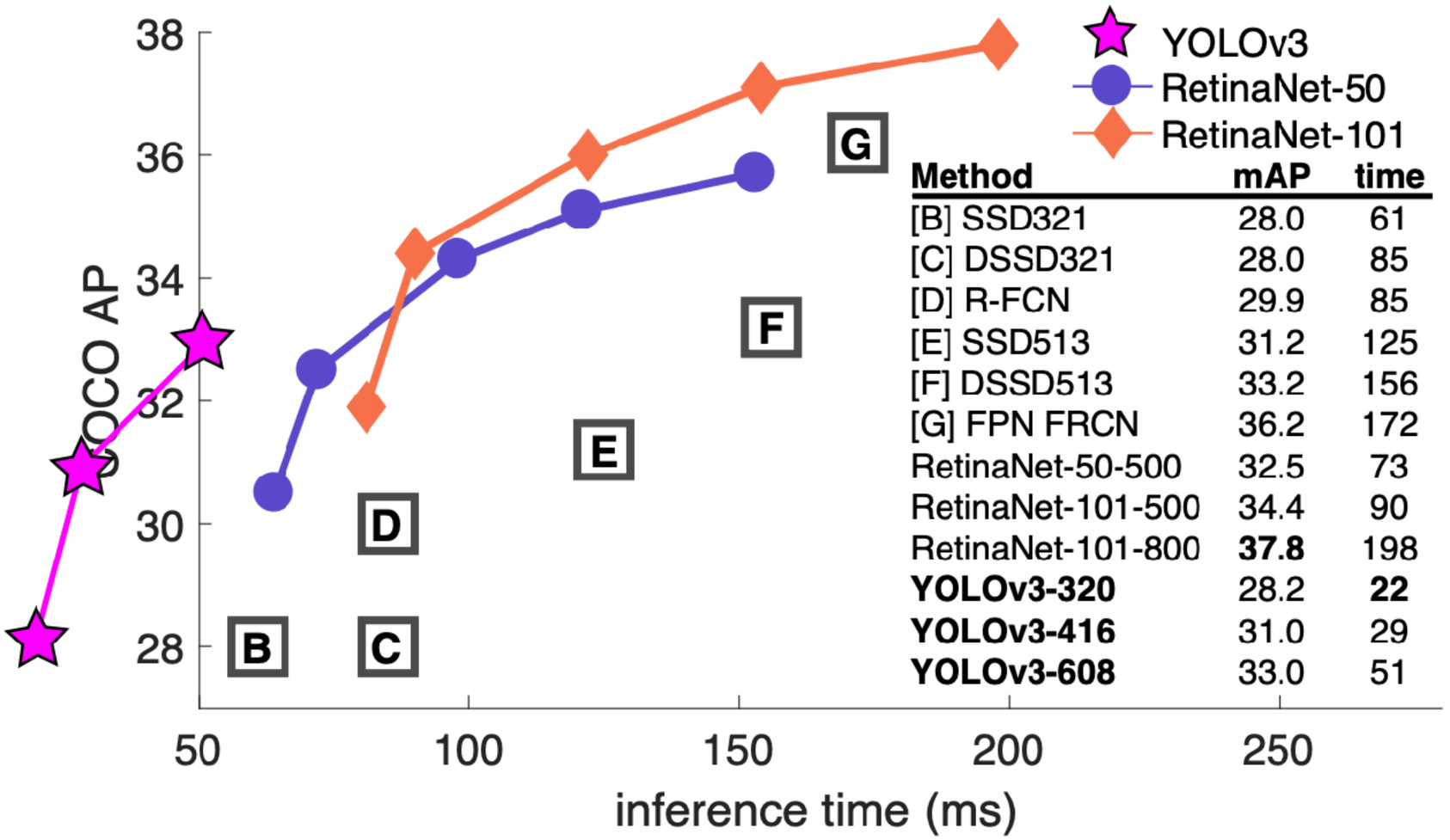
Fast R-CNN: Applies CNN only once, and then extracts regions

Faster R-CNN: Region selection on the Conv5 response map

YOLO [cvpr 2016]



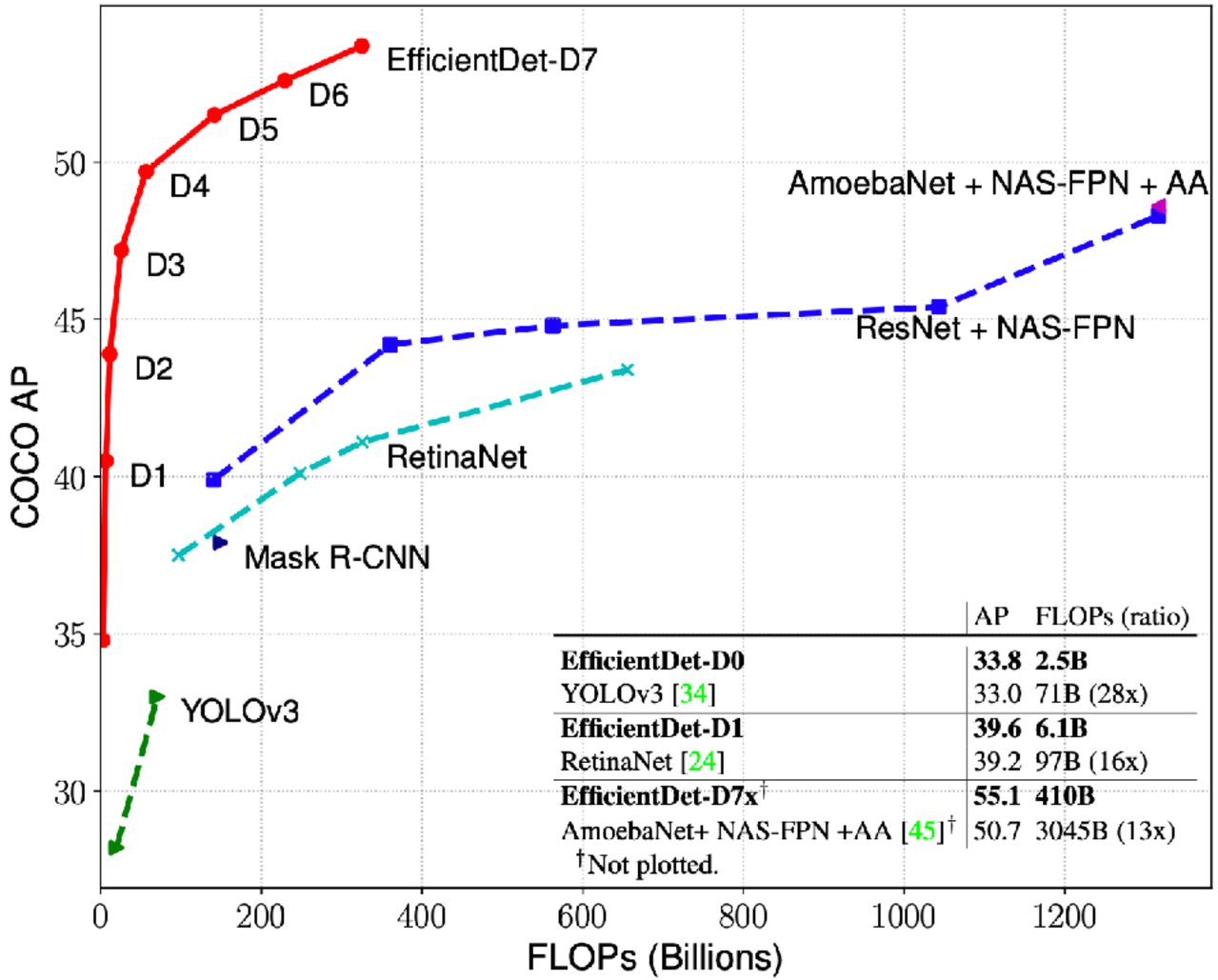
Trade-off Between Speed and Accuracy



Method	mAP	time
[B] SSD321	28.0	61
[C] DSSD321	28.0	85
[D] R-FCN	29.9	85
[E] SSD513	31.2	125
[F] DSSD513	33.2	156
[G] FPN FRCN	36.2	172
RetinaNet-50-500	32.5	73
RetinaNet-101-500	34.4	90
RetinaNet-101-800	**37.8**	198
YOLOv3-320	28.2	**22**
YOLOv3-416	31.0	29
YOLOv3-608	33.0	51

https://arxiv.org/abs/1804.02767

Trade-off Between Speed and Accuracy

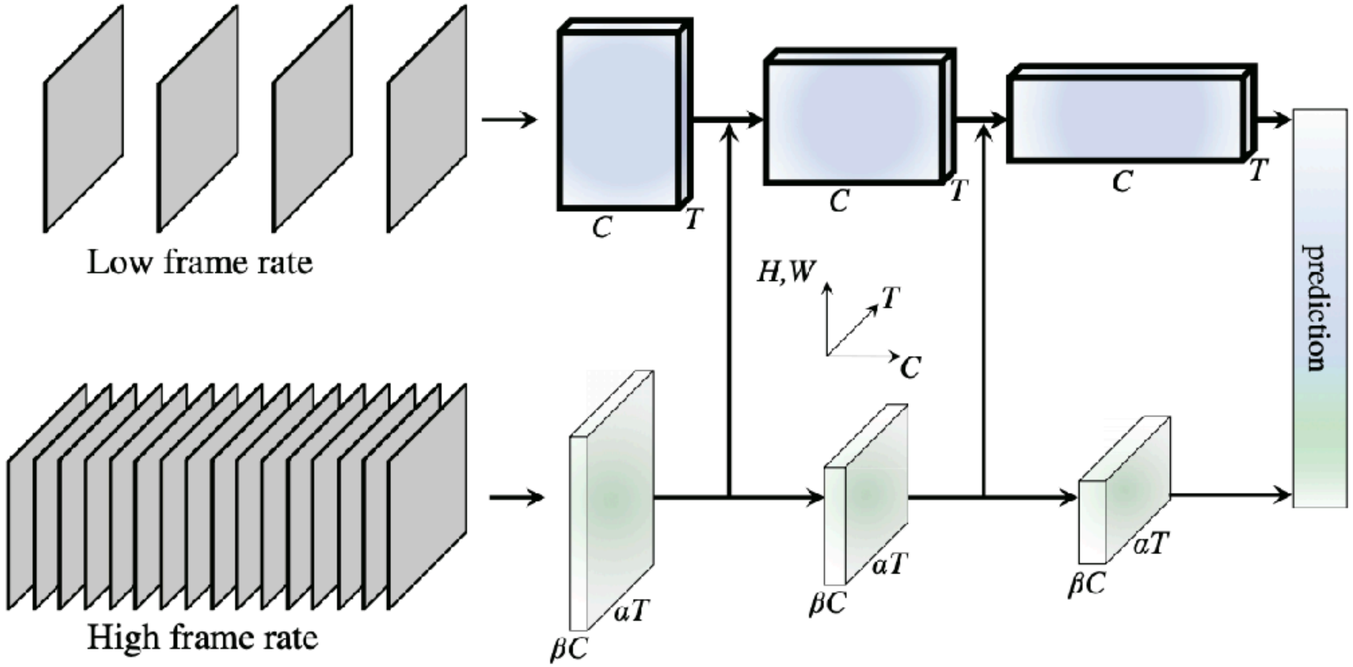


	AP	FLOPs (ratio)
EfficientDet-D0	**33.8**	**2.5B**
YOLOv3 [34]	33.0	71B (28x)
EfficientDet-D1	**39.6**	**6.1B**
RetinaNet [24]	39.2	97B (16x)
EfficientDet-D7x†	**55.1**	**410B**
AmoebaNet+ NAS-FPN +AA [45]†	50.7	3045B (13x)

†Not plotted.

