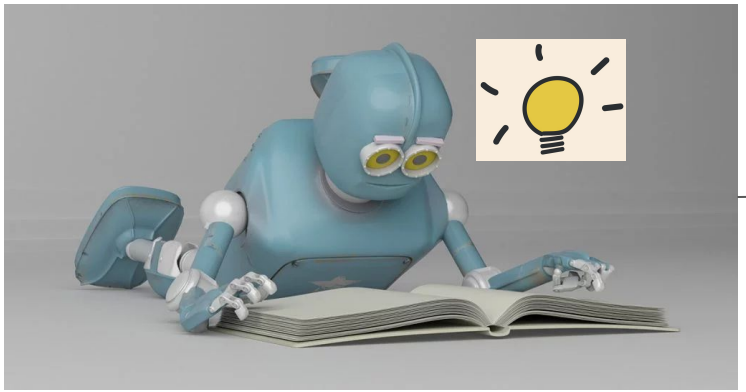
Lecture 13: Pretext Tasks

EES598 - 012 (Win2021)

Presenters: Luya Gao, Yuliang Zhu

Background

What is the pretext task for computer vision?



Solving pretext tasks

Visual representation / Model weights

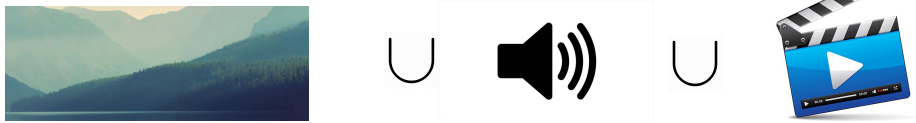
Downstream tasks

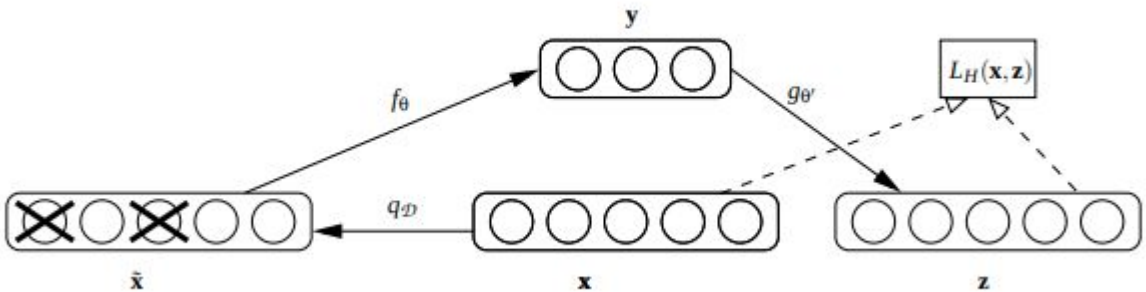
- Classification
- Regression
- Segmentation
- .
- .
- .

Background

Definition

Computer vision pretext tasks can be developed by:



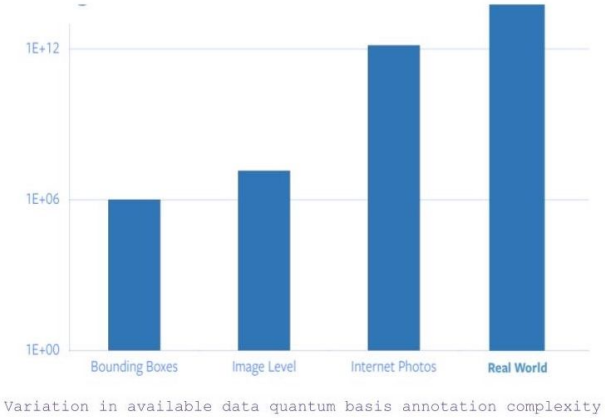


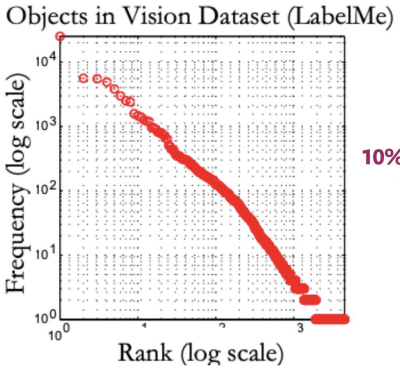
Bottom image: Vincent, Pascal, et al. "Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion." *Journal of machine learning research* 11.12 (2010).

Background

Motivation

Why do we need pretext tasks?





10% of the classes account for 93% of the data

Image (middle, right): https://atcold.github.io/pytorch-Deep-Learning/en/week10/10-1/

Deng, Jia, et al. "Imagenet: A large-scale hierarchical image database." *2009 IEEE conference on computer vision and pattern recognition*. Ieee, 2009.

Background

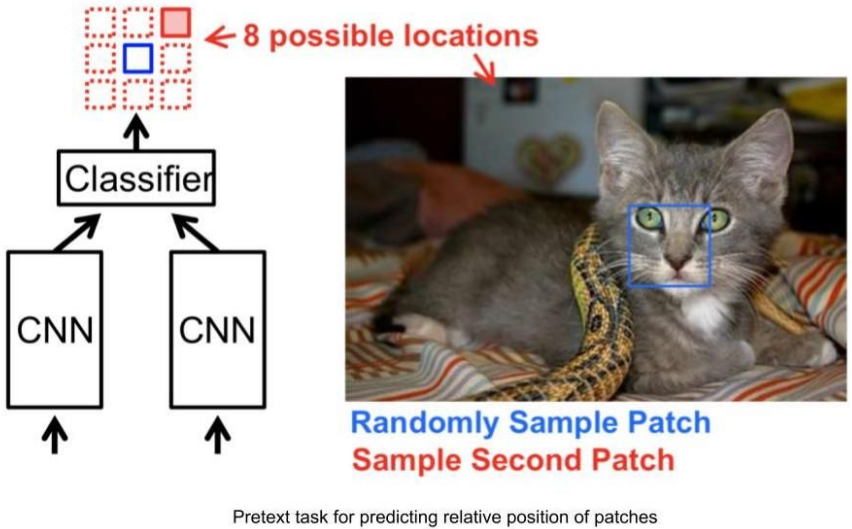
Motivation

Why do we need pretext tasks?

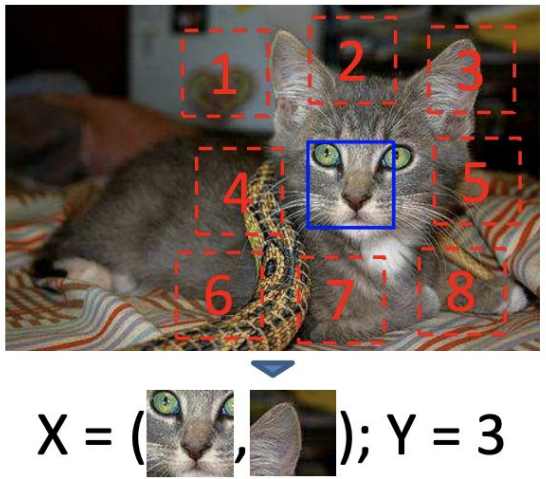
Pretext tasks ✔️

Examples of Pretext Tasks

- Jigsaw Puzzle: Predicting relative position of image patches



Pretext task for predicting relative position of patches

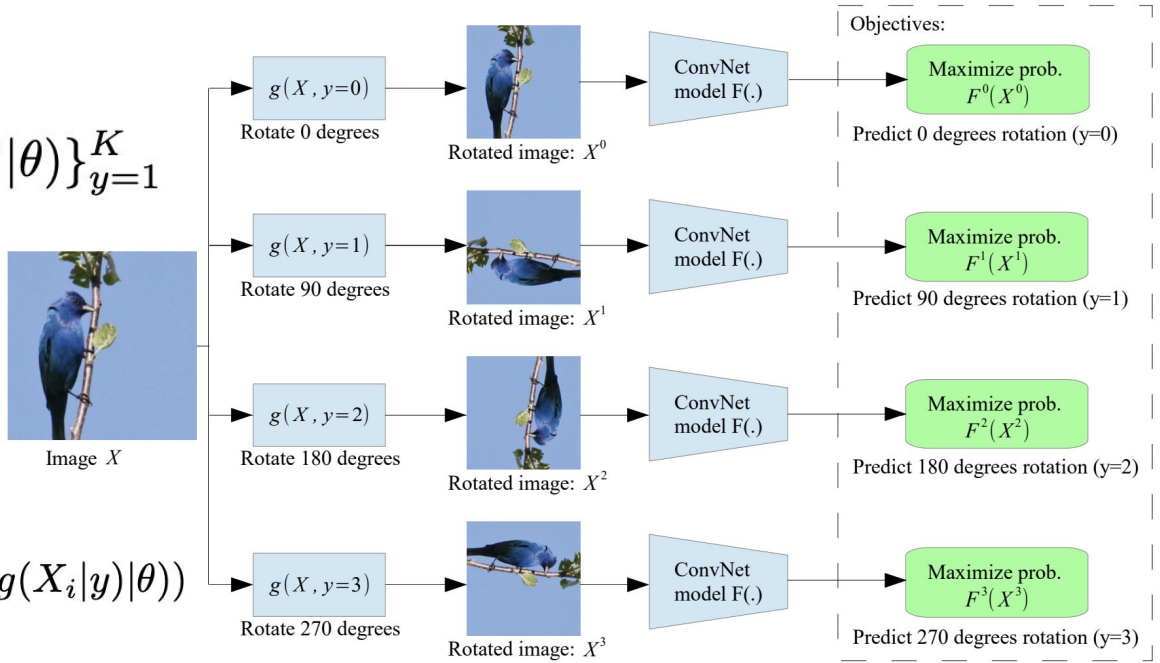


Sources: Doersch C, Gupta A, Efros A A. Unsupervised visual representation learning by context prediction[C]//Proceedings of the IEEE international conference on computer vision. 2015: 1422-1430.
https://atcold.github.io/pytorch-Deep-Learning/en/week10/10-1/

