JOHNS HOPKINS
WHITING SCHOOL
of ENGINEERING

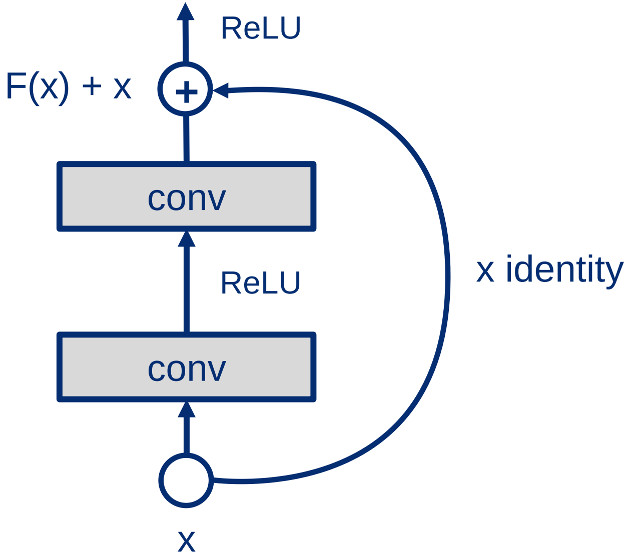
EN.601.482/682 Deep Learning

Unsupervised and Self-supervised Learning

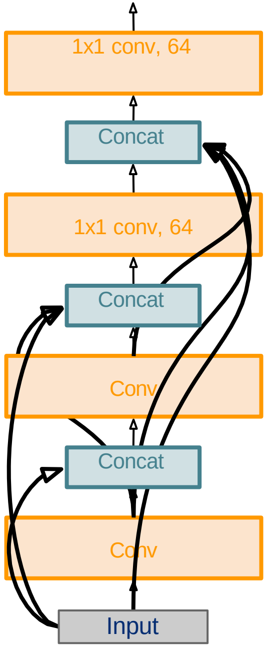
Mathias Unberath, PhD
Assistant Professor
Dept of Computer Science
Johns Hopkins University