https://arxiv.org/abs/1911.09070

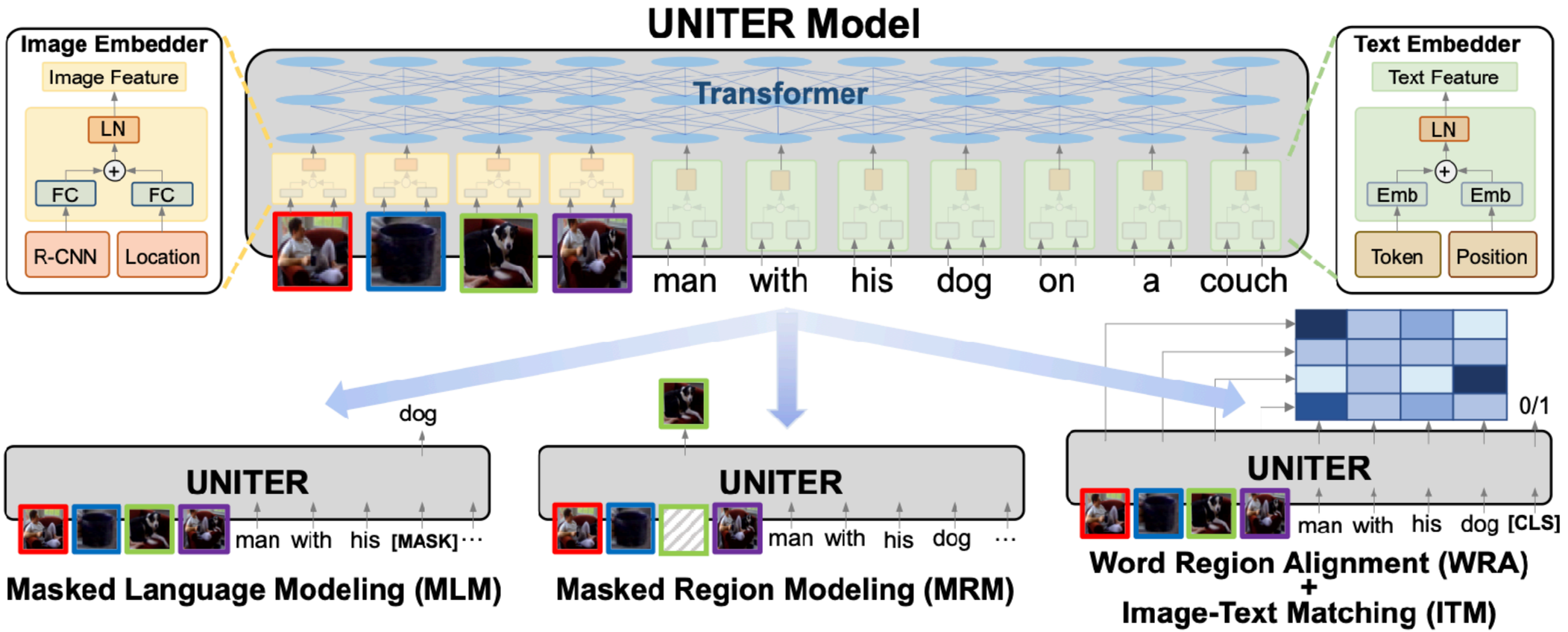
SlowFast Networks for Video Recognition



stage	*Slow* pathway	*Fast* pathway	output sizes $T \times S^2$
raw clip	-	-	64×224^2
data layer	stride 16, 1^2	stride **2**, 1^2	*Slow* : 4×224^2 *Fast* : 32×224^2
conv_1	1×7^2, 64 stride 1, 2^2	5×7^2, 8 stride 1, 2^2	*Slow* : 4×112^2 *Fast* : 32×112^2
pool_1	1×3^2 max stride 1, 2^2	1×3^2 max stride 1, 2^2	*Slow* : 4×56^2 *Fast* : 32×56^2
res_2	$\begin{bmatrix} 1 \times 1^2, 64 \\ 1 \times 3^2, 64 \\ 1 \times 1^2, 256 \end{bmatrix} \times 3$	$\begin{bmatrix} 3 \times 1^2, 8 \\ 1 \times 3^2, 8 \\ 1 \times 1^2, 32 \end{bmatrix} \times 3$	*Slow* : 4×56^2 *Fast* : 32×56^2
res_3	$\begin{bmatrix} 1 \times 1^2, 128 \\ 1 \times 3^2, 128 \\ 1 \times 1^2, 512 \end{bmatrix} \times 4$	$\begin{bmatrix} 3 \times 1^2, 16 \\ 1 \times 3^2, 16 \\ 1 \times 1^2, 64 \end{bmatrix} \times 4$	*Slow* : 4×28^2 *Fast* : 32×28^2
res_4	$\begin{bmatrix} 3 \times 1^2, 256 \\ 1 \times 3^2, 256 \\ 1 \times 1^2, 1024 \end{bmatrix} \times 6$	$\begin{bmatrix} 3 \times 1^2, 32 \\ 1 \times 3^2, 32 \\ 1 \times 1^2, 128 \end{bmatrix} \times 6$	*Slow* : 4×14^2 *Fast* : 32×14^2
res_5	$\begin{bmatrix} 3 \times 1^2, 512 \\ 1 \times 3^2, 512 \\ 1 \times 1^2, 2048 \end{bmatrix} \times 3$	$\begin{bmatrix} 3 \times 1^2, 64 \\ 1 \times 3^2, 64 \\ 1 \times 1^2, 256 \end{bmatrix} \times 3$	*Slow* : 4×7^2 *Fast* : 32×7^2
global average pool, concate, fc			# classes

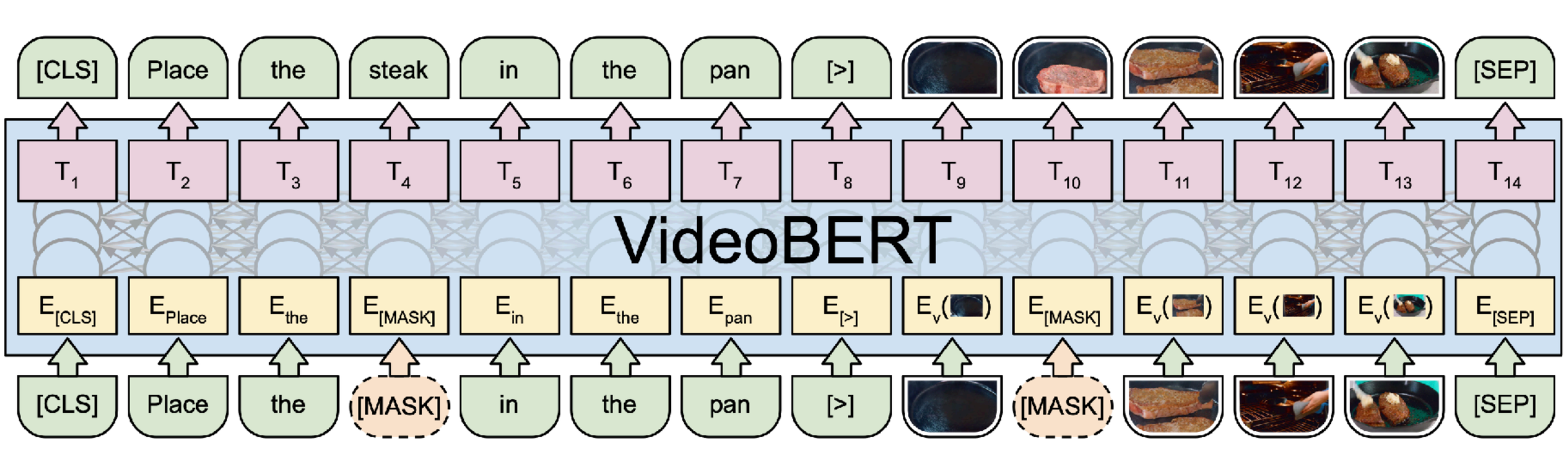
https://arxiv.org/abs/1812.03982

UNITER



https://arxiv.org/abs/1909.11740v1

VideoBERT

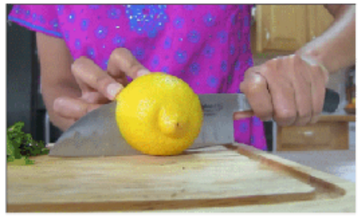


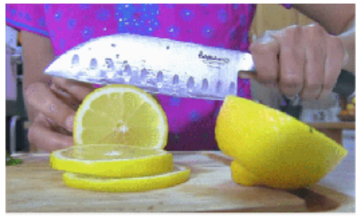
How do we get visual words now?!?

https://arxiv.org/abs/1904.01766

Clusters define visual “words” — 12^4 clusters

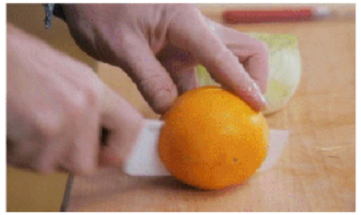
Original

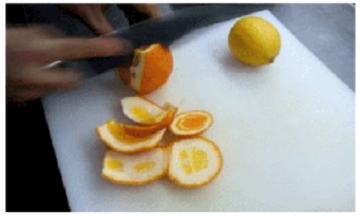




ASR:
“So it's really up to you cut it off get this and nice slices.”

Centroids





Top verbs: cut, prepare, make
Top nouns: orange, lemon, tomato

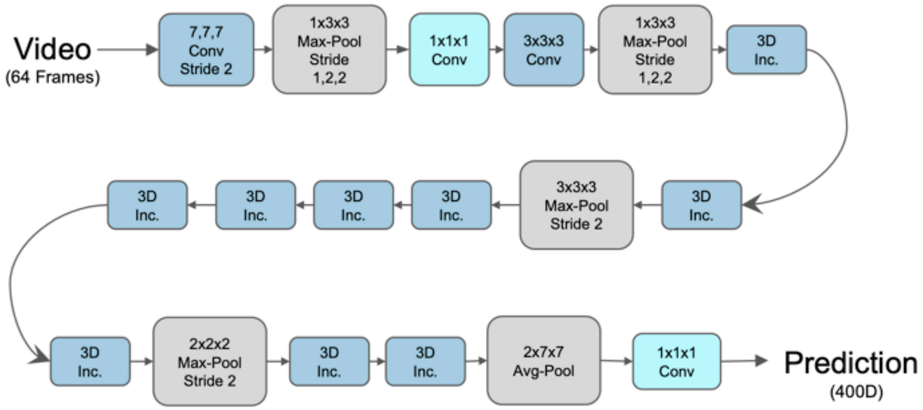
Input text	Retrieved centroid	Retrieved centroid
“Put the pizza into oven.”		
“Put the cookies into oven.”		

Video-to-Video

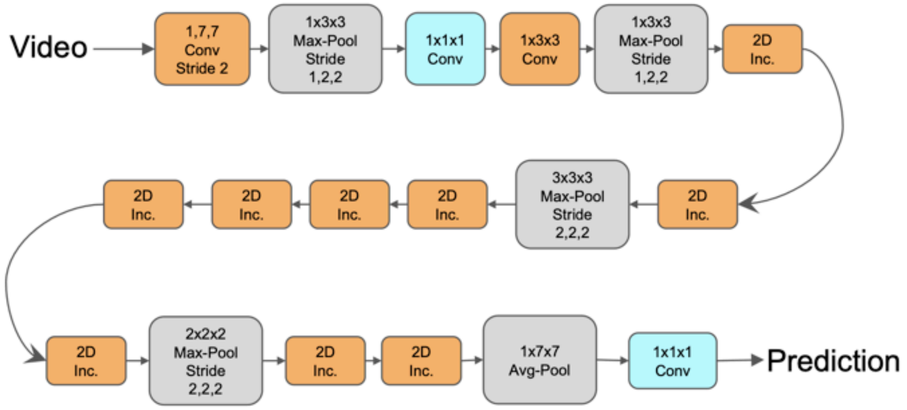




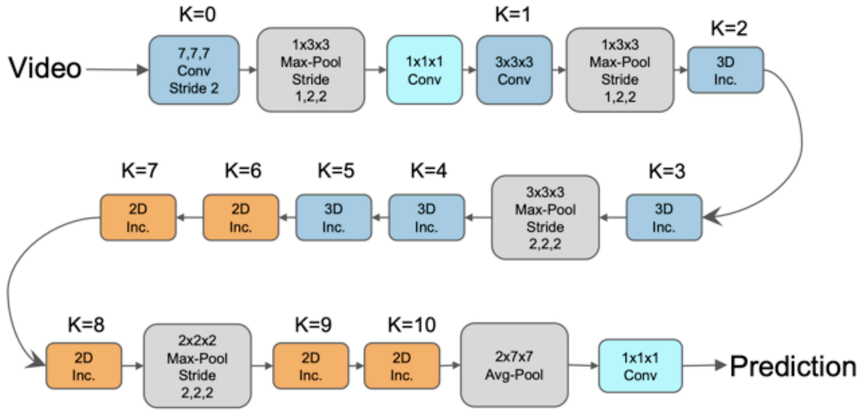

Video Features



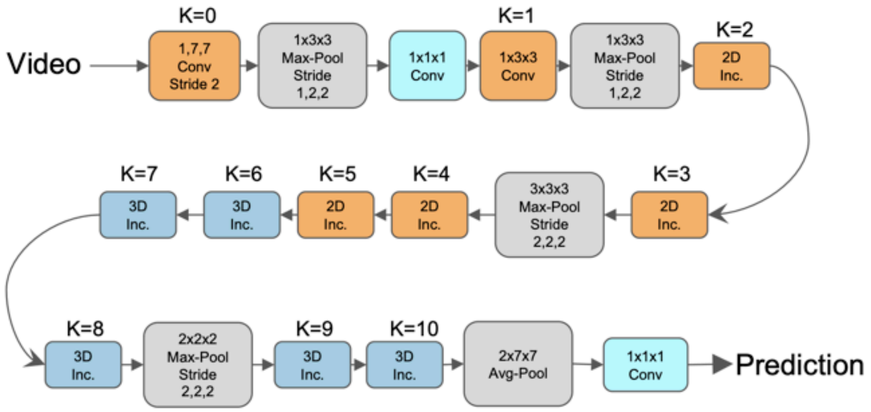
(a) I3D



(b) I2D



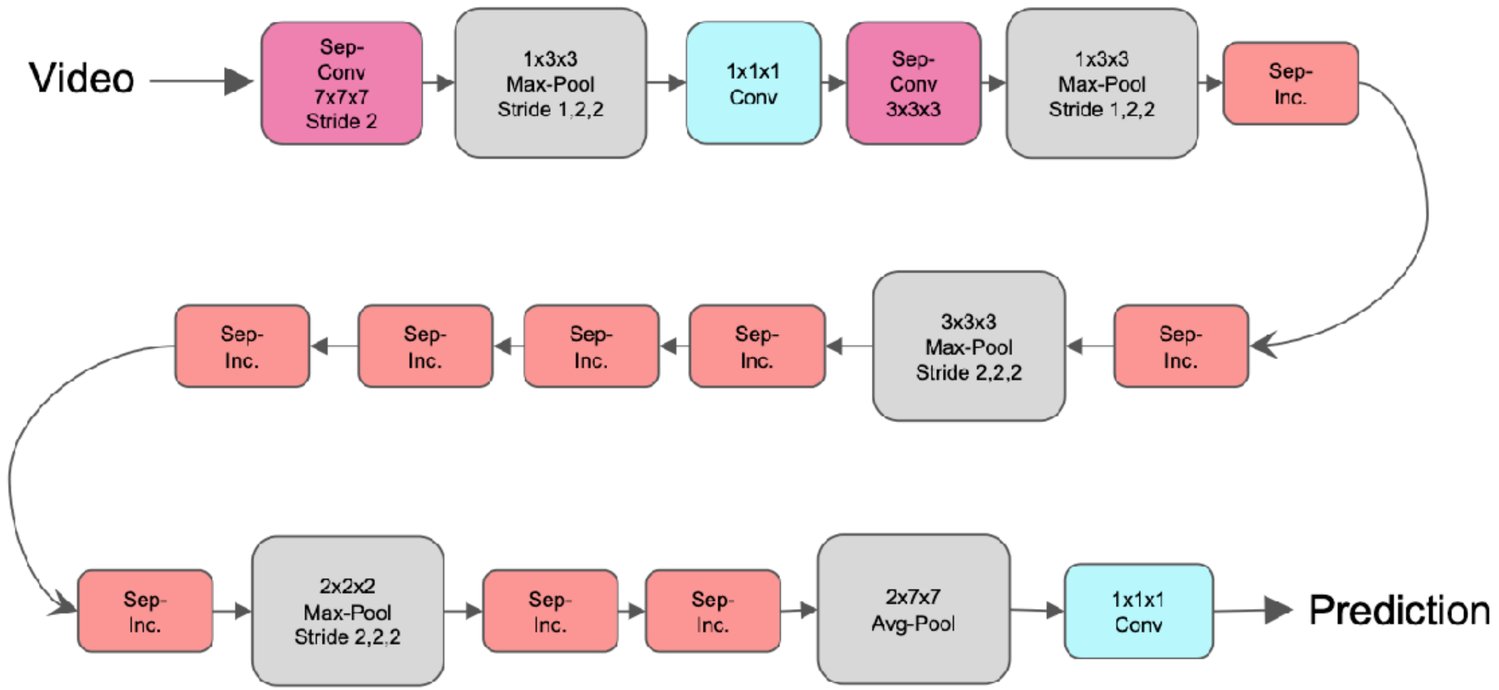
(c) Bottom-heavy I3D

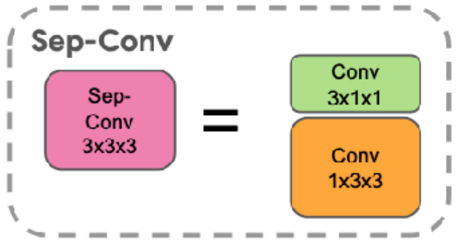


(d) Top-heavy I3D

https://arxiv.org/abs/1712.04851

S3D – Separable 3D CNN





$$k_t \times k \times k$$

$$1 \times k \times k$$

$$k_t \times 1 \times 1$$

k_t Width of the filter in time

https://arxiv.org/abs/1712.04851

Data (Something-Something)