Examples of Pretext Tasks

- Rotation

$$F(X^{y*}|\theta) = \{F^y(X^{y*}|\theta)\}_{y=1}^{K}$$



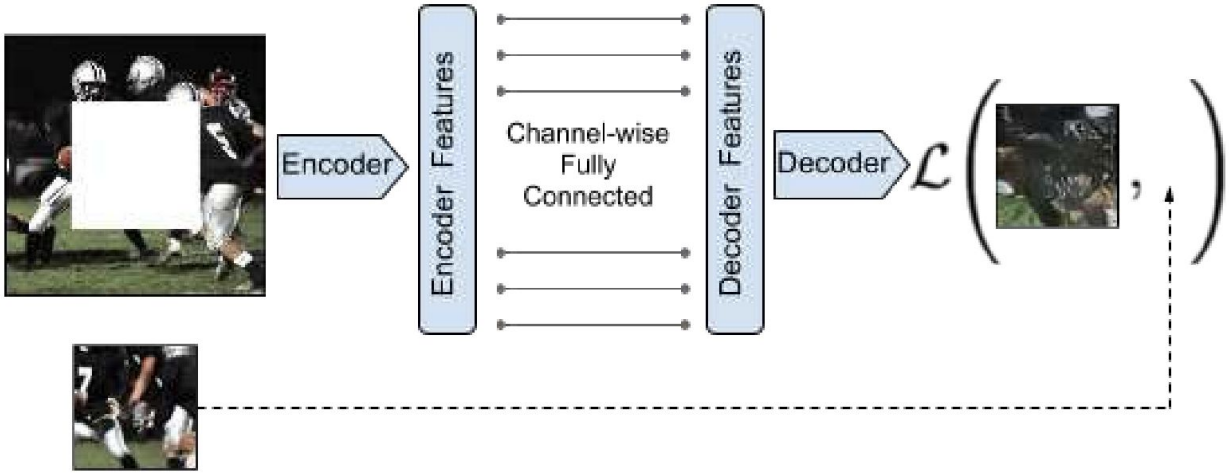
$$loss(X_i, \theta) = -\frac{1}{K}\sum_{y=1}^{K} log(F^y(g(X_i|y)|\theta))$$

Source: Gidaris S, Singh P, Komodakis N. Unsupervised representation learning by predicting image rotations[J]. arXiv preprint arXiv:1803.07728, 2018.

Examples of Pretext Tasks

- Inpainting

$$\mathcal{L}_{rec}(x) = \|\hat{M} \odot (x - F((1 - \hat{M}) \odot x))\|_2$$



Source: Pathak, Deepak, et al. "Context encoders: Feature learning by inpainting." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.

Examples of Pretext Tasks

- Super-resolution (estimating a high-resolution (HR) image from its low-resolution (LR) counterpart)

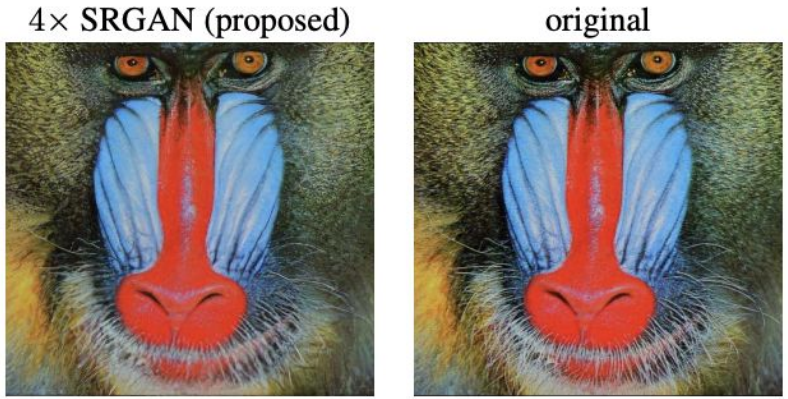


Figure 1: Super-resolved image (left) is almost indistinguishable from original (right). [4× upscaling]

$$l^{SR} = \underbrace{l^{SR}_{\mathrm{X}}}_{\text{content loss}} + \underbrace{10^{-3} l^{SR}_{Gen}}_{\text{adversarial loss}}$$

perceptual loss (for VGG based content losses)

$$l^{SR}_{MSE} = \frac{1}{r^2 WH} \sum_{x=1}^{rW} \sum_{y=1}^{rH} (I^{HR}_{x,y} - G_{\theta_G}(I^{LR})_{x,y})^2$$

$$l^{SR}_{Gen} = \sum_{n=1}^{N} -\log D_{\theta_D}(G_{\theta_G}(I^{LR}))$$

Source: Ledig, Christian, et al. "Photo-realistic single image super-resolution using a generative adversarial network." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2017.

Examples of Pretext Tasks

- Colorization

$$\mathrm{L}_2(\widehat{\mathbf{Y}}, \mathbf{Y}) = \frac{1}{2}\sum_{h,w}\|\mathbf{Y}_{h,w} - \widehat{\mathbf{Y}}_{h,w}\|_2^2 \qquad \mathbf{Y} \in \mathbb{R}^{H\times W\times 2}$$

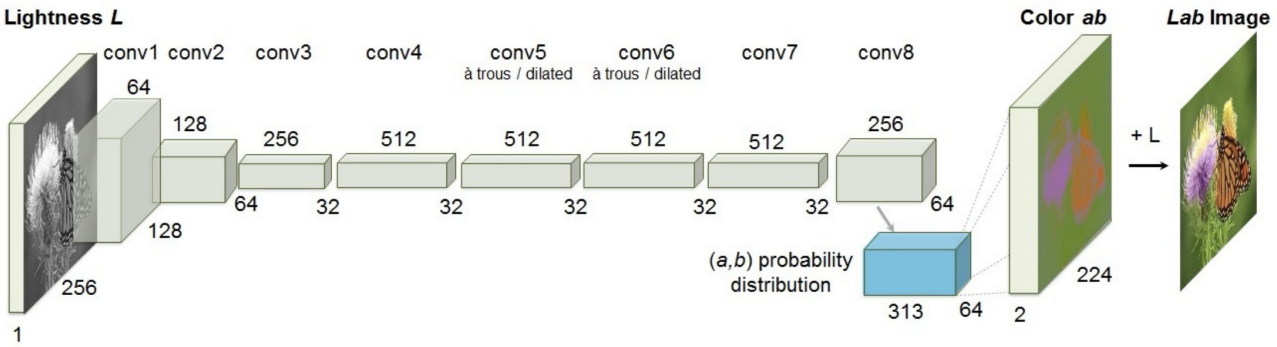


Fig. 2. Our network architecture. Each `conv` layer refers to a block of 2 or 3 repeated `conv` and `ReLU` layers, followed by a `BatchNorm` [30] layer. The net has no `pool` layers. All changes in resolution are achieved through spatial downsampling or upsampling between `conv` blocks.

$$\mathrm{L}_{cl}(\widehat{\mathbf{Z}}, \mathbf{Z}) = -\sum_{h,w} v(\mathbf{Z}_{h,w}) \sum_{q} \mathbf{Z}_{h,w,q} \log(\widehat{\mathbf{Z}}_{h,w,q})$$

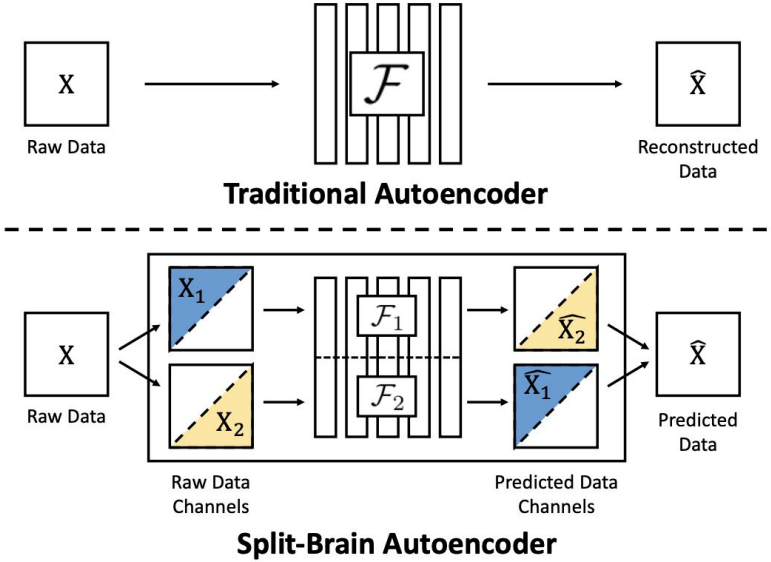
Source: Zhang, Richard, Phillip Isola, and Alexei A. Efros. "Colorful image colorization." *European conference on computer vision*. Springer, Cham, 2016.

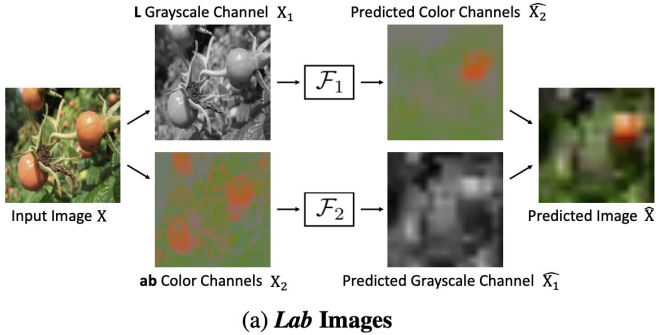
Examples of Pretext Tasks

- Split-brain

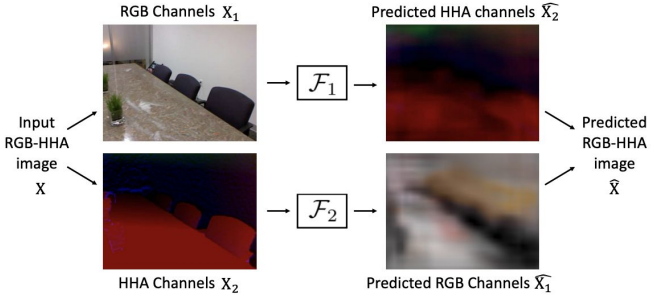
$$\mathcal{F}_1^* = \arg\min_{\mathcal{F}_1} L_1(\mathcal{F}_1(\mathbf{X_1}), \mathbf{X_2})$$

$$\mathcal{F}_2^* = \arg\min_{\mathcal{F}_2} L_2(\mathcal{F}_2(\mathbf{X_2}), \mathbf{X_1})$$





(a) ***Lab* Images**



(b) **RGB-D Images**

Source: Zhang, Richard, Phillip Isola, and Alexei A. Efros. "Split-brain autoencoders: Unsupervised learning by cross-channel prediction." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2017.

APA

Examples of Pretext Tasks

- Counting

$$\ell(\mathbf{x}) = \left|\phi(D \circ \mathbf{x}) - \sum_{j=1}^{4} \phi(T_j \circ \mathbf{x})\right|^2$$

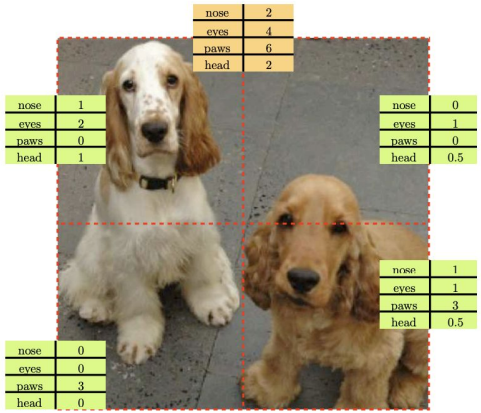
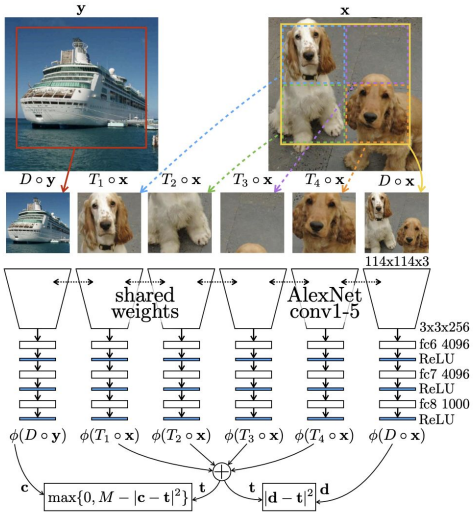


Figure 1: The number of visual primitives in the whole image should match the sum of the number of visual primitives in each tile (dashed red boxes).

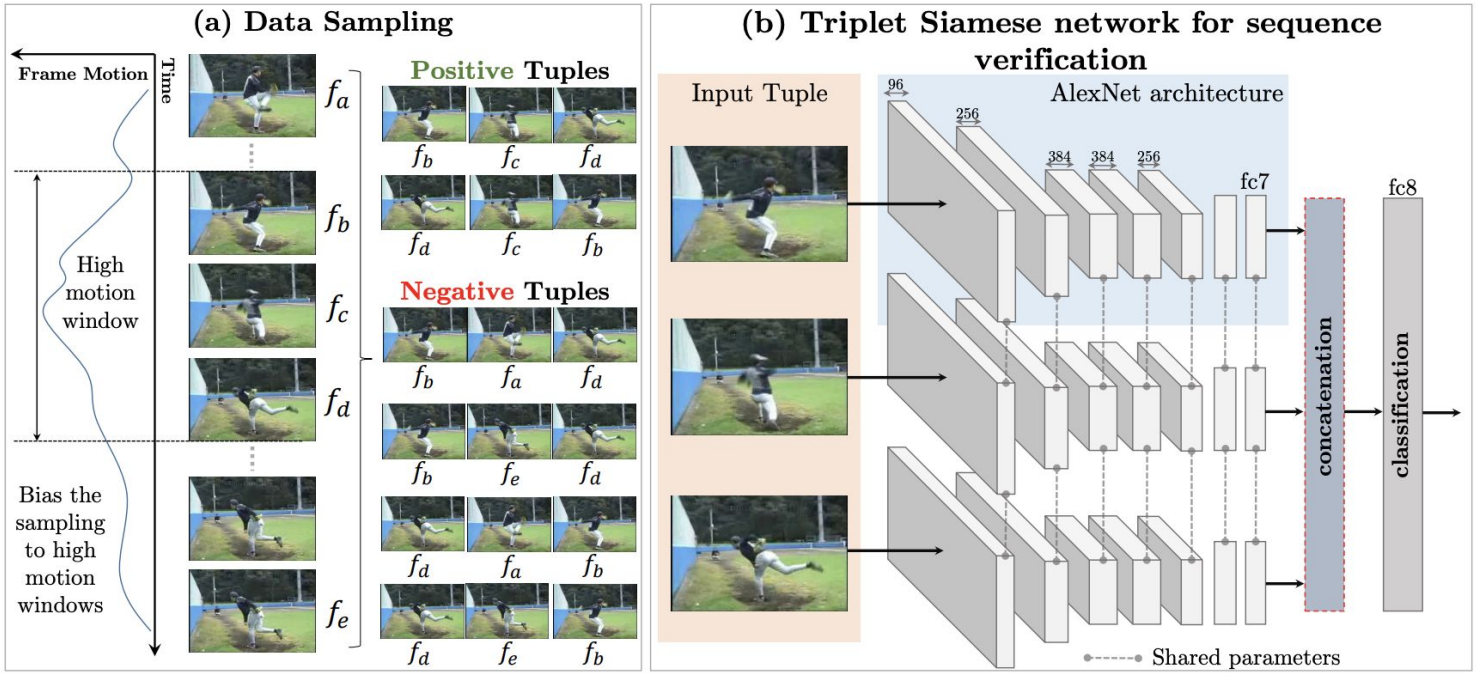


$$\ell_{\text{con}}(\mathbf{x}, \mathbf{y}) = \left|\phi(D \circ \mathbf{x}) - \sum_{j=1}^{4} \phi(T_j \circ \mathbf{x})\right|^2 + \max\left\{0, M - \left|\phi(D \circ \mathbf{y}) - \sum_{j=1}^{4} \phi(T_j \circ \mathbf{x})\right|^2\right\} \quad (4)$$

Source: Noroozi, Mehdi, Hamed Pirsiavash, and Paolo Favaro. "Representation learning by learning to count." *Proceedings of the IEEE International Conference on Computer Vision*. 2017.

Examples of Pretext Tasks for Videos

- Shuffle and learn



Source: Misra, Ishan, C. Lawrence Zitnick, and Martial Hebert. "Shuffle and learn: unsupervised learning using temporal order verification." *European Conference on Computer Vision*. Springer, Cham, 2016.

Examples of Pretext Tasks for Videos

- Videos and sound

$$E = \frac{1}{N}\sum_{n=1}^{N}(y^{(n)})||f_v(v^{(n)}) - f_a(a^{(n)})||_2^2 + (1-y^{(n)})\max(\eta - ||f_v(v^{(n)}) - f_a(a^{(n)})||_2, 0)^2$$

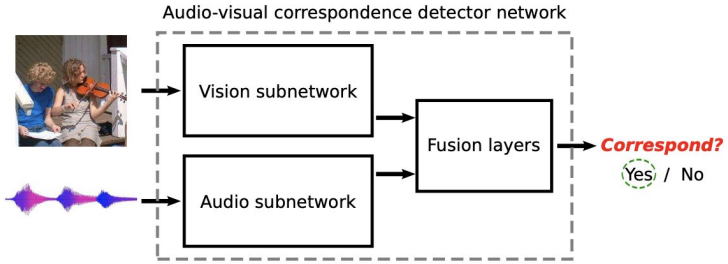
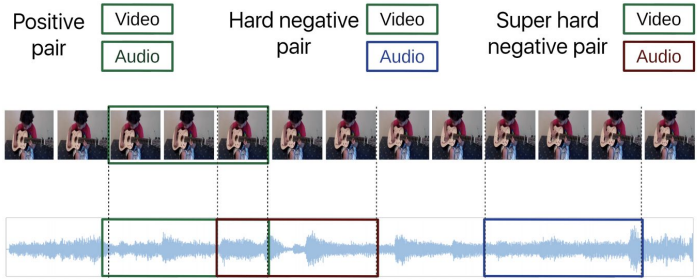


Figure 1. **Audio-visual correspondence task (AVC)**. A network should learn to determine whether a pair of (video frame, short audio clip) correspond to each other or not. Positives are (frame, audio) extracted from the same time of one video, while negatives are a frame and audio extracted from different videos.



Sources: Korbar, Bruno, Du Tran, and Lorenzo Torresani. "Cooperative learning of audio and video models from self-supervised synchronization." *arXiv preprint arXiv:1807.00230* (2018).
Arandjelovic, Relja, and Andrew Zisserman. "Look, listen and learn." *Proceedings of the IEEE International Conference on Computer Vision*. 2017.

Background

Evaluation by downstream tasks: Fine-tuning vs. Linear Classifier

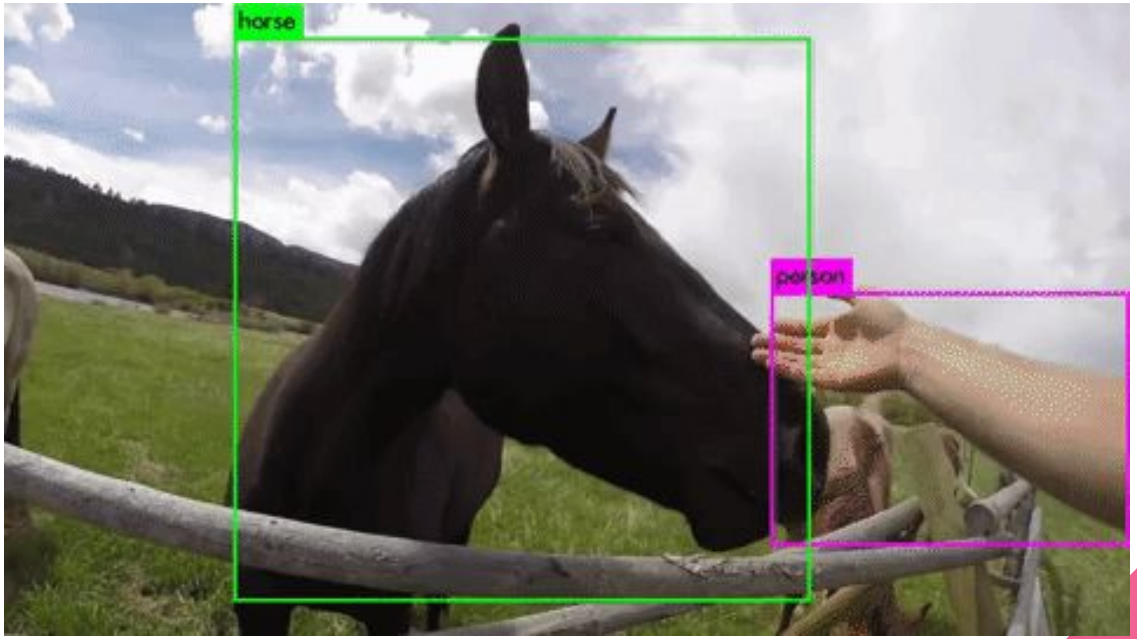
- Fine Tuning: When applying to our downstream task, we use our entire network as an initialization for which to train a new one, updating all the weights.
- Linear Classifier: On top of our pretext network, we train a small linear classifier to perform our downstream task, leaving the rest of the network intact.
- A good representation should transfer with a little training.