https://20bn.com/products/datasets

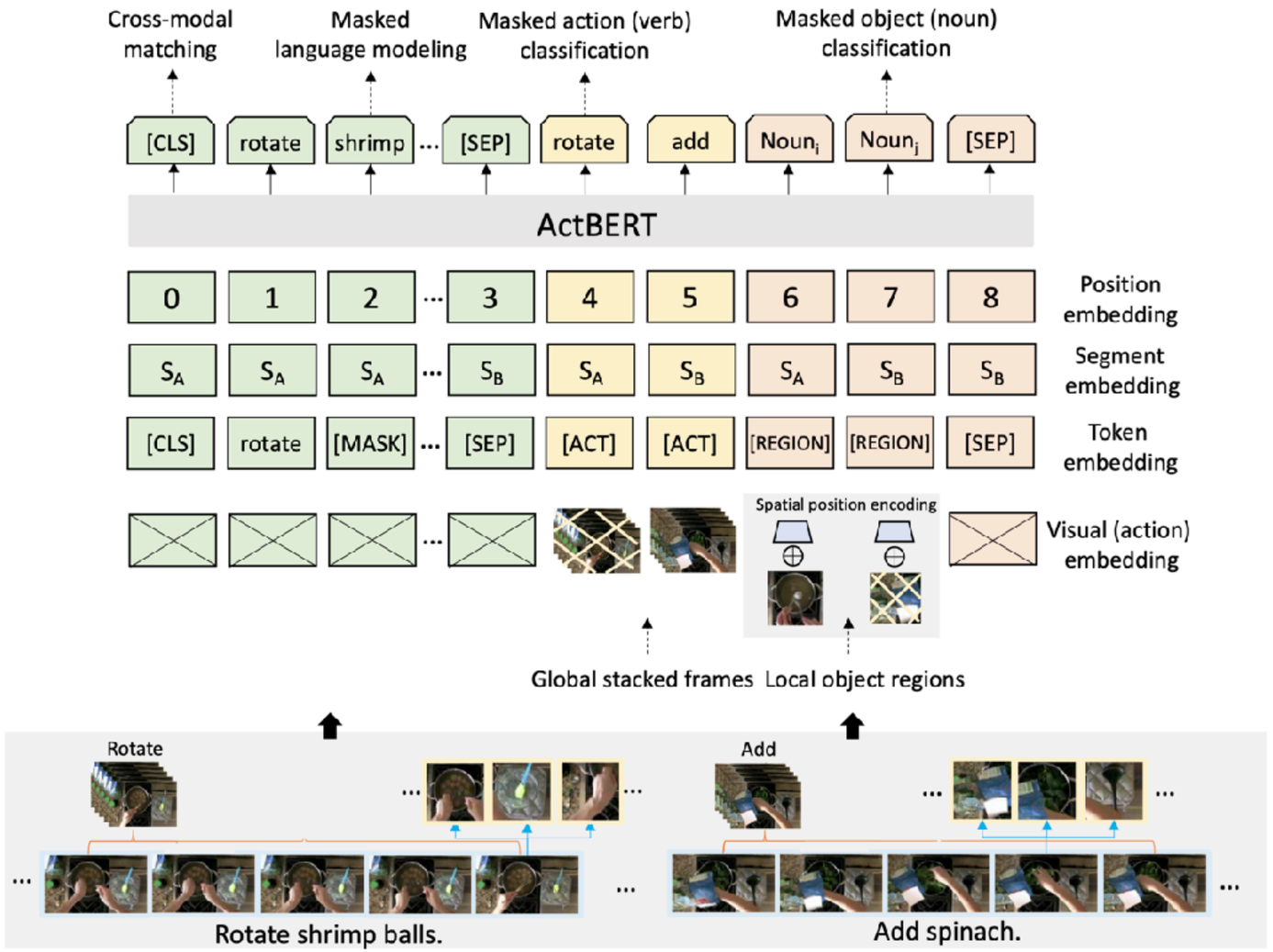
HowTo100M

Category	Tasks	Videos	Clips
Food and Entertaining	11504	497k	54.4M
Home and Garden	5068	270k	29.5M
Hobbies and Crafts	4273	251k	29.8M
Cars & Other Vehicles	810	68k	7.8M
Pets and Animals	552	31k	3.5M
Holidays and Traditions	411	27k	3.0M
Personal Care and Style	181	16k	1.6M
Sports and Fitness	205	16k	2.0M
Health	172	15k	1.7M
Education and Communications	239	15k	1.6M
Arts and Entertainment	138	10k	1.2M
Computers and Electronics	58	5k	0.6M
Total	23.6k	1.22M	136.6M

Table 2: Number of tasks, videos and clips within each category.

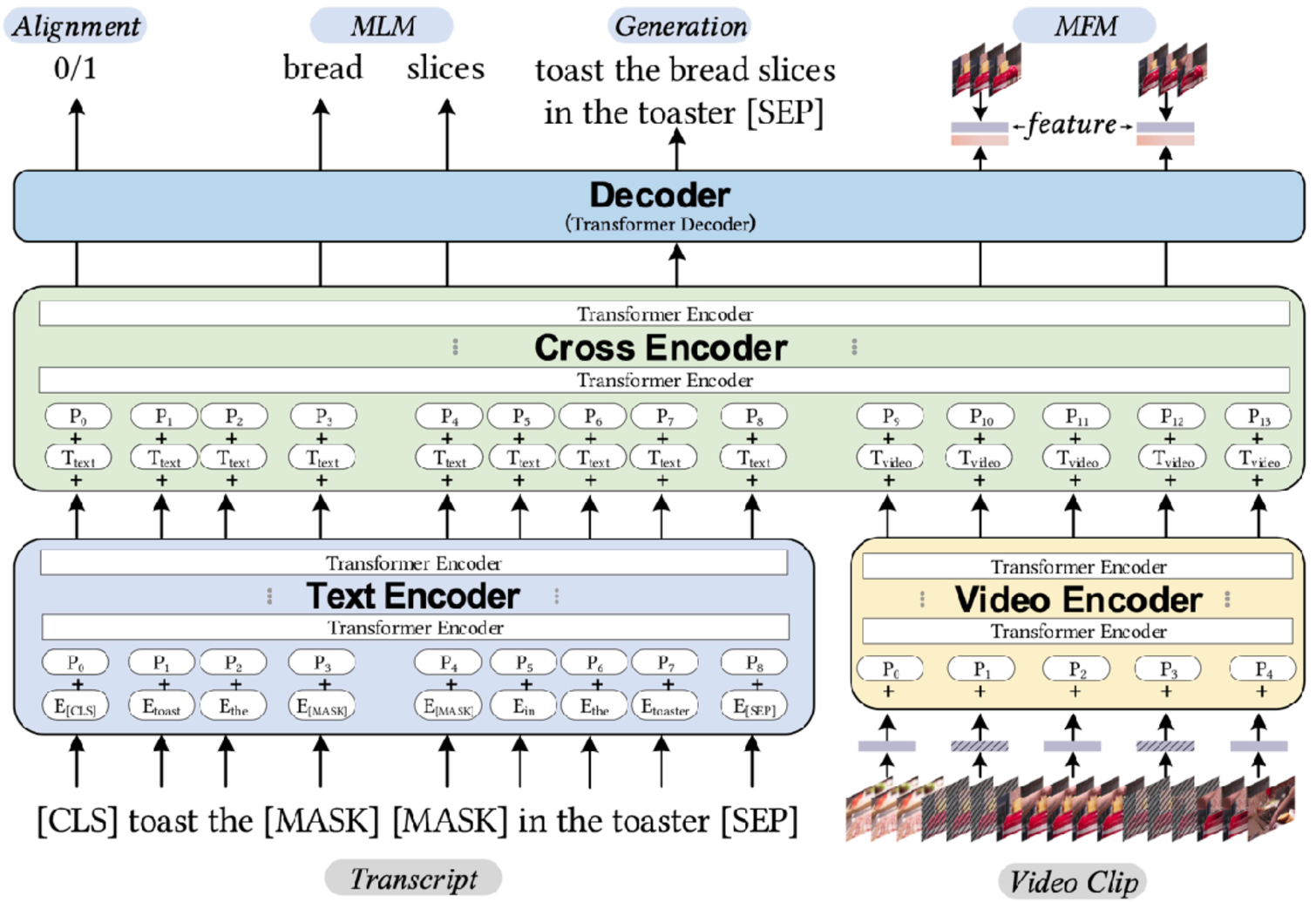
https://www.di.ens.fr/willow/research/howto100m/

ActBERT



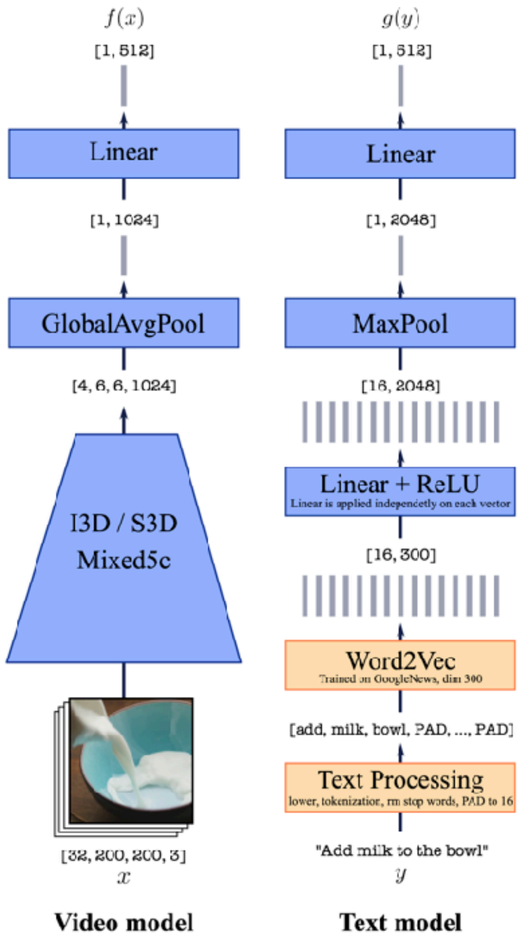
https://openaccess.thecvf.com/content_CVPR_2020/papers/Zhu_ActBERT_Learning_Global-Local_Video-Text_Representations_CVPR_2020_paper.pdf

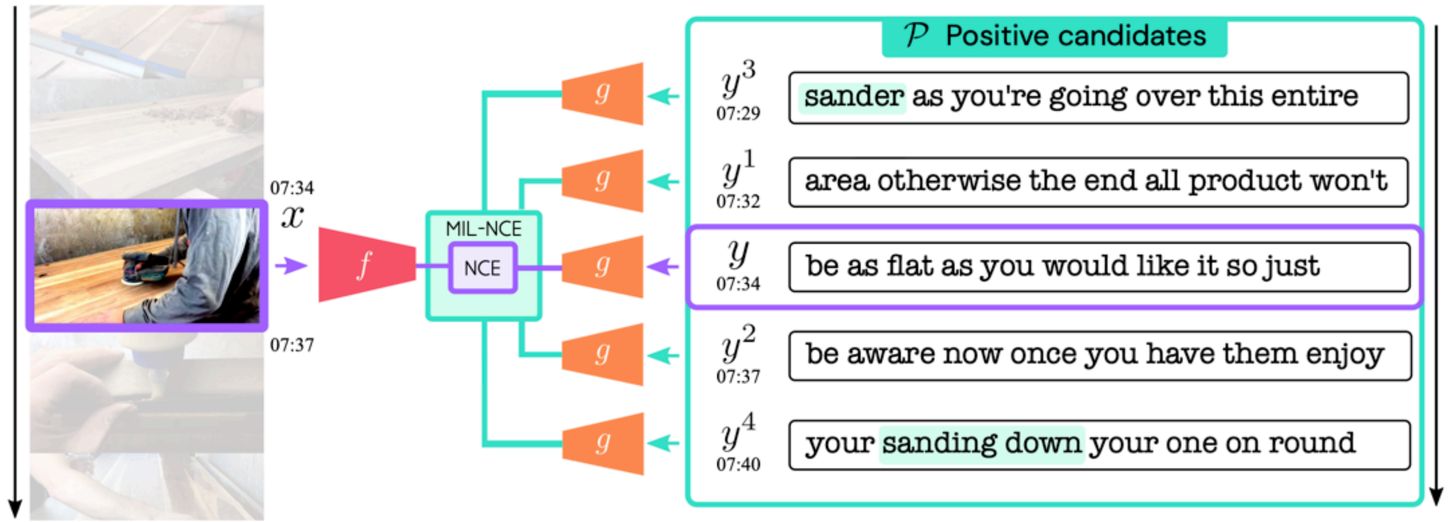
UniViLM



https://arxiv.org/abs/2002.06353v1

End-to-End Training





https://arxiv.org/abs/1912.06430

End-to-End Training

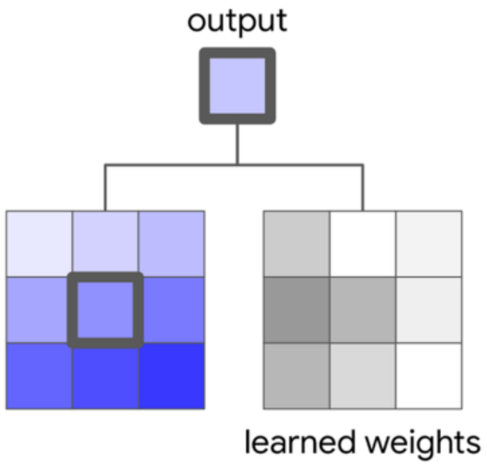
Input word vectors	YR10	MR10
BERT wo. stop words	19.0	19.7
BERT w. stop words	17.6	23.9
Word2Vec	**35.0**	**31.8**