Evolving Losses for Unsupervised Video Representation Learning

by AJ Piergiovanni, Anelia Angelova, Michael S. Ryoo

EES598 - 012 (Winter 2021)
Presenter: Luya Gao

Motivations for Video Representation Learning



Source: “The YOLOv3 Object Detection Network Is Fast!” *Synced*, 27 Mar. 2018, syncedreview.com/2018/03/27/the-yolov3-object-detection-network-is-fast/.

Motivations for Video Representation Learning

Source: Babu, Sudharshan Chandra. “A 2019 Guide to Human Pose Estimation with Deep Learning.” *AI & Machine Learning Blog*, AI & Machine Learning Blog, 5 Aug. 2019, nanonets.com/blog/human-pose-estimation-2d-guide/.

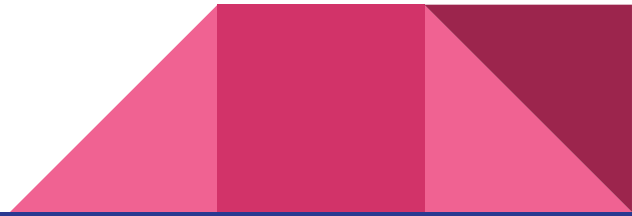
Motivations for Video Representation Learning

Source: A still from ‘Quo Vadis’ (1951).

Motivations for Pretext Tasks in Video Representation Learning

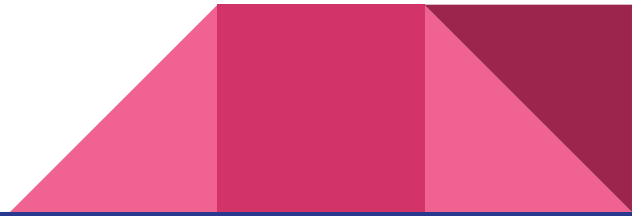
Roadblocks to collecting labeled data:

- Expensive to collect
- Expensive to label
- Sufficient quantity might not always be available



Approach

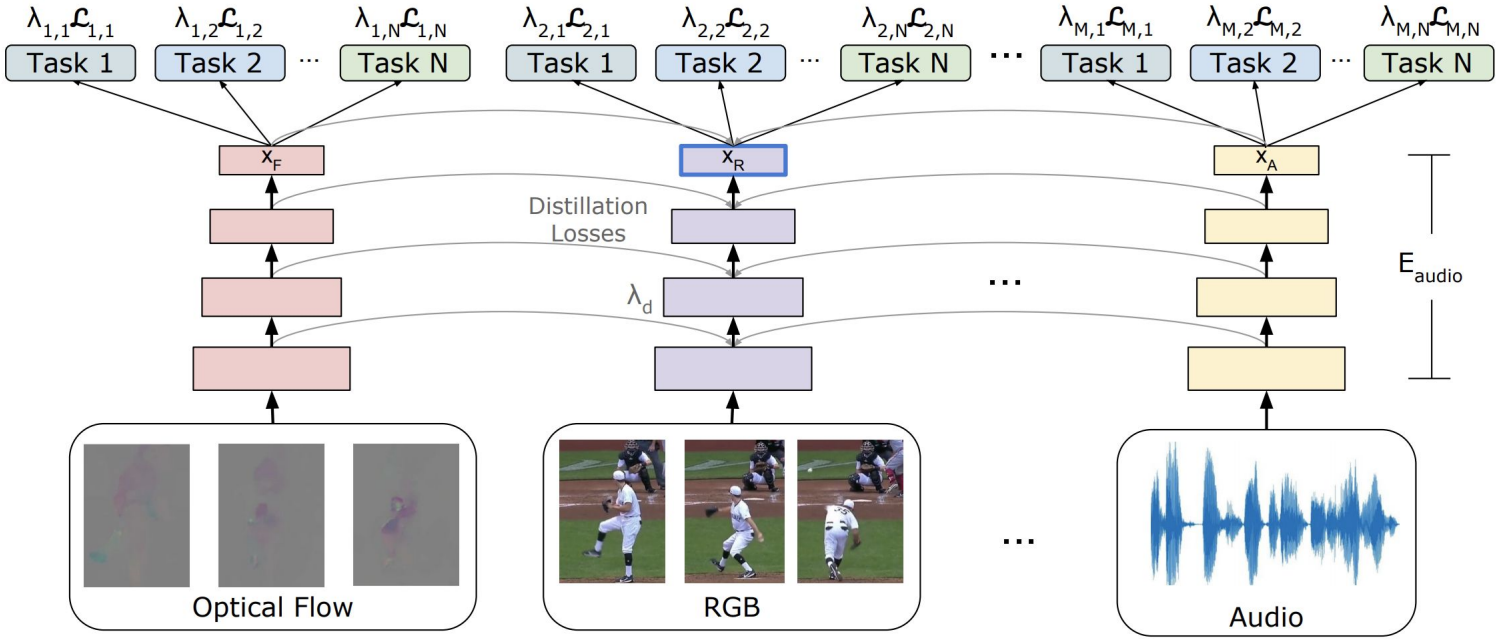
- Observation: optimized combination of multiple self-supervised tasks (fused by multi-modal distillation) is often sufficient to learn good feature representations
- Multi-modal, multi-task (single-modality + cross-modality) unsupervised learning
- Include distillation tasks to transfer features across modalities into a single-stream network



Evolutionary Algorithm

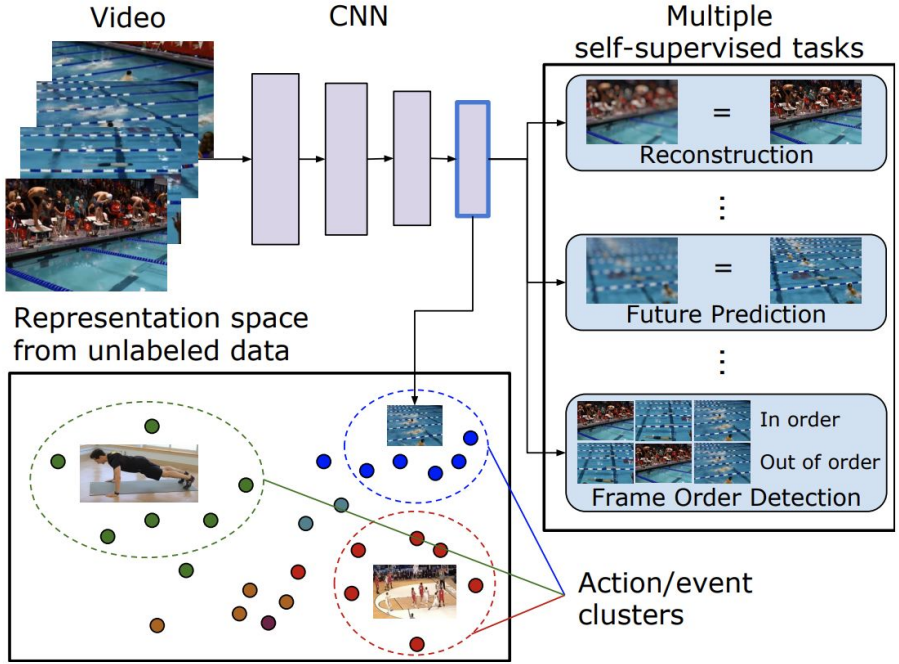
- Rationale: some tasks are more relevant to the end task than others
- Automatically evolving the main loss function
- Computationally search for the optimal combination of multi-tasks and distillation losses

Network Structure



Network Structure

$$x_m = E_m(I_m)$$



Problem statement

$L_{m,t}$ is the loss from task t and modality m

$$\mathcal{L} = \sum_{m} \sum_{t} \lambda_{m,t} \mathcal{L}_{m,t} + \sum_{d} \lambda_{d} \mathcal{L}_{d}$$

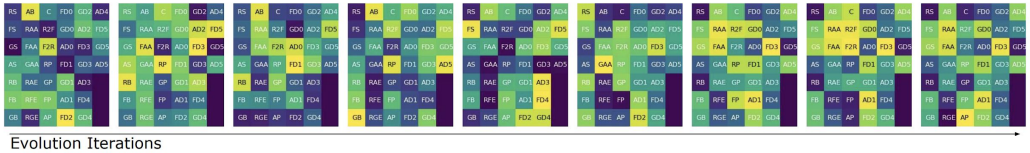


Figure 3: Evolution of the weights deciding our final loss function. Each square represents a $\lambda_{m,t}$ and how it changes over the evolutionary search. The weight symbols are as follows: the first letter is representation modality (R=RGB, A=Audio, F=Flow, G=Grey), The tasks are S=Shuffle, C=colorize, A=Audio align, P=Future prediction, B=backward detection, D=Distill, E=Embed. The numbers indicate the layer the distillation loss is applied.