https://github.com/antoine77340/MIL-NCE_HowTo100M

Implementation	Epochs	Total batch size	Accelerator	CPU cores	Training input size	R@1	R@5	R@10	Median Rank
TPU Tensorflow [1]	2645	8192	64 x Cloud TPU v3 128Gb	64 x N.A.	32 frames at 200x200	15.1	38.0	51.2	10
TPU Tensorflow [1]	206	512	4 x Cloud TPU v3 128Gb	4 x N.A.	16 frames at 200x200	8.2	24.6	36.2	23
This implementation	150	512	2 x 4 Tesla V100 32Gb	2 x 40	16 frames at 224x224	7.4	21.5	32.2	29
This implementation	300	1024	4 x 4 Tesla V100 32Gb	4 x 40	16 frames at 224x224	8.8	24.3	34.6	23

https://arxiv.org/abs/1912.06430

Replacing a CNN w/ Self-Attention



Convolution

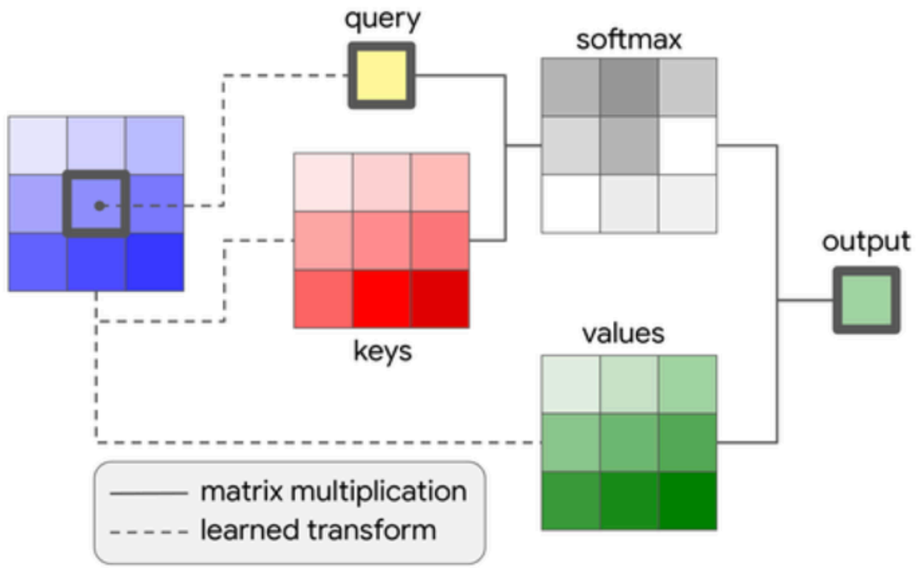


Figure 3: An example of a local attention layer over spatial extent of $k = 3$.

-1, -1	-1, 0	-1, 1	-1, 2
0, -1	0, 0	0, 1	0, 2
1, -1	1, 0	1, 1	1, 2
2, -1	2, 0	2, 1	2, 2

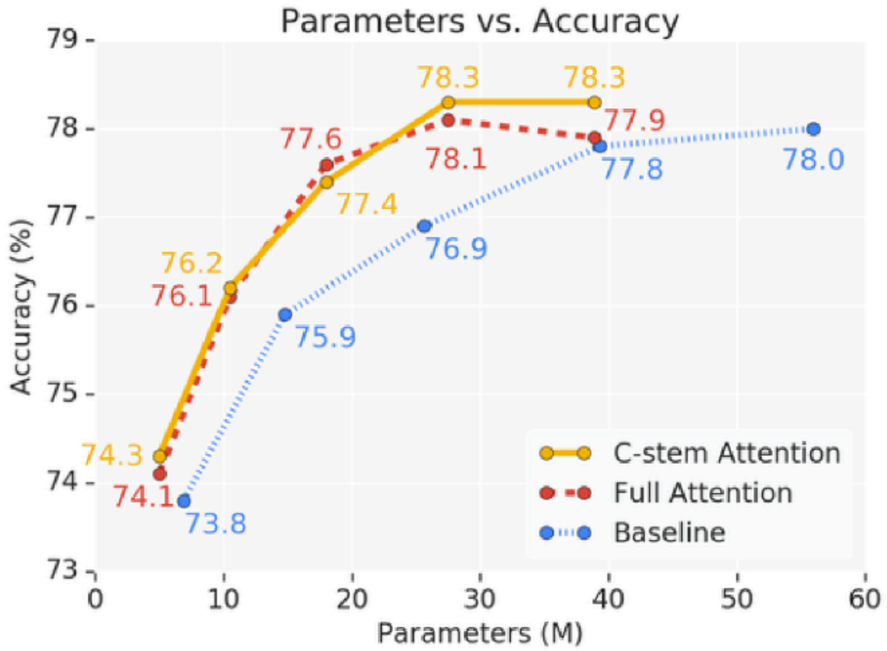
Parameters vs Computation

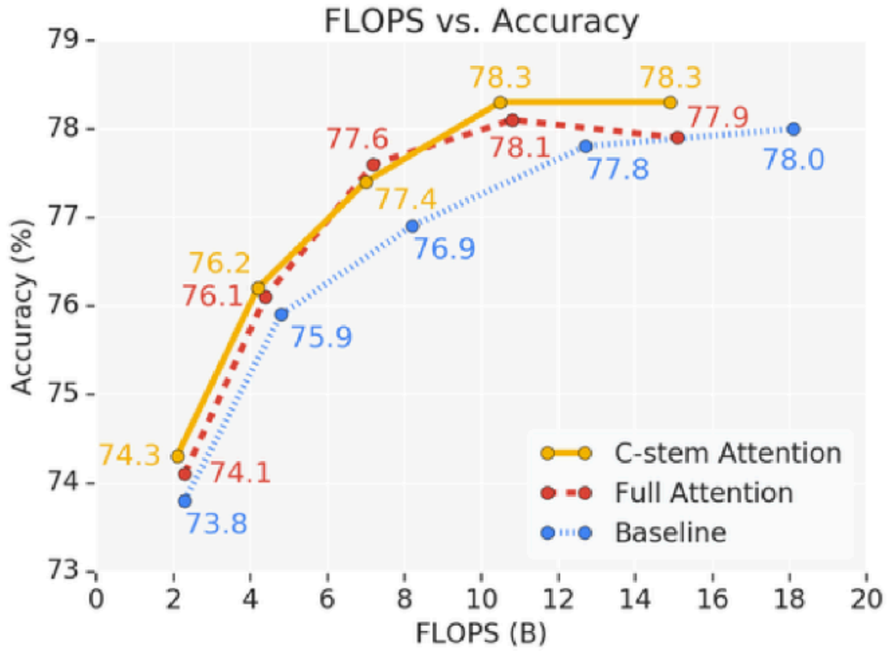
How do they change with receptive field?

https://arxiv.org/abs/1906.05909

Parameters/FLOPs vs Accuracy

ResNet 50 and k=7, 8 attention heads





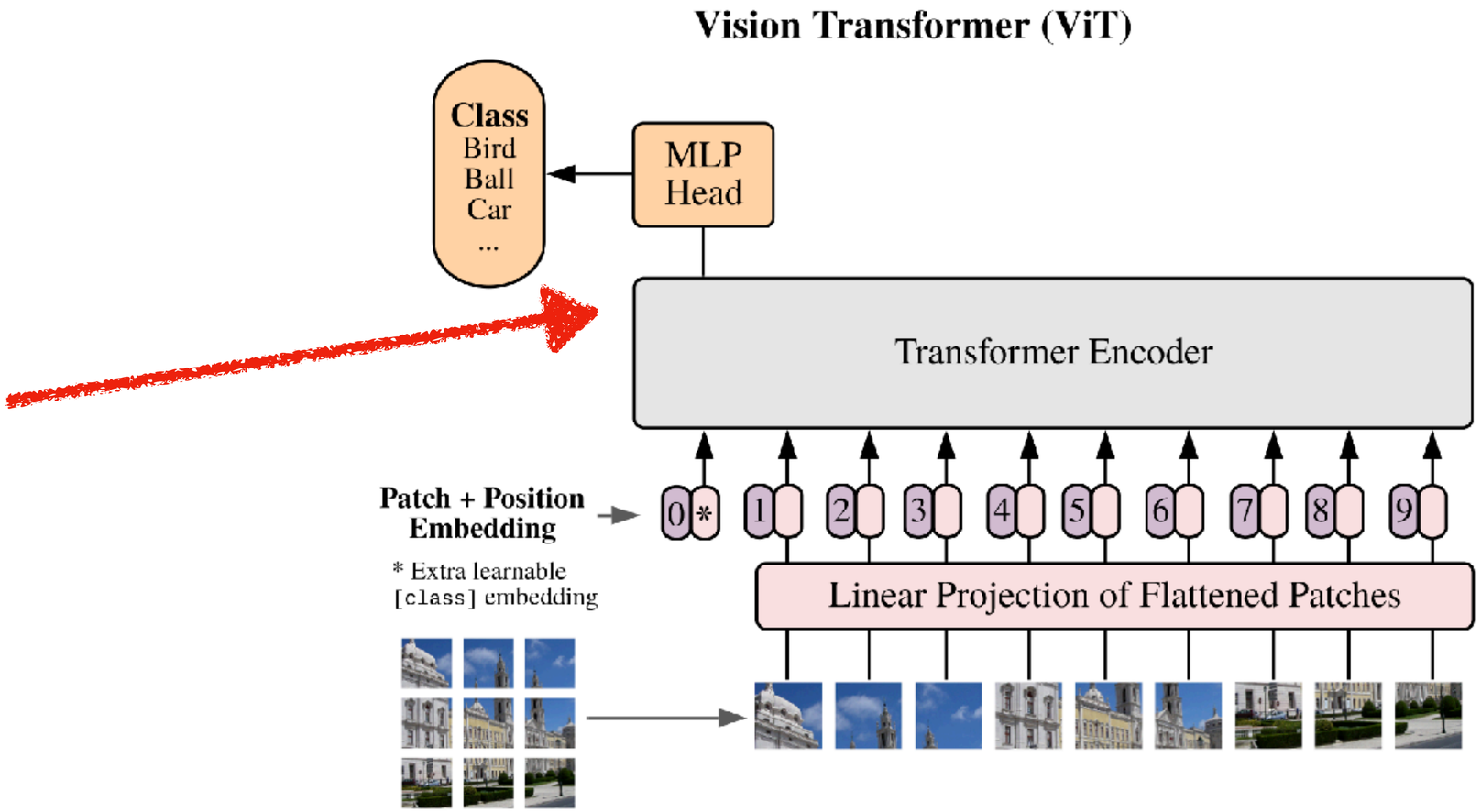
Spatial Extent ($k \times k$)	FLOPS (B)	Top-1 Acc. (%)
3×3	6.6	76.4
5×5	6.7	77.2
7×7	7.0	77.4
9×9	7.3	77.7
11×11	7.7	77.6

https://arxiv.org/abs/1906.05909

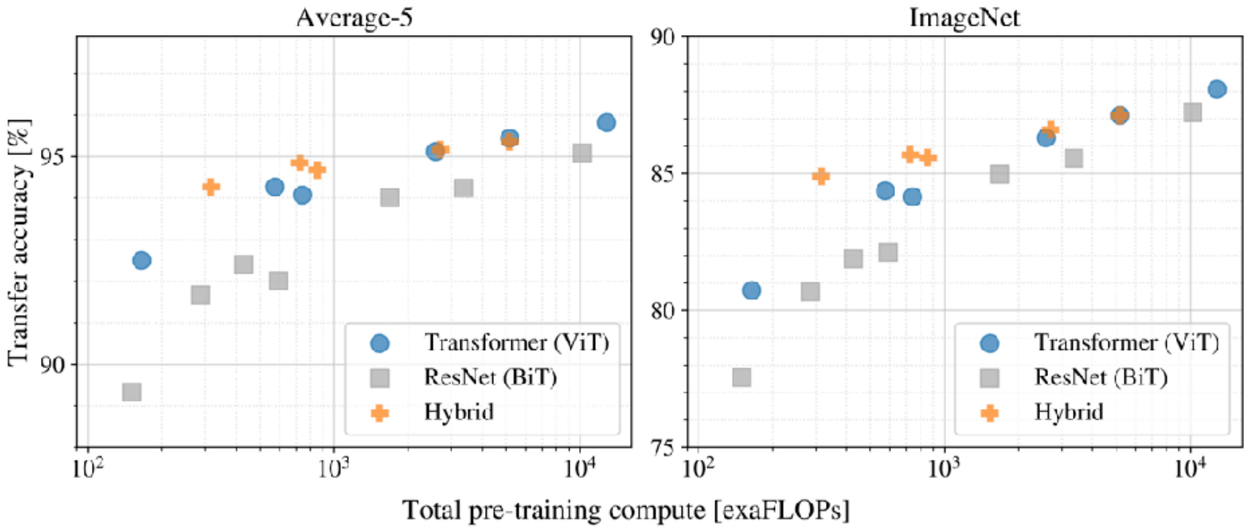
ViT

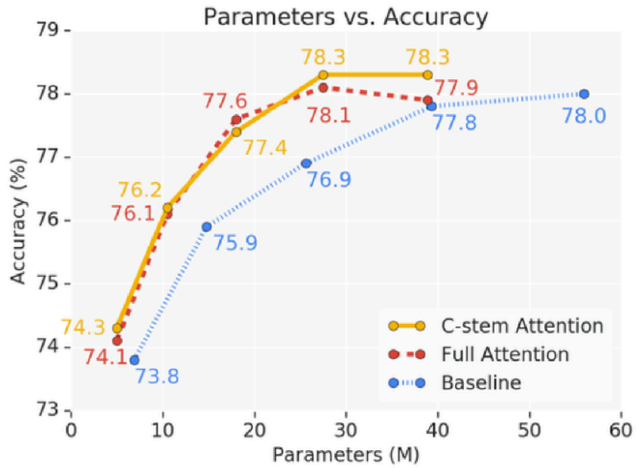
https://arxiv.org/abs/2010.11929

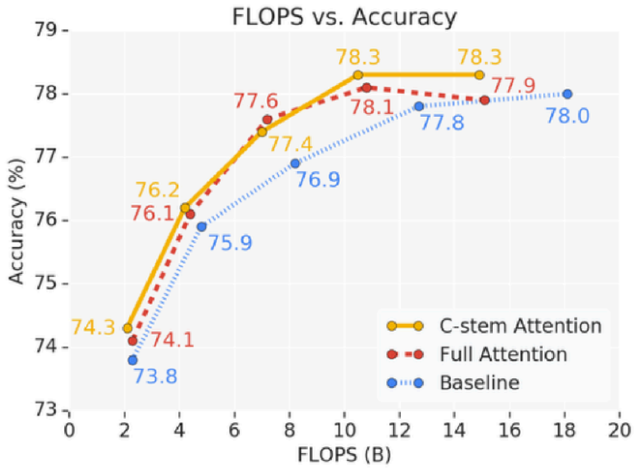
“Visual” Features



Curves


Average-5
ImageNet
Transfer accuracy [%]
Total pre-training compute [exaFLOPs]
Transformer (ViT)
ResNet (BiT)
Hybrid