Distillation

M_i : i-th layer of the main network

L_i : I-th layer of the other network

$$\mathcal{L}_d(L_i, M_i) = ||L_i - M_i||_2$$

- Distill representations jointly while training

Evolving Weighting for the Losses (ELo)

- Search space consists of all the weights of the loss function
- Fitness measure: k-means clustering on the representation learned with the corresponding loss function
- $c_i \in \mathcal{R}^D$: cluster centroid

$$p(x|c_i) = \frac{1}{\sqrt{2\sigma^2\pi}} \exp\left(-\frac{(x-c_i)^2}{2\sigma^2}\right)$$

Evolving Weighting for the Losses (ELo)

$$p(x|c_i) = \frac{1}{\sqrt{2\sigma^2\pi}} \exp\left(-\frac{(x-c_i)^2}{2\sigma^2}\right)$$

$$p(c_i|x) = \frac{p(c_i)p(x|c_i)}{\sum_j^k p(c_j)p(x|c_j)} = \frac{\exp -\frac{(x-c_i)^2}{2\sigma^2}}{\sum_{j=1}^k \exp -\frac{(x-c_j)^2}{2\sigma^2}}$$

$$= \frac{\exp -(x-c_i)^2}{\sum_{j=1}^k \exp -(x-c_j)^2}$$

Prior: constrained by Zipf distribution $q(c_i) = \frac{1/i^s}{H_{k,s}}$

$$KL(p||q) = \sum_{i=1}^{k} p(c_i) \log\left(\frac{p(c_i)}{q(c_i)}\right)$$

Evolution Strategies

- Tournament Selection: randomly pick one weight and assign it new value by uniformly sampling from [0, 1]
- CMA-ES: all the components are changed based on the fitness of all the individuals in the evolution pool

Method	Num iter.	Acc
Random Search	2000	52.4
Grid Search	2000	57.3
Tournament Selection	2000	61.4
CMA-ES	250	67.4

Table 4: Comparison of best loss found with different evolutionary strategies evaluated on HMDB.

Pretext Tasks: Reconstruction and Prediction Tasks

- Reconstruction decoder: 6 convolutional layers
- No temporal convolution
- Each modality is reconstructed (RGB, optical flow, audio)
- Cross-modality transfer tasks are also used (e.g. RGB to optical flow, flow to RGB)
- Future prediction of frames: train decoder to predict the next N frames given T frames

$$\mathcal{L}_R(\hat{I}, I) = ||\hat{I} - I||_2$$

Pretext Tasks: Temporal Ordering

- Binary Classification:
  - Ordered or shuffled?
  - Forward or backward?

$$p = Wx_m$$

$$\mathcal{L}_B(p, y) = -(y \log(p) + (1 - y) \log(1 - p))$$

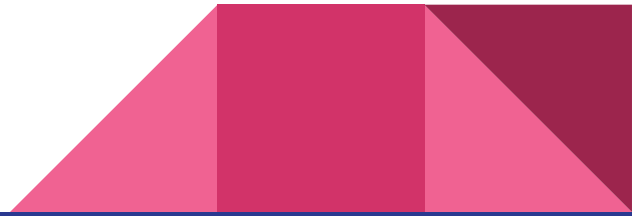
Pretext Tasks: Multi-modal Contrastive Loss

- encourages similar representations from the same video but different modalities

$$\mathcal{L}_c(x_1, x_2, x_n) = ||x_1 - x_2||_2 + \max(0, \alpha - ||x_1 - x_n||_2)$$

Pretext Tasks: Multi-modal Alignment

- Input:
  - Temporally aligned samples from two modalities
  - Sample from one modality from a temporally different region
- Make binary prediction if the two samples are temporally aligned



Datasets

Unsupervised training data: two million random, unlabeled YouTube video clips sampled randomly from the YouTube-8M dataset

Evaluation:

- HMDB: A Large Video Database for Human Motion Recognition
- UCF101: Action Recognition Data Set
- Kinetics: The Kinetics Human Action Video Dataset

Results

Method	HMDB	UCF101
Supervised		
(2+1)D ResNet-50 Scratch	35.2	63.1
(2+1)D ResNet-50 ImageNet	49.8	84.5
(2+1)D ResNet-50 Kinetics	74.3	95.1
Unsupervised		
Shuffle [26]	18.1	50.2
O3N [12]	32.5	60.3
OPN [24]	37.5	37.5
Patch [44]	-	41.5
Multisensory [29]	-	82.1
AVTS [22]	61.6	89.0
Weakly guided, HMDB		
Evolved Loss (ours)	67.8	94.1
Unsupervised		
Evolved Loss (ours, no distiliation)	53.7	84.2
Evolved Loss - ELo (ours)	67.4	93.8

Table 2: Comparison to the state-of-art on HMDB51 and UCF101. Note that previous approaches train on activity recognition datasets (e.g., Kinetics) are much more aligned to the final task, whereas we use random video clips. Even using more difficult data, we outperform the previous methods. (The top portion shows results for (2+1)D ResNet-50 with supervised pretraining as in Table 1).

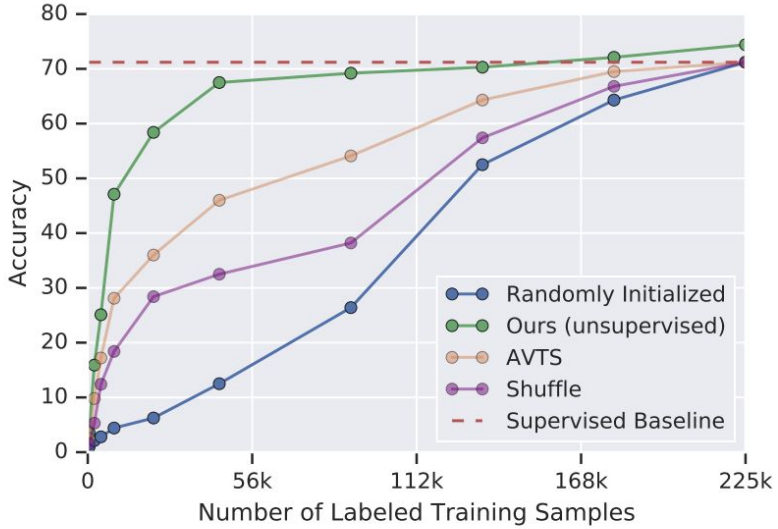


Figure 5: How much labeled, supervised data is needed once the unsupervised representation is learned. We achieve comparable performance with roughly half the data and outperform the supervised baselines using the entire dataset.

Results

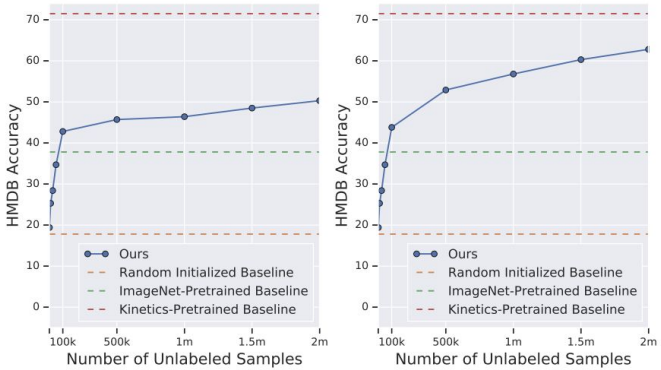
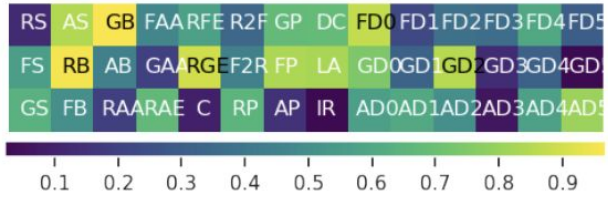


Figure 6: Comparisons of different amounts of unsupervised data. **Left**: Total number of training iterations fixed (i.e., less epochs as data is added). **Right**: Total number of epochs fixed (i.e., more iterations as more data is added). We observe that adding more data without increasing training time improves performance, while training longer on more data is better. On HMDB.



Conclusion

- Unified framework for multi-task, multimodal unsupervised video representation learning
- Loss function evolution: automatically find the weights of tasks (with unsupervised fitness measure)
- Improved end task (e.g. activity recognition) performance:
  - Outperform prior self-supervised learning
  - Match or outperform supervised learning

References

- “The YOLOv3 Object Detection Network Is Fast!” *Synced*, 27 Mar. 2018, syncedreview.com/2018/03/27/the-yolov3-object-detection-network-is-fast/.
- Babu, Sudharshan Chandra. “A 2019 Guide to Human Pose Estimation with Deep Learning.” *AI & Machine Learning Blog*, AI & Machine Learning Blog, 5 Aug. 2019, nanonets.com/blog/human-pose-estimation-2d-guide/.
- Piergiovanni, A. J., Anelia Angelova, and Michael S. Ryoo. "Evolving losses for unsupervised video representation learning." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020.

Are Labels Necessary for Neural Architecture Search?

By Chenxi Liu , Piotr Doll´ar, Kaiming He, Ross Girshick, Alan Yuille, and Saining Xie

Presenter: Yuliang Zhu

Neural Architecture Search (NAS)

Motivation:

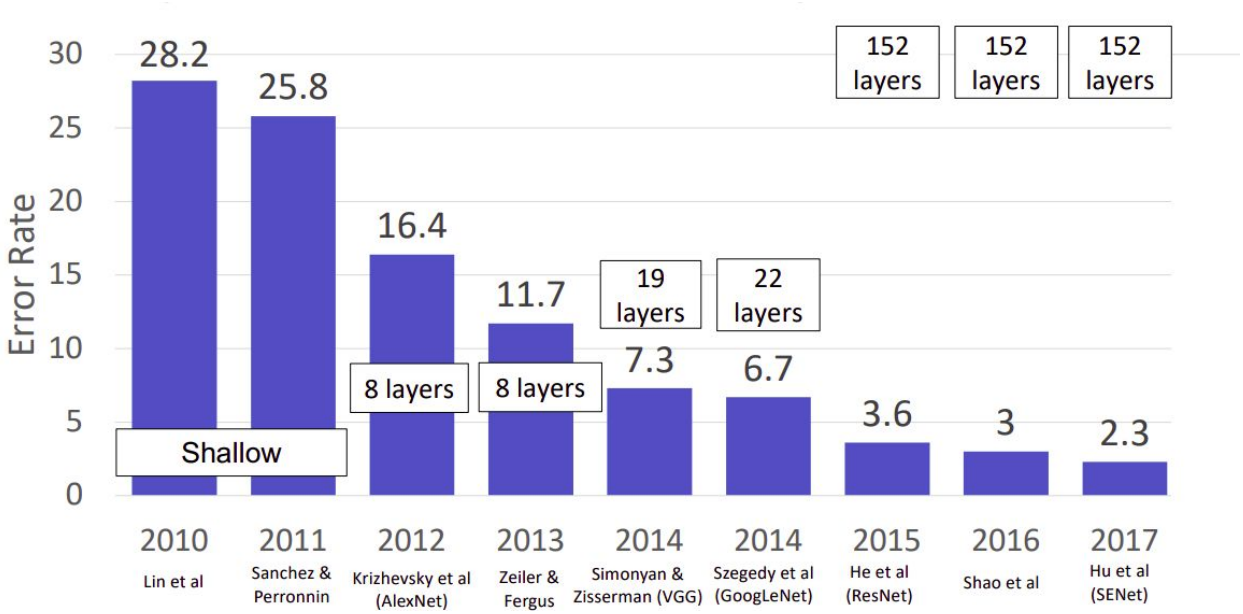


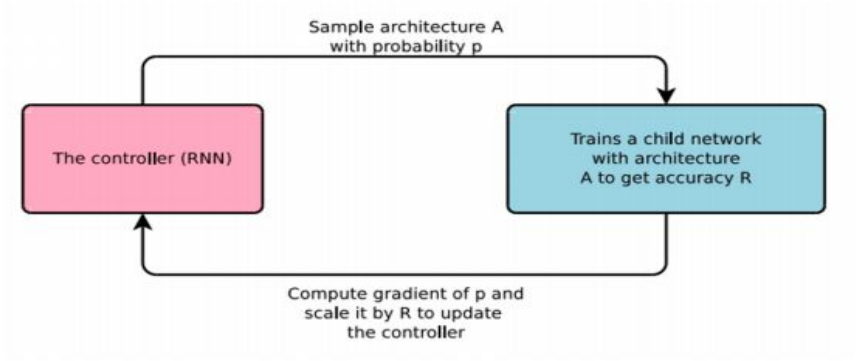
Image source: https://web.eecs.umich.edu/~justincj/slides/eecs498/FA2020/598_FA2020_lecture08.pdf

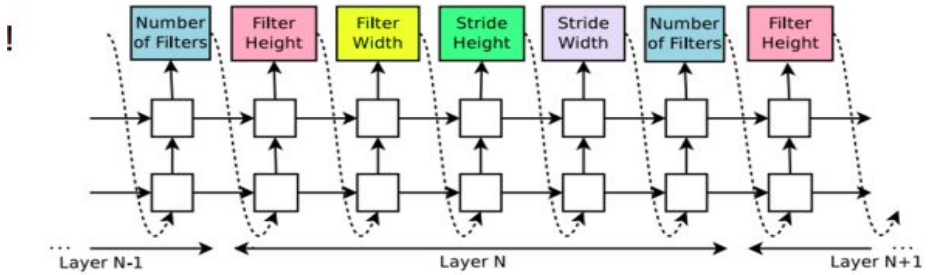
Neural Architecture Search (NAS)

Definition:

Image source: https://towardsdatascience.com/intuitive-explanation-of-differentiable-architecture-search-darts-692bdadcc69c

Neural Architecture Search (NAS)





800 GPUs

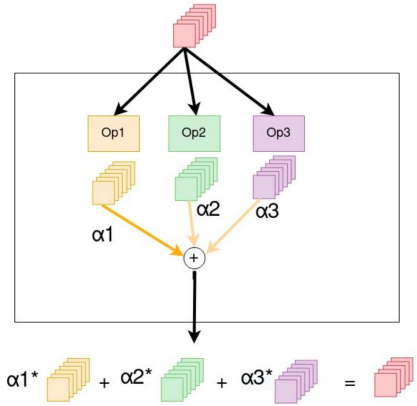
28 days

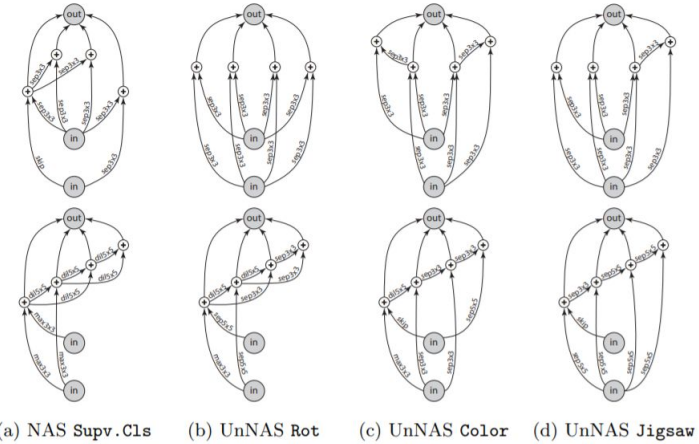
Zoph, Barret, and Quoc V. Le. "Neural architecture search with reinforcement learning." *arXiv preprint arXiv:1611.01578* (2016).

Neural Architecture Search (NAS)

- DARTS: Differentiable Architecture Search
- 3,000 GPU days -> 2-3 GPU days
- O = {conv_3x3, max_pool_3x3, dilated_conv_5x5}

$$\bar{o}^{(i,j)}(x) = \sum_{o \in \mathcal{O}} \frac{\exp(\alpha_o^{(i,j)})}{\sum_{o' \in \mathcal{O}} \exp(\alpha_{o'}^{(i,j)})} o(x)$$





(a) NAS Supv.Cls (b) UnNAS Rot (c) UnNAS Color (d) UnNAS Jigsaw

Liu, Hanxiao, Karen Simonyan, and Yiming Yang. "Darts: Differentiable architecture search." *arXiv preprint arXiv:1806.09055* (2018).

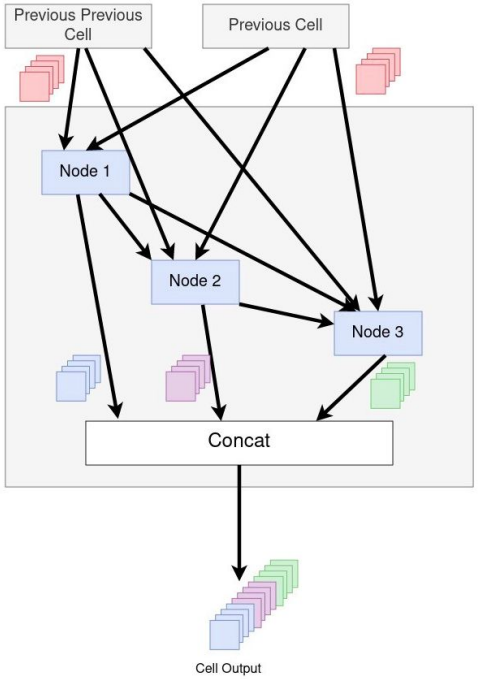
Neural Architecture Search (NAS)

DARTS: Differentiable Architecture Search

$$x^{(j)} = \sum_{i<j} o^{(i,j)}(x^{(i)})$$

$$o^{(i,j)} = \text{argmax}_{o \in \mathcal{O}} \ \alpha_o^{(i,j)}$$

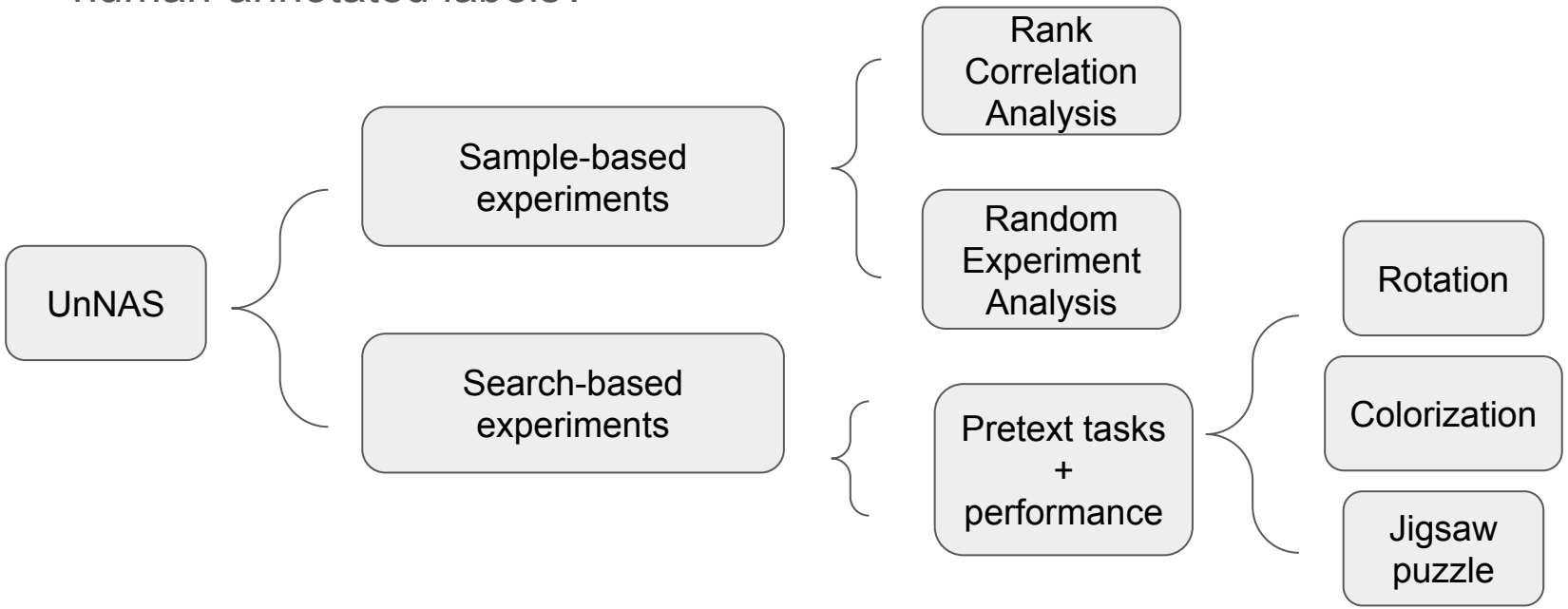
$$\begin{aligned} \min_{\alpha} \quad & \mathcal{L}_{val}(w^*(\alpha), \alpha) \\ \text{s.t.} \quad & w^*(\alpha) = \text{argmin}_w \ \mathcal{L}_{train}(w, \alpha) \end{aligned}$$



Liu, Hanxiao, Karen Simonyan, and Yiming Yang. "Darts: Differentiable architecture search." *arXiv preprint arXiv:1806.09055* (2018).