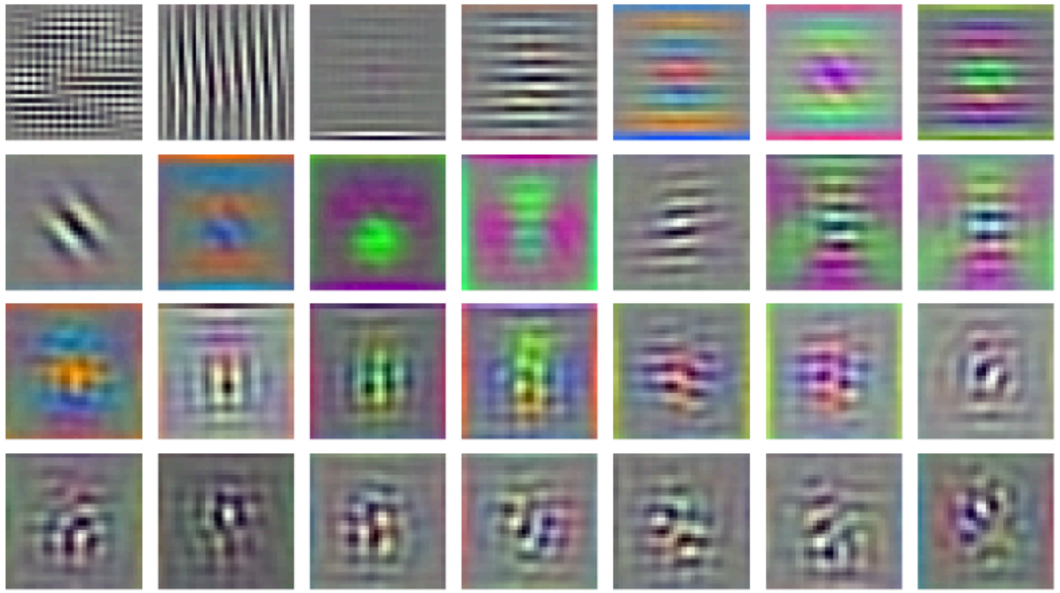

Parameters vs. Accuracy
Accuracy (%)
Parameters (M)
C-stem Attention
Full Attention
Baseline


FLOPS vs. Accuracy
Accuracy (%)
FLOPS (B)
C-stem Attention
Full Attention
Baseline

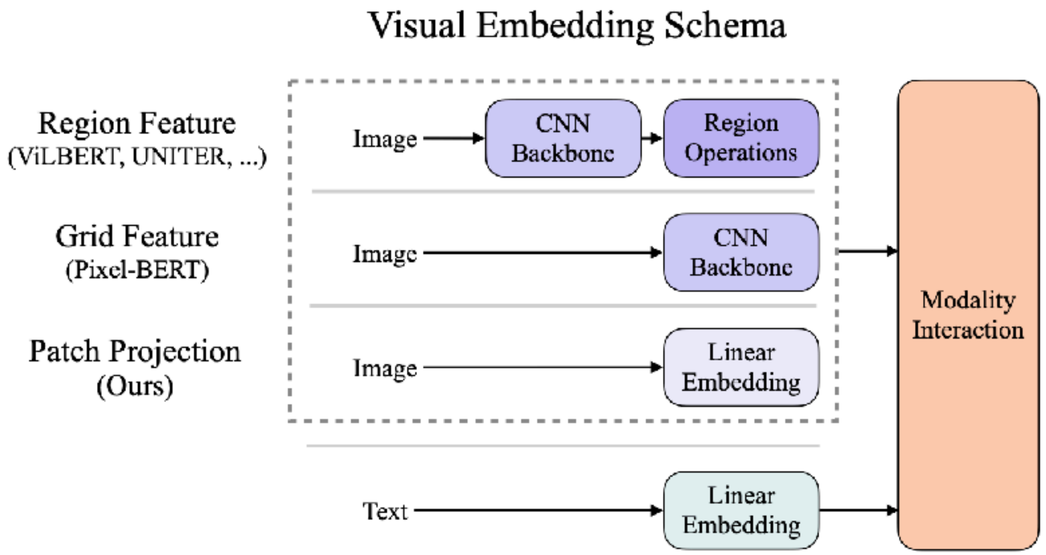
Which is the best?

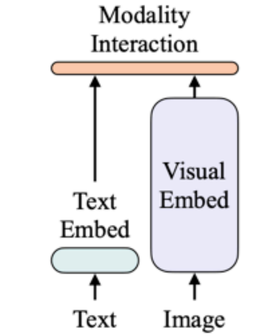
ImageNet	ImageNet ReaL	CIFAR-10	CIFAR-100	Pets	Flowers	exaFLOPs
80.73	86.27	98.61	90.49	93.40	99.27	164
84.15	88.85	99.00	91.87	95.80	99.56	743
84.37	88.28	99.19	92.52	95.83	99.45	574
86.30	89.43	99.38	93.46	96.81	99.66	2586
87.12	89.99	99.38	94.04	97.11	99.56	5172
88.08	90.36	99.50	94.71	97.11	99.71	12826
77.54	84.56	97.67	86.07	91.11	94.26	150
82.12	87.94	98.29	89.20	93.43	97.02	592
80.67	87.07	98.48	89.17	94.08	95.95	285
81.88	87.96	98.82	90.22	94.17	96.94	427
84.97	89.69	99.06	92.05	95.37	98.62	1681
85.56	89.89	99.24	91.92	95.75	98.75	3362
87.22	90.15	99.34	93.53	96.32	99.04	10212
84.90	89.15	99.01	92.24	95.75	99.46	315
85.58	89.65	99.14	92.63	96.65	99.40	855
85.68	89.04	99.24	92.93	96.97	99.43	725
86.60	89.72	99.18	93.64	97.03	99.40	2704
87.12	89.76	99.31	93.89	97.36	99.11	5165

Filters

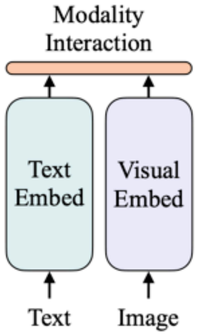


What about ROIs? What about Different Pipelines?

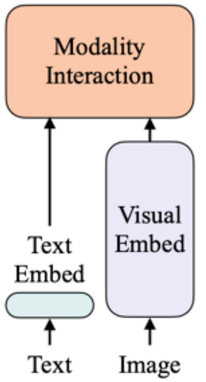




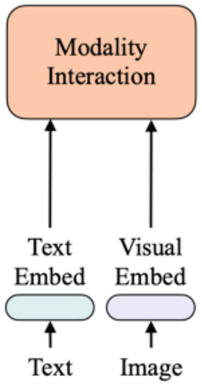
(a) VE > TE > MI



(b) VE = TE > MI



(c) VE > MI > TE

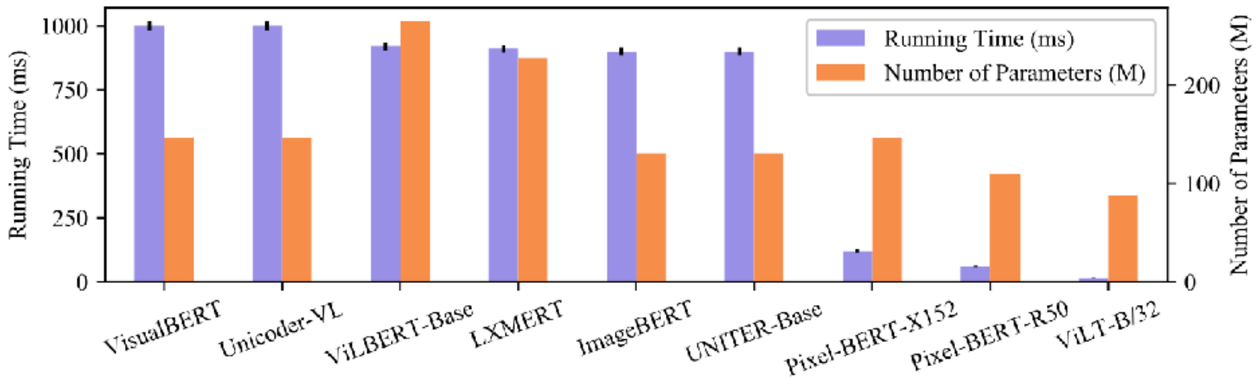


(d) MI > VE = TE

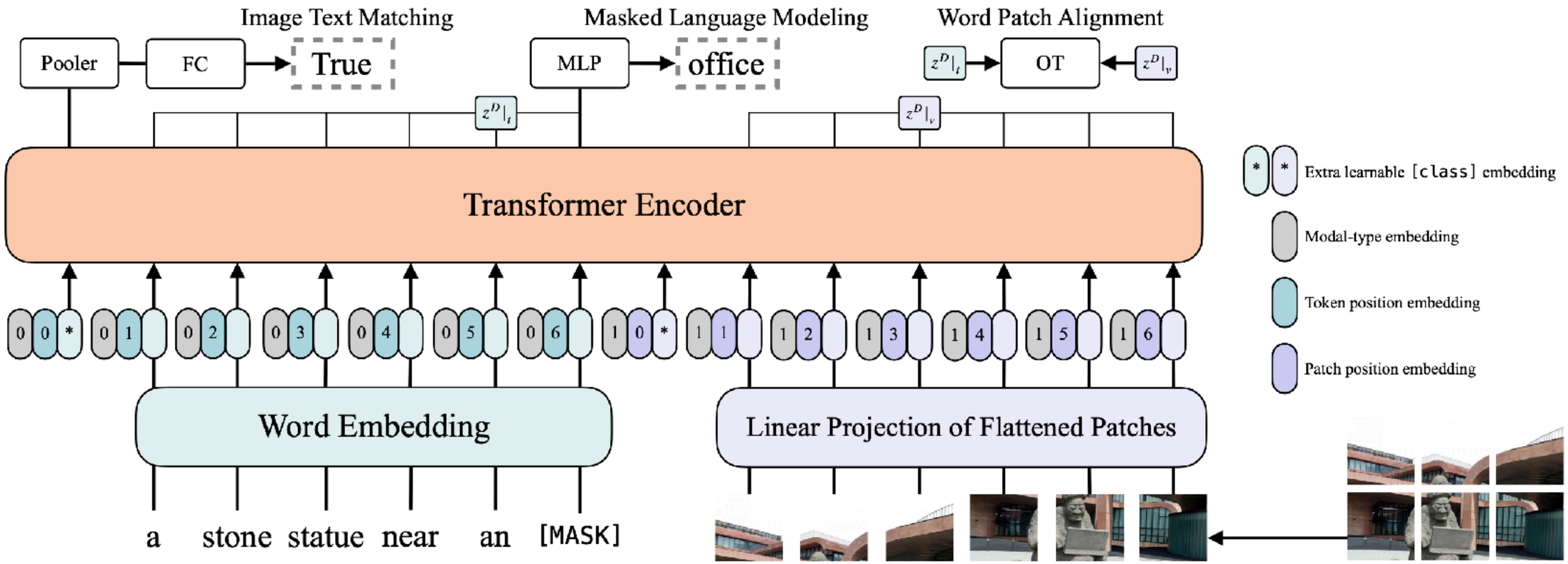
https://arxiv.org/abs/2102.03334

Faster Inference? What about training the features?

Visual Embed	Model	Time (ms)	VQAv2 test-dev	NLVR2 dev	NLVR2 test-P
Region	w/o VLP SOTA	~900	70.63	54.80	53.50
	ViLBERT	~920	70.55	-	-
	VisualBERT-Base	~1000	70.80	67.40	67.00
	LXMERT	~910	72.42	74.90	74.50
	UNITER-Base	~900	72.70	75.85	75.80
	OSCAR-Base†	~900	73.16	78.07	78.36
	VinVL-Base†‡	~1000	75.95	82.05	83.08
Grid	Pixel-BERT-X152	~120	74.45	76.50	77.20
	Pixel-BERT-R50	~60	71.35	71.70	72.40
Linear	ViLT-B/32	~15	70.34	74.56	74.66
	ViLT-B/32ⓐ	~15	70.94	75.24	76.21



Transformer “figures it out”



Alignment

a display of flowers growing out and over the retaining wall in front of cottages on a cloudy day.


flowers


wall


cottages


cloudy

a room with a rug, a chair, a painting, and a plant.