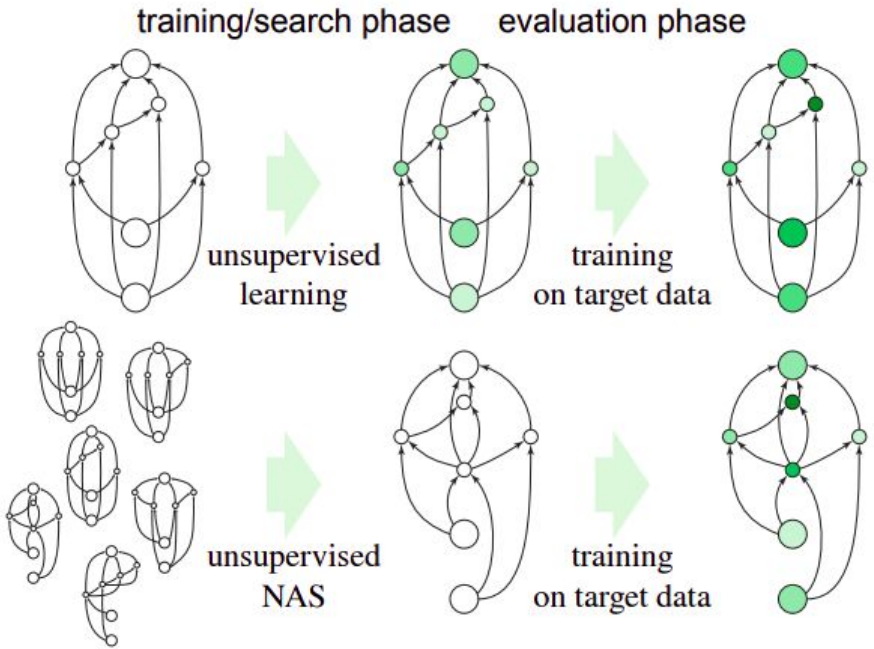
Problem Statement

- Are Labels Necessary for Neural Architecture Search?
- Can we find high-quality neural architectures using only images, but no human-annotated labels?



Approach

UnNAS:



Analogy to unsupervised learning

Sample-Based Experiments

rank correlation analysis

Spearman's Rank Correlation r

Store	Distance	Price
1	50	1.8
2	170	1.25
3	300	0.8
...	...	...

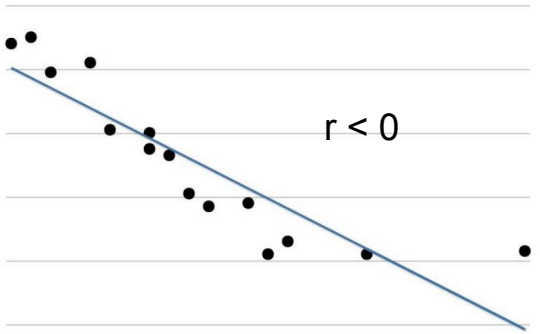
$$r_s = \rho_{\mathrm{rg}_X, \mathrm{rg}_Y} = \frac{\mathrm{cov}(\mathrm{rg}_X, \mathrm{rg}_Y)}{\sigma_{\mathrm{rg}_X} \sigma_{\mathrm{rg}_Y}}$$

$$r_s = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)}$$

Derivation:

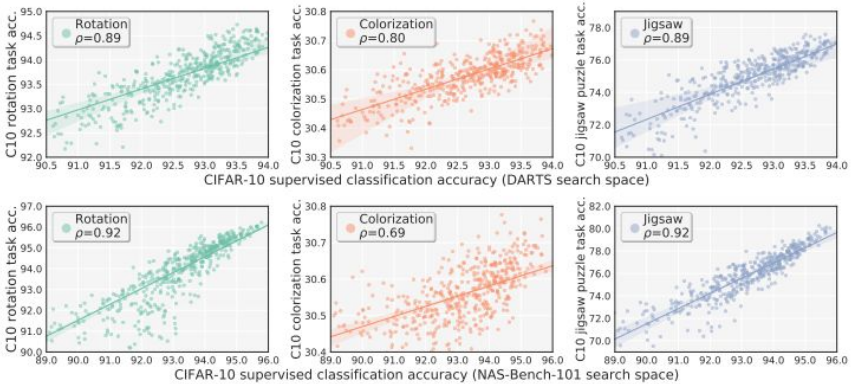
$$\sum_i (x_i - \bar{x})^2 = \frac{n(n^2-1)}{12}$$

$$\sum_i (x_i - \bar{x})(y_i - \bar{y}) = \frac{n(n^2-1)}{12} - \sum d_i^2/2$$

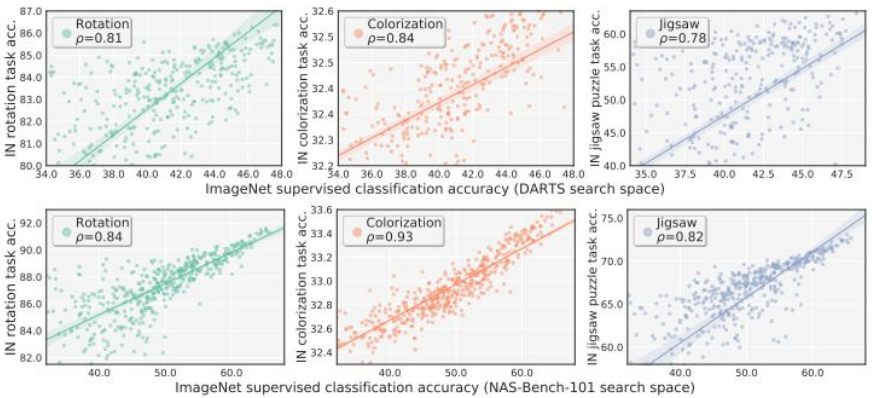


Sample-Based Experiments

Results



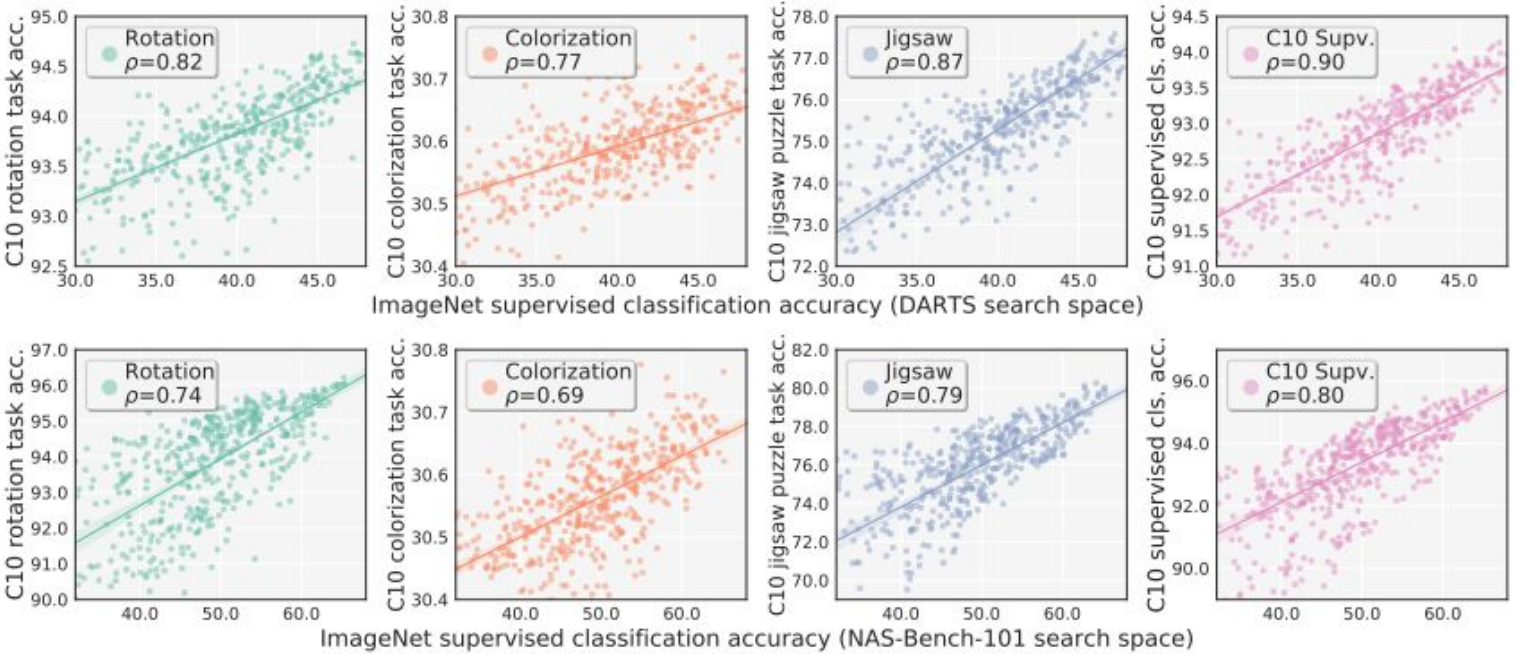
CIFAR10



ImageNet

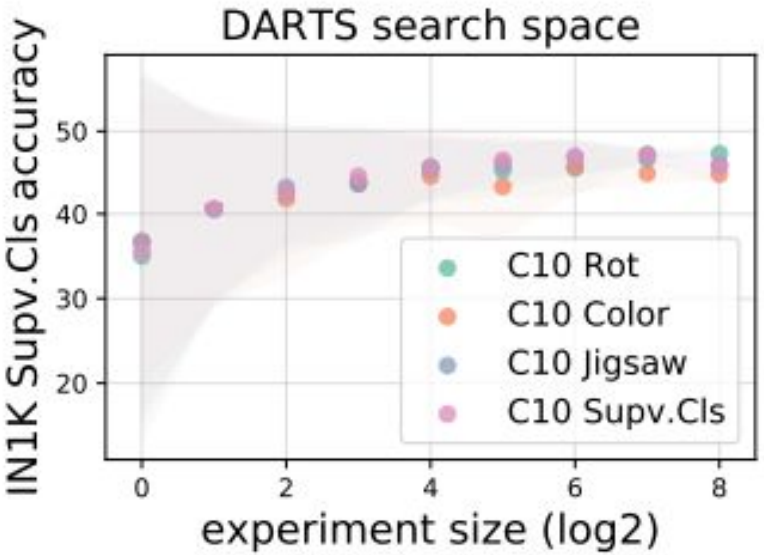
Sample-Based Experiments

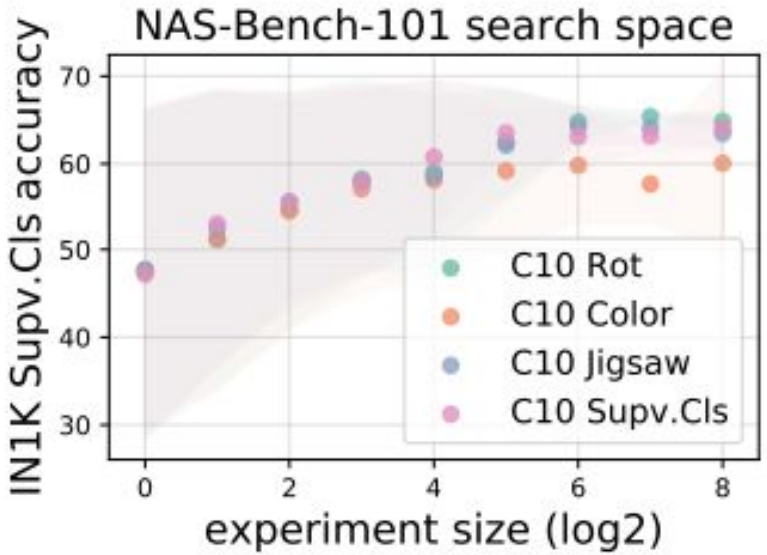
Rank correlation between supervised accuracy and pretext accuracy across datasets



Sample-Based Experiments

Random experiment efficiency curves





Search-Based Experiments

method	search dataset & task	top-1 acc.	FLOPs (M)	params (M)
NAS-DARTS [20]	CIFAR-10 Supv.Cls	73.3	574	4.7
NAS-P-DARTS [6]	CIFAR-10 Supv.Cls	75.6	557	4.9
NAS-PC-DARTS [33]	CIFAR-10 Supv.Cls	74.9	586	5.3
NAS-PC-DARTS [33]	IN1K Supv.Cls	75.8	597	5.3
NAS-DARTS†	CIFAR-10 Supv.Cls	$74.9_{\pm 0.08}$	538	4.7
NAS-DARTS	IN1K Supv.Cls	$76.3_{\pm 0.06}$	590	5.3
UnNAS-DARTS	IN1K Rot	$75.8_{\pm 0.18}$	558	5.1
UnNAS-DARTS	IN1K Color	$75.7_{\pm 0.12}$	547	4.9
UnNAS-DARTS	IN1K Jigsaw	$75.9_{\pm 0.15}$	567	5.2
NAS-DARTS	IN22K Supv.Cls	$75.9_{\pm 0.09}$	585	5.2
UnNAS-DARTS	IN22K Rot	$75.7_{\pm 0.23}$	549	5.0
UnNAS-DARTS	IN22K Color	$75.9_{\pm 0.21}$	547	5.0
UnNAS-DARTS	IN22K Jigsaw	$75.9_{\pm 0.31}$	559	5.1
NAS-DARTS	Cityscapes Supv.Seg	$75.8_{\pm 0.13}$	566	5.1
UnNAS-DARTS	Cityscapes Rot	$75.9_{\pm 0.19}$	554	5.1
UnNAS-DARTS	Cityscapes Color	$75.2_{\pm 0.15}$	594	5.1
UnNAS-DARTS	Cityscapes Jigsaw	$75.5_{\pm 0.06}$	566	5.0

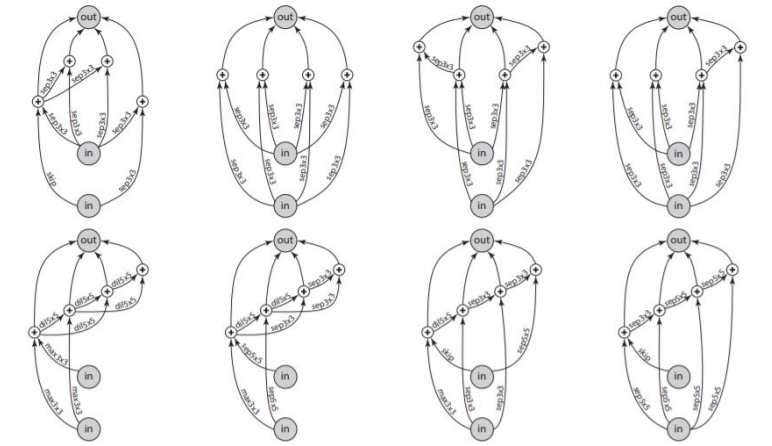
Table 1: **ImageNet-1K classification results of the architectures searched by NAS and UnNAS algorithms**. Rows in gray correspond to invalid UnNAS configurations where the search and evaluation datasets are the same. † is our training result of the DARTS architecture released in [20].

Search-Based Experiments

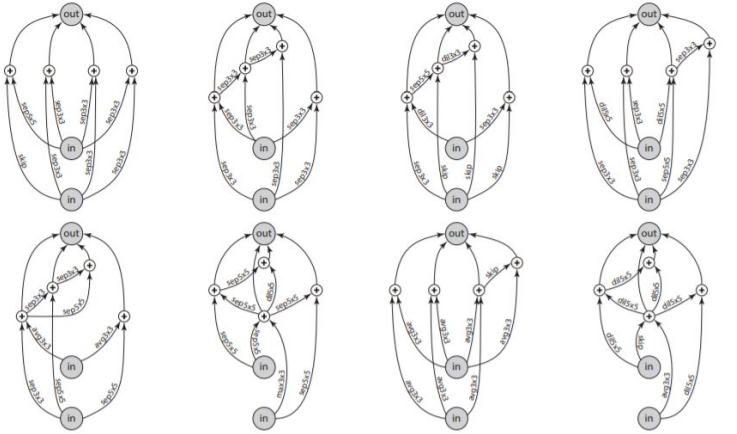
method	search dataset & task	mIoU	FLOPs (B)	params (M)
NAS-DARTS†	CIFAR-10 Supv.Cls	$72.6_{\pm 0.55}$	121	9.6
NAS-DARTS	IN1K Supv.Cls	$73.6_{\pm 0.31}$	127	10.2
UnNAS-DARTS	IN1K Rot	$73.6_{\pm 0.29}$	129	10.4
UnNAS-DARTS	IN1K Color	$72.2_{\pm 0.56}$	122	9.7
UnNAS-DARTS	IN1K Jigsaw	$73.1_{\pm 0.17}$	129	10.4
NAS-DARTS	IN22K Supv.Cls	$72.4_{\pm 0.29}$	126	10.1
UnNAS-DARTS	IN22K Rot	$72.9_{\pm 0.23}$	128	10.3
UnNAS-DARTS	IN22K Color	$73.6_{\pm 0.41}$	128	10.3
UnNAS-DARTS	IN22K Jigsaw	$73.1_{\pm 0.59}$	129	10.4
NAS-DARTS	Cityscapes Supv.Seg	$72.4_{\pm 0.15}$	128	10.3
UnNAS-DARTS	Cityscapes Rot	$73.0_{\pm 0.25}$	128	10.3
UnNAS-DARTS	Cityscapes Color	$72.5_{\pm 0.31}$	122	9.5
UnNAS-DARTS	**Cityscapes Jigsaw**	$\mathbf{74.1}_{\pm 0.39}$	**128**	**10.2**

Table 2: **Cityscapes semantic segmentation results of the architectures searched by NAS and UnNAS algorithms**. These are trained from scratch: there is no fine-tuning from ImageNet checkpoint. Rows in gray correspond to an illegitimate setup where the search dataset is the same as the evaluation dataset. † is our training result of the DARTS architecture released in [20].

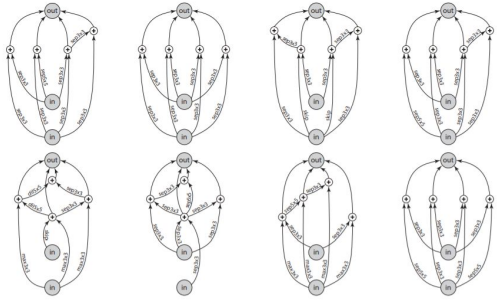
Conclusion



(a) NAS Supv.Cls (b) UnNAS Rot (c) UnNAS Color (d) UnNAS Jigsaw



(a) NAS Supv.Seg (b) UnNAS Rot (c) UnNAS Color (d) UnNAS Jigsaw



(a) NAS Supv.Cls (b) UnNAS Rot (c) UnNAS Color (d) UnNAS Jigsaw

Discussion Questions

- Aside from the pretext tasks mentioned today, can you think of any other pretext tasks that might have helped in any of the unsupervised learning settings?
- Distillation can be computationally expensive: is it worth the computational resources to search the whole weighting space? Or is there a better way?
- If not label and pretext tasks, what is important for a good neural network design? (e.g. image statistics)

Thank you listening!