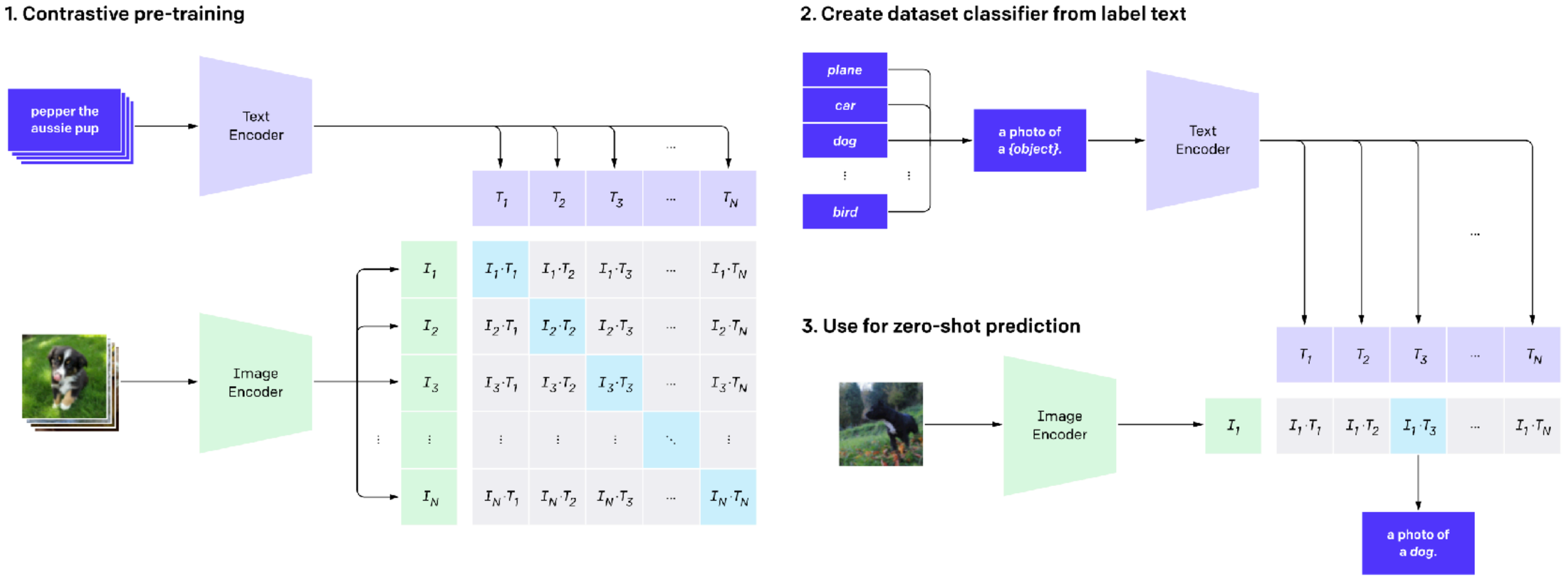

rug


chair


painting


plant

CLIP — 400 million (image, text) pairs



https://openai.com/blog/clip/

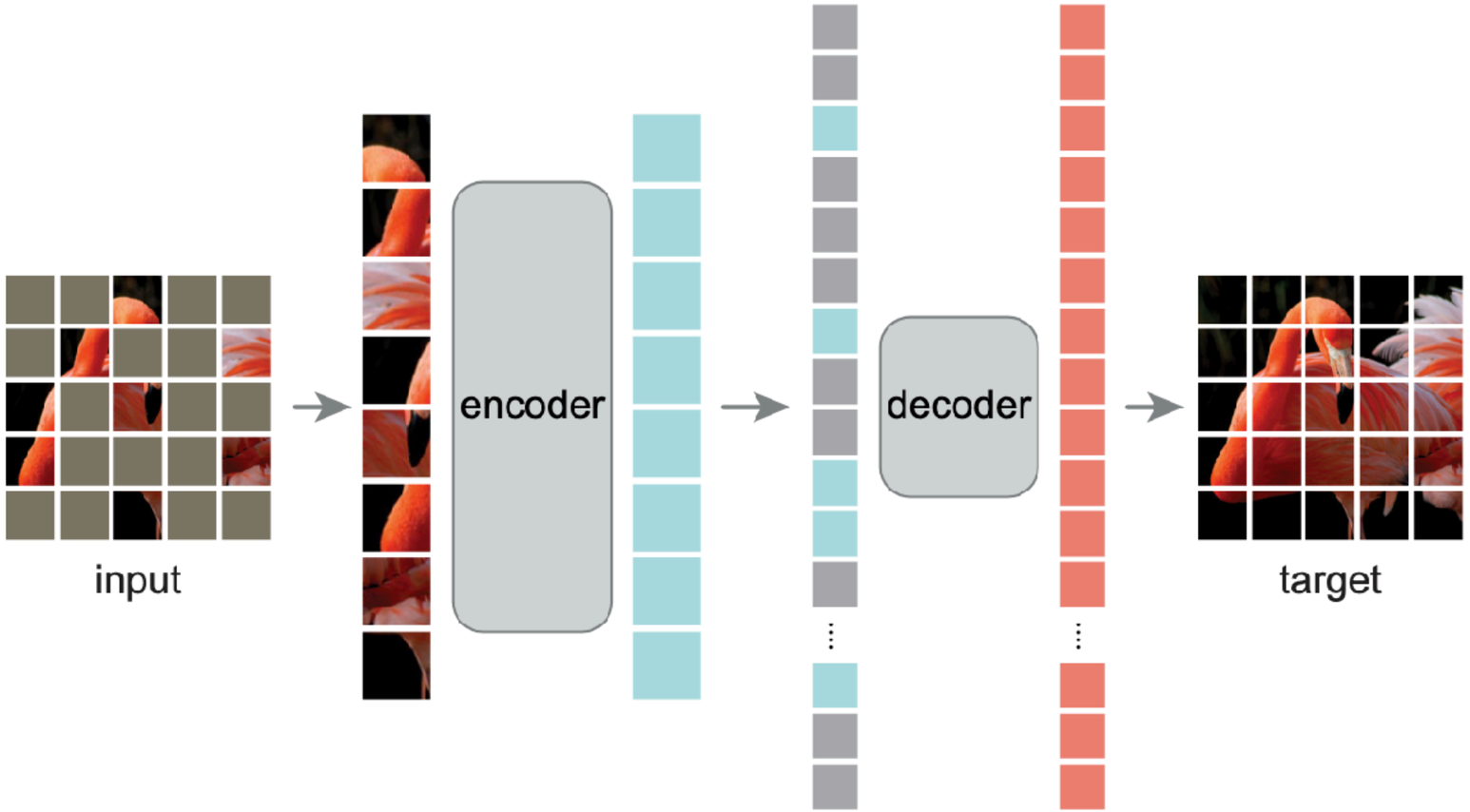
So should we Transformer?

		Food101	CIFAR10	CIFAR100	Birdsnap	SUN397	Cars	Aircraft	VOC2007	DTD	Pets	Caltech101	Flowers	MNIST	FER2013	STL10*	EuroSAT	RESISC45	GTSRB	KITTI	Country211	PCAM	UCF101	Kinetics700	CLEVR	HatefulMemes	SST	ImageNet
LM RN50		81.3	82.8	61.7	44.2	69.6	74.9	44.9	85.5	71.5	82.8	85.5	91.1	96.6	60.1	95.3	93.4	84.0	73.8	70.2	19.0	82.9	76.4	51.9	51.2	65.2	76.8	65.2
CLIP-RN	50	86.4	88.7	70.3	56.4	73.3	78.3	49.1	87.1	76.4	88.2	89.6	96.1	98.3	64.2	96.6	95.2	87.5	82.4	70.2	25.3	82.7	81.6	57.2	53.6	65.7	72.6	73.3
	101	88.9	**91.1**	73.5	58.6	75.1	84.0	50.7	88.0	76.3	91.0	92.0	96.4	98.4	65.2	97.8	95.9	89.3	82.4	**73.6**	26.6	82.8	84.0	60.3	50.3	68.2	73.3	75.7
	50x4	91.3	90.5	73.0	65.7	77.0	85.9	57.3	88.4	79.5	91.9	92.5	97.8	98.5	68.1	97.8	96.4	89.7	85.5	59.4	30.3	83.0	85.7	62.6	52.5	68.0	76.6	78.2
	50x16	93.3	92.2	74.9	72.8	79.2	88.7	62.7	**89.0**	79.1	93.5	93.7	98.3	**98.9**	68.7	98.6	97.0	91.4	89.0	69.2	34.8	83.5	88.0	66.3	53.8	71.1	**80.0**	81.5
	50x64	94.8	94.1	78.6	77.2	81.1	90.5	67.7	**88.9**	**82.0**	94.5	95.4	98.9	**98.9**	**71.3**	**99.1**	97.1	92.8	90.2	69.2	40.7	83.7	89.5	69.1	55.0	**75.0**	**81.2**	83.6
CLIP-ViT	B/32	88.8	95.1	80.5	58.5	76.6	81.8	52.0	87.7	76.5	90.0	93.0	96.9	**99.0**	69.2	98.3	97.0	90.5	85.3	66.2	27.8	83.9	85.5	61.7	52.1	66.7	70.8	76.1
	B/16	92.8	96.2	83.1	67.8	78.4	86.7	59.5	**89.2**	79.2	93.1	94.7	98.1	**99.0**	69.5	99.0	97.1	92.7	86.6	67.8	33.3	83.5	88.4	66.1	**57.1**	70.3	75.5	80.2
	L/14	95.2	98.0	87.5	77.0	**81.8**	**90.9**	69.4	**89.6**	**82.1**	**95.1**	**96.5**	99.2	**99.2**	**72.2**	**99.7**	**98.2**	94.1	**92.5**	64.7	42.9	85.8	**91.5**	72.0	**57.8**	**76.2**	**80.8**	83.9
	L/14-336px	**95.9**	97.9	87.4	**79.9**	**82.2**	**91.5**	**71.6**	**89.9**	**83.0**	**95.1**	**96.0**	99.2	**99.2**	**72.9**	**99.7**	**98.1**	**94.9**	**92.4**	69.2	**46.4**	85.6	**92.0**	**73.0**	**60.3**	**77.3**	**80.5**	85.4
EfficientNet	B0	74.3	92.5	76.5	59.7	62.0	62.5	55.7	84.4	71.2	93.0	93.3	91.7	98.2	57.2	97.1	97.3	85.5	80.0	**73.8**	12.4	83.1	74.4	47.6	47.9	55.7	53.4	76.9
	B1	74.2	93.2	77.2	61.3	62.6	62.5	56.1	84.7	74.2	93.4	93.6	92.4	98.3	57.0	97.5	96.8	84.5	75.9	**75.5**	12.5	82.7	74.7	48.5	44.3	54.5	54.4	78.6
	B2	75.8	93.6	77.9	64.4	64.0	63.2	57.0	85.3	73.5	93.9	93.5	92.9	98.5	56.6	97.7	96.9	**84.4**	76.4	**73.1**	12.6	84.3	75.1	49.4	42.6	55.4	55.2	79.7
	B3	77.4	94.0	78.0	66.5	64.4	66.0	59.3	85.8	73.1	94.1	93.7	93.3	98.5	57.1	98.2	97.3	85.0	75.8	**76.1**	13.4	83.3	78.1	50.9	45.1	53.8	54.8	81.0
	B4	79.7	94.1	78.7	70.1	65.4	66.4	60.4	86.5	73.4	94.7	93.5	93.2	98.8	57.9	98.6	96.8	85.0	78.3	**72.3**	13.9	83.1	79.1	52.5	46.5	54.4	55.4	82.9
	B5	81.5	93.6	77.9	72.4	67.1	72.7	68.9	86.7	73.9	**95.0**	94.7	94.5	98.4	58.5	98.7	96.8	86.0	78.5	69.6	14.9	84.7	80.9	54.5	46.6	53.3	56.3	83.7
	B6	82.4	94.0	78.0	73.5	65.8	71.1	68.2	87.6	73.9	**95.0**	94.1	93.7	98.4	60.2	98.7	96.8	85.4	78.1	**72.7**	15.3	84.2	80.0	54.1	51.1	53.3	57.0	84.0
	B7	84.5	94.9	80.1	74.7	69.0	77.1	**72.3**	87.2	76.8	**95.2**	94.7	95.9	98.6	61.3	**99.1**	96.3	86.8	80.8	**75.8**	16.4	85.2	81.9	56.8	51.9	54.4	57.8	84.8
	B8	84.5	95.0	80.7	75.2	69.6	76.8	**71.5**	87.4	77.1	**94.9**	95.2	96.3	98.6	61.4	99.2	97.0	87.4	80.4	70.9	17.4	85.2	82.4	57.7	51.4	51.7	55.8	85.3
EfficientNet Noisy Student	B0	78.1	94.0	78.6	63.5	65.5	57.2	53.7	85.6	75.6	93.8	93.1	94.5	98.1	55.6	98.2	97.0	84.3	74.0	71.6	14.0	83.1	76.7	51.7	47.3	55.7	55.0	78.5
	B1	80.4	95.1	80.2	66.6	67.6	59.6	53.7	86.2	77.0	94.6	94.4	95.1	98.0	56.1	98.6	96.9	84.3	73.1	67.1	14.5	83.9	79.9	54.5	46.1	54.3	54.9	81.1
	B2	80.9	95.3	81.3	67.6	67.9	60.9	55.2	86.3	77.7	**95.0**	94.7	94.4	98.0	55.5	98.8	97.3	84.6	71.7	70.0	14.6	82.9	80.1	**55.1**	**46.1**	54.1	55.3	82.2
	B3	82.6	95.9	82.1	68.6	68.8	60.6	55.4	86.5	77.2	**95.0**	94.8	95.2	98.1	56.0	99.1	96.5	85.0	70.5	69.5	15.1	83.1	81.8	56.8	45.1	55.7	52.0	83.8
	B4	85.2	95.6	81.0	72.5	69.7	56.1	52.6	87.0	78.7	**94.8**	95.2	95.3	98.2	56.0	99.3	95.3	84.8	61.9	64.8	16.0	82.8	83.4	59.8	43.2	55.3	53.0	85.4
	B5	87.6	96.3	82.4	75.3	71.6	64.7	64.8	87.8	79.6	**95.5**	95.6	96.6	98.8	60.9	99.4	96.1	87.0	68.5	**73.7**	16.4	83.5	86.4	61.6	46.3	53.4	55.8	85.8
	B6	87.3	97.0	83.9	75.8	71.4	67.6	65.6	87.3	78.5	**95.2**	**96.4**	97.2	98.6	61.9	**99.5**	96.6	86.1	70.7	**72.4**	17.6	84.2	85.5	61.0	49.6	54.6	55.7	86.4
	B7	88.4	96.0	82.0	76.9	72.6	72.2	**71.2**	88.1	**80.5**	**95.5**	95.5	96.6	98.5	62.7	99.4	96.2	88.5	73.4	**73.0**	18.5	83.8	86.6	63.2	50.5	57.2	56.7	87.0
	L2-475	91.6	**99.0**	**91.0**	74.8	76.4	75.1	66.8	**89.5**	**81.9**	**95.6**	**96.5**	97.7	**98.9**	67.5	**99.6**	97.0	89.5	73.4	68.9	22.2	86.3	89.4	68.2	**58.3**	58.6	55.2	**88.3**
	L2-800	92.0	**98.7**	89.0	**78.5**	75.7	75.5	68.4	**89.4**	**82.5**	**95.6**	94.7	97.9	98.5	68.4	**99.7**	97.2	89.9	77.7	66.9	23.7	**86.8**	88.9	66.7	**62.7**	58.4	56.9	**88.4**
Instagram	32x8d	84.8	95.9	80.9	63.8	69.0	74.2	56.0	88.0	75.4	**95.4**	93.9	91.7	97.4	60.7	99.1	95.7	82.1	72.3	69.2	16.7	82.3	80.1	56.8	42.2	53.3	55.2	83.3
	32x16d	85.7	96.5	80.9	64.8	70.5	77.5	56.7	87.9	76.2	**95.6**	94.9	92.5	97.4	61.6	99.3	95.5	82.8	73.8	66.1	17.5	83.4	81.1	58.2	41.3	54.2	56.1	84.4
	32x32d	86.7	96.8	82.7	67.1	71.5	77.5	55.4	88.3	78.5	**95.8**	95.3	94.4	97.9	62.4	99.3	95.7	85.4	71.2	66.8	18.0	83.7	82.1	58.8	39.7	55.3	56.7	85.0
	32x48d	86.9	96.8	83.4	65.9	72.2	76.6	53.2	88.0	77.2	**95.5**	95.8	93.6	98.1	63.7	99.4	95.3	85.4	73.0	67.2	18.5	82.7	82.8	59.2	41.3	55.5	56.7	85.2
	FixRes-v1	88.5	95.7	81.1	67.4	72.9	80.5	57.6	88.0	77.9	**95.8**	**96.1**	94.5	97.9	62.2	99.4	96.2	86.6	76.5	64.8	19.3	82.5	83.4	59.8	43.5	56.6	59.0	86.0
	FixRes-v2	88.5	95.7	81.1	67.3	72.9	80.7	57.5	88.0	77.9	**95.0**	**96.0**	94.5	98.0	62.1	99.4	96.5	86.6	76.3	64.8	19.5	82.3	83.5	59.8	44.2	56.6	59.0	86.0
BiT-S	R50x1	72.5	91.7	74.8	57.7	61.1	53.5	52.5	83.7	72.4	92.3	91.2	92.0	98.4	56.1	96.4	97.4	85.0	70.0	66.0	12.5	83.0	72.3	47.5	48.3	54.1	55.3	75.2
	R50x3	75.1	93.7	79.0	61.1	63.7	55.2	54.1	84.8	74.6	92.5	91.6	92.8	98.8	58.7	97.0	**97.8**	86.4	73.1	**73.8**	14.0	84.2	76.4	50.0	49.2	54.7	54.2	77.2
	R101x1	73.5	92.8	77.4	58.4	61.3	54.0	52.4	84.4	73.5	92.5	91.8	90.6	98.3	56.5	96.8	97.3	84.6	69.4	68.9	12.6	82.0	73.5	48.6	45.4	52.6	55.5	76.0
	R101x3	74.7	93.9	79.8	57.8	62.9	54.7	53.3	84.7	75.5	92.3	91.2	92.6	98.8	59.7	97.3	**98.0**	85.5	71.8	**80.2**	14.1	83.1	75.9	50.4	49.7	54.1	54.6	77.4
	R152x2	74.9	94.3	79.7	58.7	62.7	55.9	53.6	85.3	74.9	93.0	92.0	91.7	98.6	58.3	**97.1**	**97.8**	86.2	71.8	71.6	13.9	84.1	76.2	49.9	48.2	53.8	55.9	77.1
	R152x4	74.7	94.2	79.2	57.8	62.9	51.2	50.8	85.4	75.4	93.1	91.2	91.4	**98.9**	61.4	97.2	**98.0**	85.5	72.8	67.9	14.9	83.1	76.0	50.3	42.9	53.6	56.0	78.5
BiT-M	R50x1	83.3	94.9	82.2	70.9	69.9	59.0	55.6	86.8	77.3	91.5	93.9	**99.4**	98.0	60.6	98.4	97.5	87.4	68.6	68.2	16.6	82.5	79.4	53.2	49.4	54.5	53.4	76.7
	R50x3	86.9	96.7	86.2	75.7	74.6	60.6	54.2	87.7	78.5	93.2	95.3	**99.4**	98.6	64.6	99.3	**98.0**	88.1	69.9	59.6	19.6	83.4	83.5	57.8	51.3	55.8	55.6	80.7
	R101x1	85.5	95.7	84.4	73.0	72.5	59.8	55.0	87.3	78.1	92.2	95.0	**99.5**	98.1	62.5	99.0	**97.6**	87.8	68.7	67.7	18.0	84.0	82.3	55.9	53.4	54.8	53.1	79.4
	R101x3	87.2	97.4	87.5	72.4	75.0	57.4	47.4	87.5	79.6	93.2	95.4	**99.6**	98.6	64.3	99.4	**98.2**	87.7	68.8	64.1	20.7	80.4	84.0	58.7	52.6	54.9	54.3	81.2
	R152x2	88.0	97.5	87.8	75.8	75.9	61.5	55.3	88.1	79.8	93.6	**95.9**	**99.5**	98.5	64.3	**99.5**	**97.9**	89.0	70.0	70.3	20.7	82.6	85.5	59.6	50.8	54.9	55.1	81.9
	R152x4	87.2	97.6	88.2	72.4	75.0	49.1	43.4	87.1	79.9	92.4	95.4	99.3	98.5	65.7	**99.5**	**97.8**	87.7	68.2	57.1	20.6	80.4	84.6	59.0	49.7	57.2	55.1	81.5
ViT	B/32	81.8	96.7	86.3	65.2	70.7	49.1	42.7	85.3	73.1	90.4	94.5	98.7	97.8	59.0	99.0	96.3	83.0	68.1	65.1	15.7	82.6	79.1	51.7	38.9	57.1	54.6	76.6
	B/16	86.7	96.9	86.4	74.0	74.2	54.7	46.0	86.7	74.3	92.7	94.1	99.2	97.4	61.3	99.5	96.4	84.5	63.1	61.5	17.5	85.4	82.7	56.6	40.0	57.0	56.1	80.9
	L/16	87.4	97.9	89.0	76.5	74.9	62.5	52.2	86.1	75.0	92.9	94.7	99.3	98.0	64.0	**99.6**	96.5	85.7	70.4	58.8	17.7	85.7	84.1	58.0	38.4	58.4	52.8	81.9
	H/14	88.4	95.8	84.5	70.2	69.2	62.3	54.8	84.7	75.4	91.7	93.7	98.9	98.5	62.4	98.4	97.3	87.0	73.9	63.4	15.4	**87.0**	79.4	52.1	41.1	55.9	54.1	75.9
SimCLRv2	R50x1	76.4	93.2	77.9	48.6	64.1	56.3	51.7	84.4	77.0	88.3	91.8	92.9	97.6	59.7	96.7	97.5	85.8	71.1	69.1	15.8	84.8	78.4	51.0	56.2	53.9	53.8	73.8
	R50x3	81.0	95.6	82.4	56.5	67.0	65.6	61.1	85.9	78.8	90.9	94.1	95.4	98.7	62.6	98.2	**97.9**	88.2	78.2	**74.7**	17.6	85.4	82.6	54.6	55.4	54.2	55.2	77.3
	R101x1	77.9	94.8	79.9	51.9	65.2	57.1	52.0	85.4	77.2	90.0	91.6	92.7	97.2	59.4	97.6	96.8	84.6	65.7	70.6	16.1	84.3	78.8	52.4	53.6	55.1	55.7	76.1
	R101x3	82.2	96.4	83.4	57.5	68.2	64.6	60.0	86.2	78.9	91.8	95.0	95.4	98.4	63.0	98.5	**97.9**	88.0	77.5	69.1	18.3	85.5	82.9	55.9	52.2	54.5	56.3	78.8
	R152x1	78.6	95.0	79.9	50.3	65.6	55.6	52.2	85.8	77.3	90.1	92.5	91.8	97.6	59.8	98.1	96.6	84.3	64.8	70.3	16.6	83.9	79.4	53.1	**57.2**	55.8	54.8	76.9
	R152x2	82.3	96.7	82.9	58.1	68.5	64.9	58.7	86.6	79.1	92.2	94.1	96.0	98.2	64.1	98.5	**98.0**	88.1	77.0	69.8	18.4	85.3	82.7	56.2	53.6	56.0	56.5	79.2
	R152x3	83.6	96.8	84.5	60.3	69.1	68.5	63.1	86.7	**80.5**	92.6	94.9	96.3	98.7	65.4	98.8	**98.1**	89.5	78.4	68.5	19.4	85.2	83.5	57.0	54.4	54.6	54.2	80.0
BYOL	50x1	74.0	93.6	79.1	47.6	63.7	61.6	62.3	82.6	77.0	88.3	93.7	94.3	98.7	58.8	96.4	**97.6**	88.2	80.1	71.4	14.1	84.8	77.3	49.3	56.1	53.8	54.4	73.3
	200x2	78.5	96.2	83.3	53.4	68.5	61.7	55.4	86.6	77.4	91.9	95.5	93.9	98.7	62.6	98.6	**97.7**	87.4	**77.1**	**76.4**	16.4	84.0	82.6	55.1	54.1	52.5	52.4	79.2
MoCo	v1	65.9	85.0	63.1	27.5	52.6	35.9	43.5	75.7	70.0	70.4	78.1	85.4	97.6	54.3	85.6	97.1	82.9	62.6	60.2	12.6	85.7	64.2	40.7	54.7	55.6	53.5	57.2
	v2	72.2	93.4	76.3	39.6	60.2	48.3	51.1	82.6	75.1	84.4	89.9	90.7	98.4	58.3	95.7	97.2	85.4	75.7	**75.4**	13.2	85.6	72.7	47.8	**56.9**	53.9	53.8	69.1
VirTex		57.9	83.9	57.5	17.0	49.8	22.4	34.5	83.8	58.2	53.6	70.6	74.7	98.1	56.5	86.7	94.8	74.1	69.5	71.3	8.7	83.1	61.5	39.9	45.5	53.5	55.8	50.7
ResNet	50	71.3	91.8	74.5	52.7	60.5	49.9	48.5	83.8	72.3	92.4	90.8	90.8	98.3	54.9	96.4	96.7	83.6	70.6	67.1	11.7	82.5	71.2	46.8	43.0	56.5	55.5	74.3
	101	72.7	93.0	77.2	53.7	60.8	50.1	47.0	84.4	71.6	92.3	91.9	90.4	98.5	56.6	97.0	97.1	83.4	72.5	63.6	11.9	83.3	72.7	48.3	43.2	53.0	54.7	75.8
	152	73.7	93.5	78.0	55.1	61.6	52.8	48.4	84.5	71.9	93.0	92.1	89.6	98.2	57.0	97.6	97.0	83.1	70.1	70.2	12.3	82.9	75.3	49.2	42.4	53.2	53.9	77.1

Table 10. Linear probe performance of various pre-trained models over 27 datasets. Scores within the 99.5% Clopper-Pearson confidence interval of each dataset's top score are shown in bold.

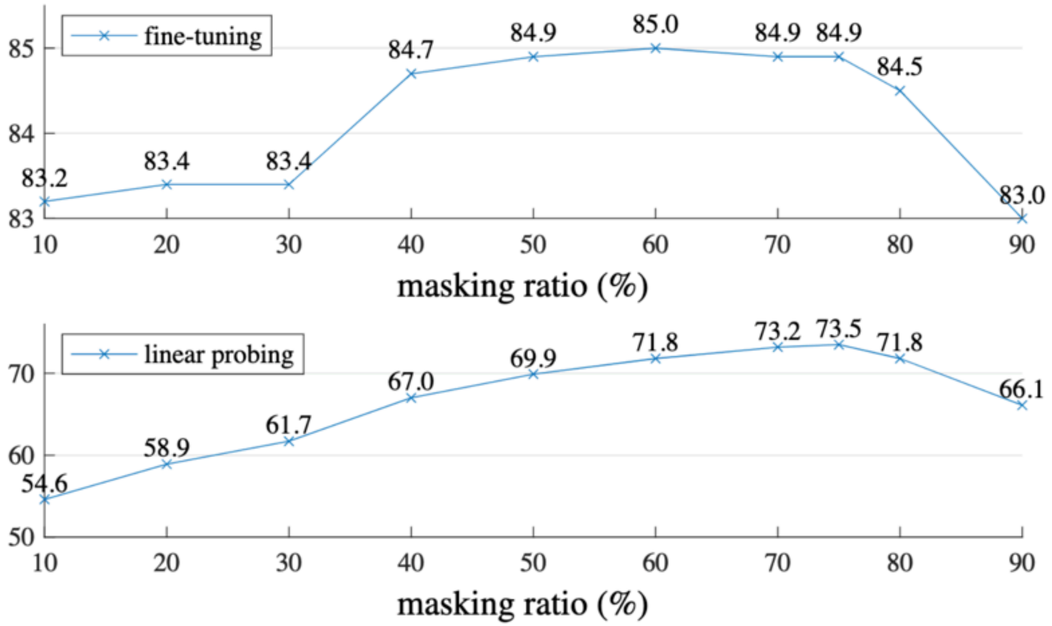
Self-Supervised Vision Transformers

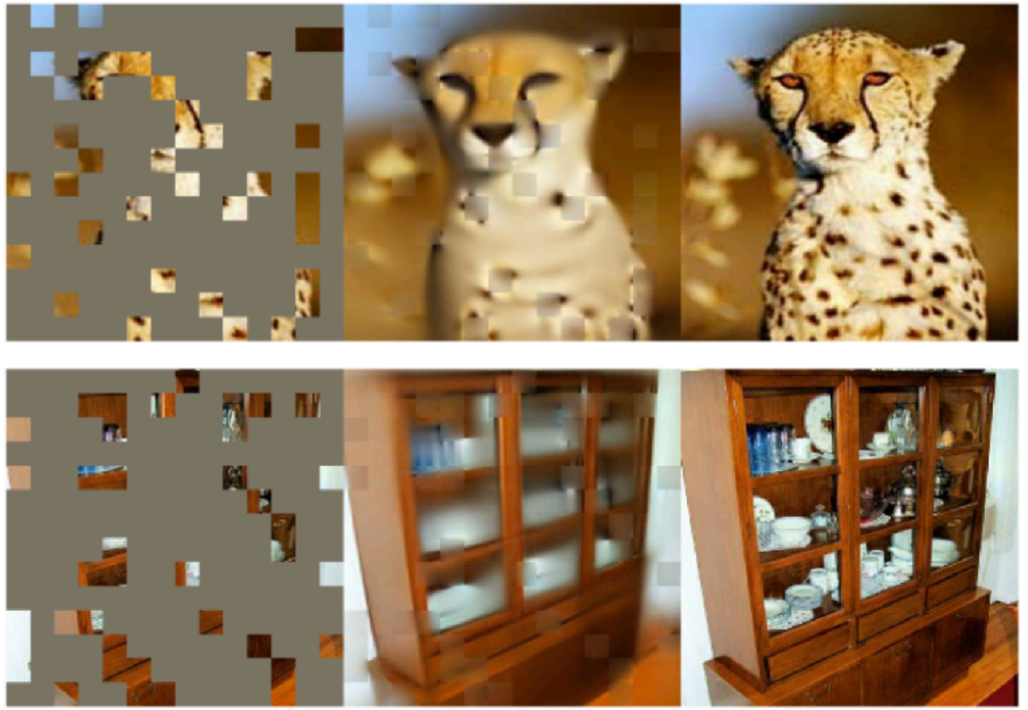
Masked Autoencoders Are Scalable Vision Learners



https://arxiv.org/abs/2111.06377

What are we learning?



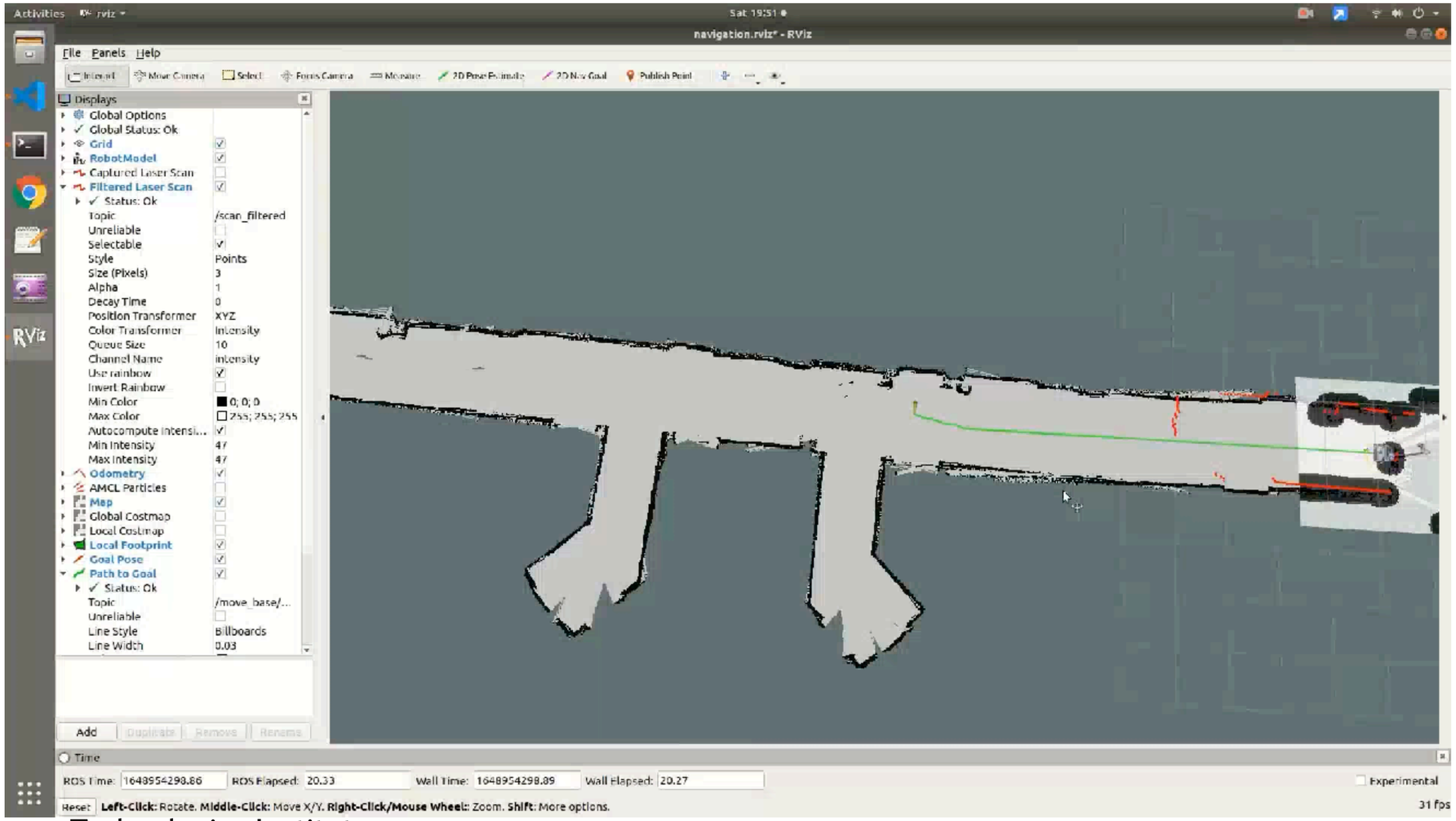


What even is a Robot?

Are you seriously going to try and cram a robotics class into this lecture? What do you have? Like 5m left?

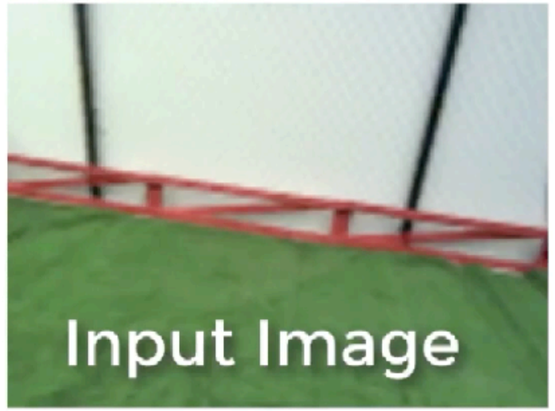
Mapping

- Sensors
- Location estimation
- Map updates



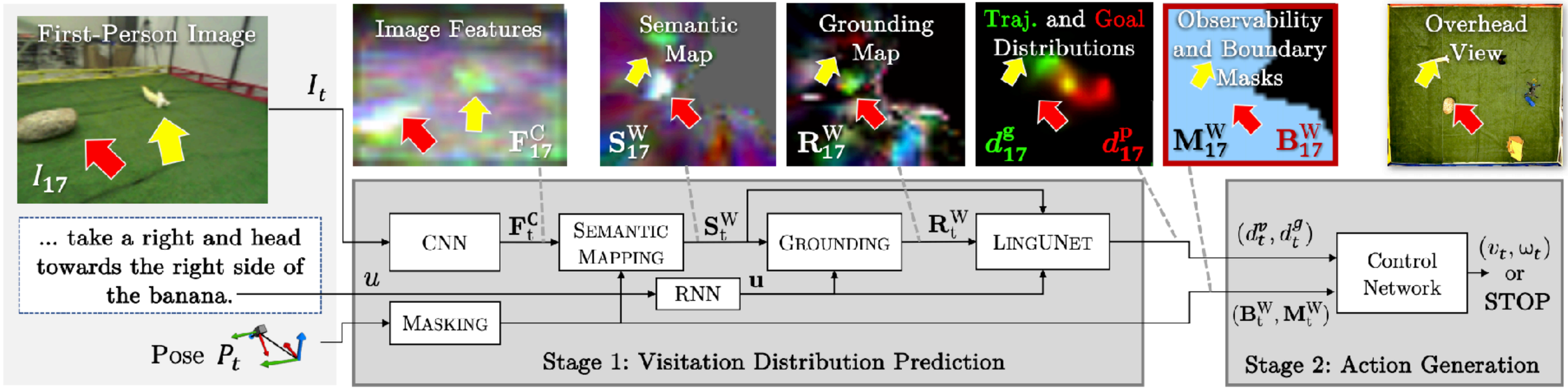
Real Valued Predictions

Once near the rear of the gorilla, turn right and head towards the rock stopping once near it

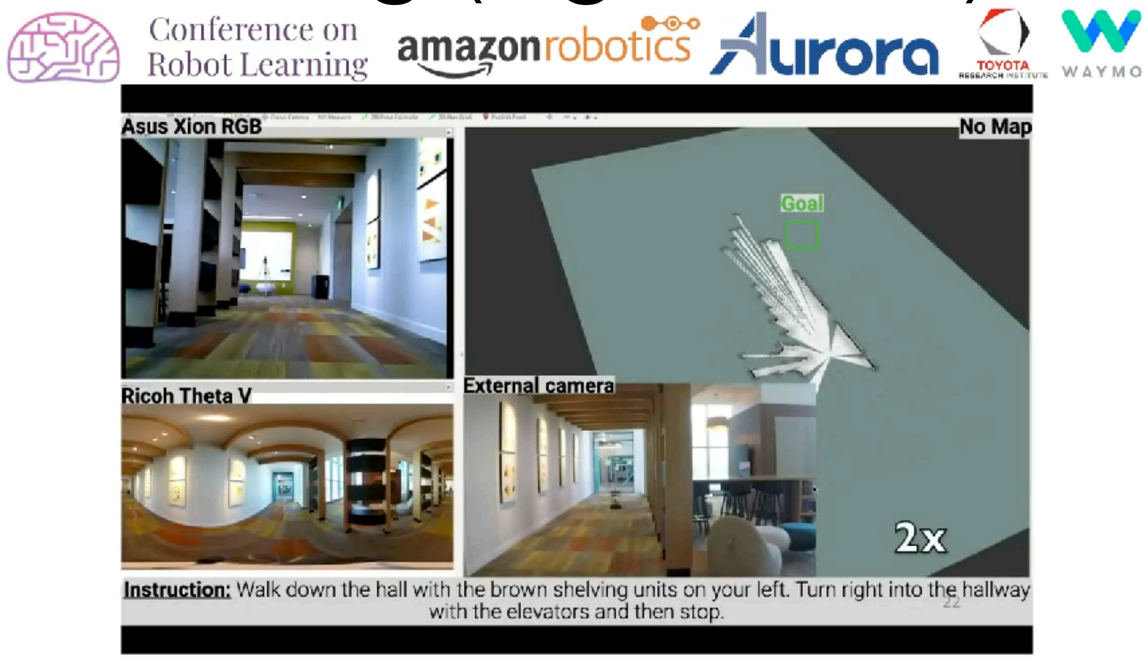


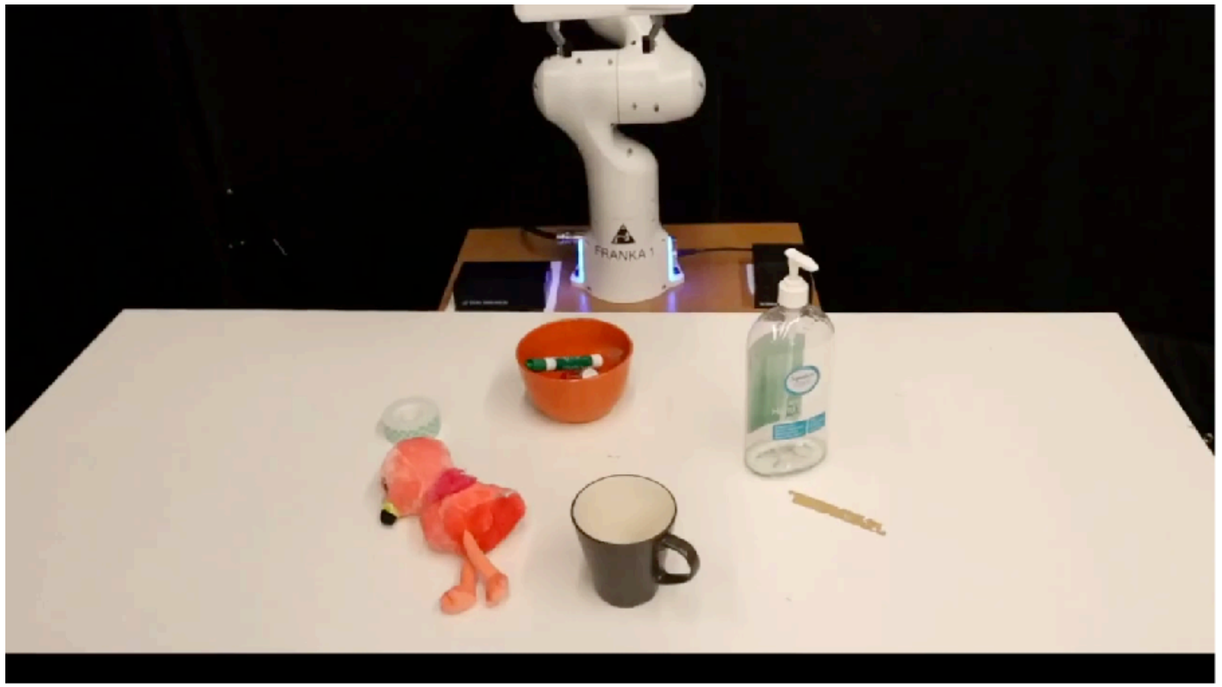
https://arxiv.org/abs/1910.09664

Lots of individual models / algorithms



Planning (algorithms)





- Inverse Kinematics — What sequence of joint movements is required to get to a target position (e.g. gripper)
- A*, Dijkstra — What's the shortest path from point A to B
- PDDL — Given pre- and post-conditions, what sequence of actions are required to complete a task?