Reminder

Gradient flow



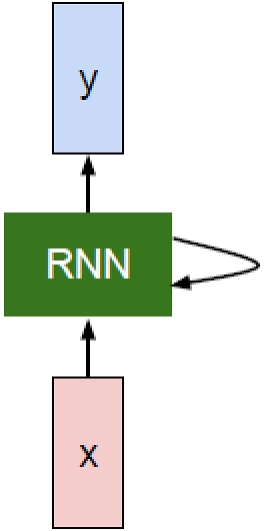
Residual block



Dense block

Reminder

Recurrent Neural Networks: Standard Architecture



$$h_t = f_W(h_{t-1}, x_t)$$

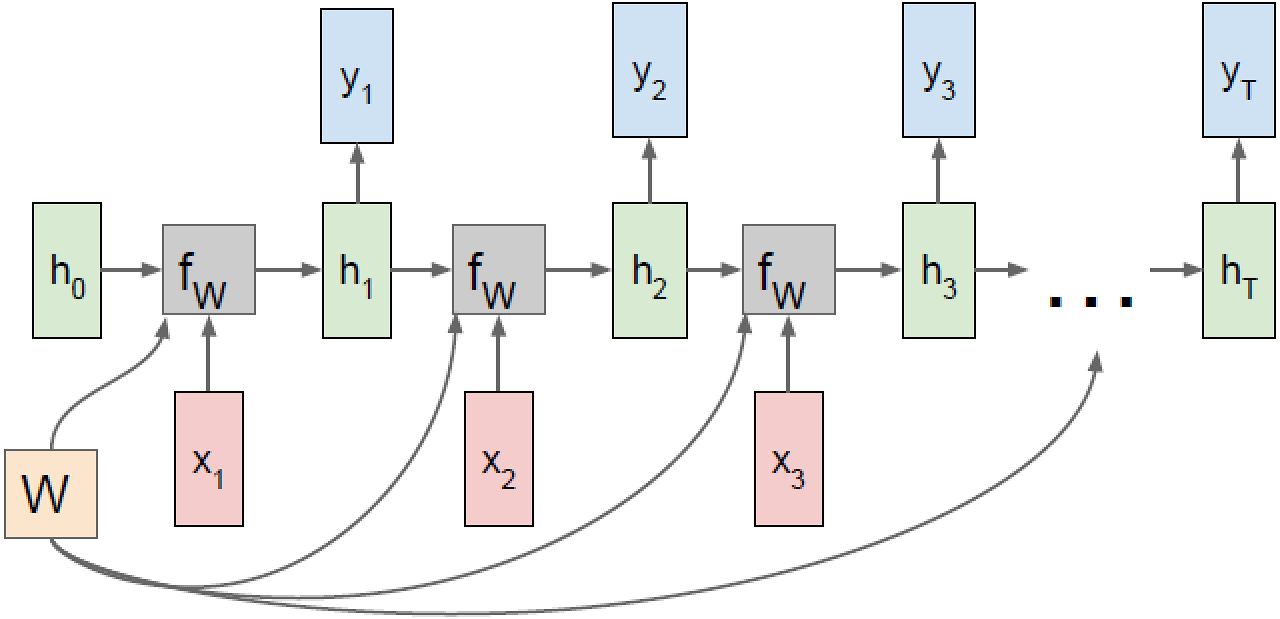
$$\downarrow$$

$$h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t)$$

$$y_t = W_{hy}h_t$$

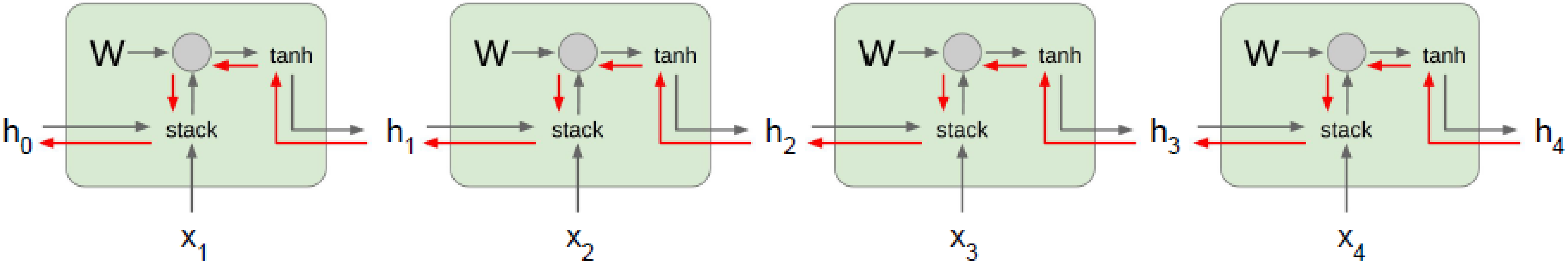
Reminder

Recurrent Neural Networks: Unrolling → Same weights at every timestep!



Reminder

Recurrent Neural Networks: Gradient flow



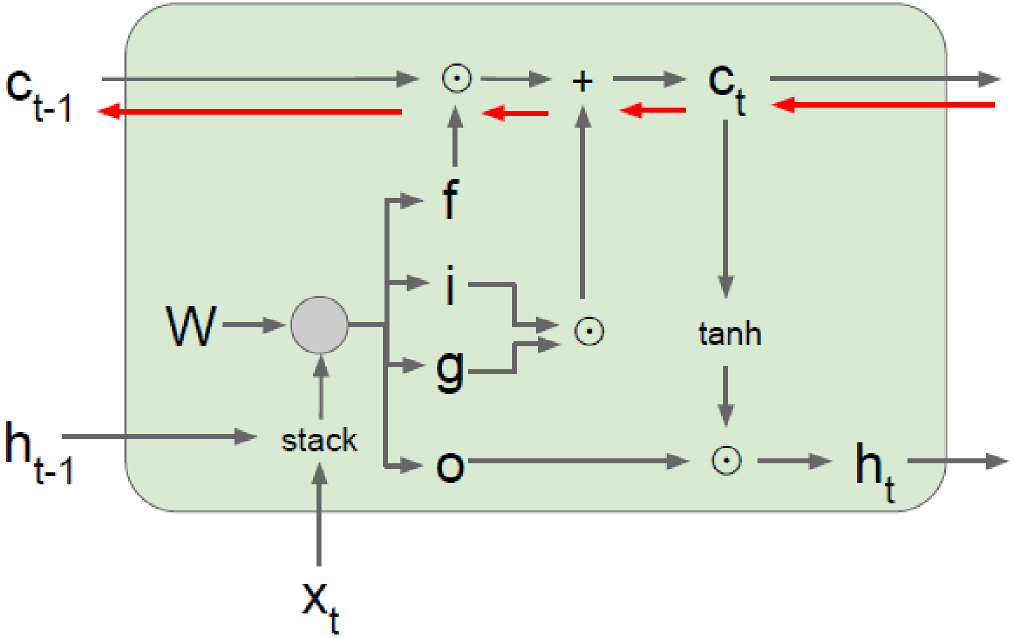
Computing gradient of h_0 involves many factors of W (and repeated tanh)

Largest singular value > 1:
Exploding gradients

Largest singular value < 1:
Vanishing gradients

Reminder

Long Short Term Memory: Mitigating the gradient problem



$$\begin{pmatrix} i \\ f \\ o \\ g \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{pmatrix} W \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix}$$

$$c_t = f \odot c_{t-1} + i \odot g$$

$$h_t = o \odot \tanh(c_t)$$

Our Deep Learning Success Story So Far

1. Download or collect large database

2. Design or download state-of-the-art ConvNet architecture

3. Define the desired output targets via loss function (e.g. DICE loss, …)

4. Optimize parameters with respect to loss

5. ??

Running example: Image classification

6. Profit

Our Deep Learning Success Story So Far

Running example: Image classification in computer vision

1. **Download** or collect large database: ImageNet

2. Design or download state-of-the-art ConvNet architecture: ResNet

3. Define the desired output targets via loss function (e.g. DICE loss, …): CE

4. Optimize parameters with respect to loss: ADAM

Our Deep Learning Success Story So Far

Running example: Image classification in computer vision, but new categories!

1. Download or **collect** large database: Proprietary images
2. Design or download state-of-the-art ConvNet architecture: ResNet
3. Define the desired output targets via loss function (e.g. DICE loss, …): CE
4. Optimize parameters with respect to loss: ADAM

Q: Which assumption are we making?

Our Deep Learning Success Story So Far

Running example: Image classification in computer vision, but new categories!

1. Download or **collect** large database: Proprietary images
2. Design or download state-of-the-art ConvNet architecture: ResNet
3. Define the desired output targets via loss function (e.g. DICE loss, …): CE
4. Optimize parameters with respect to loss: ADAM

We need ground truth labels for every single image to be able to compute such loss.
→ Someone will need to annotate every single image

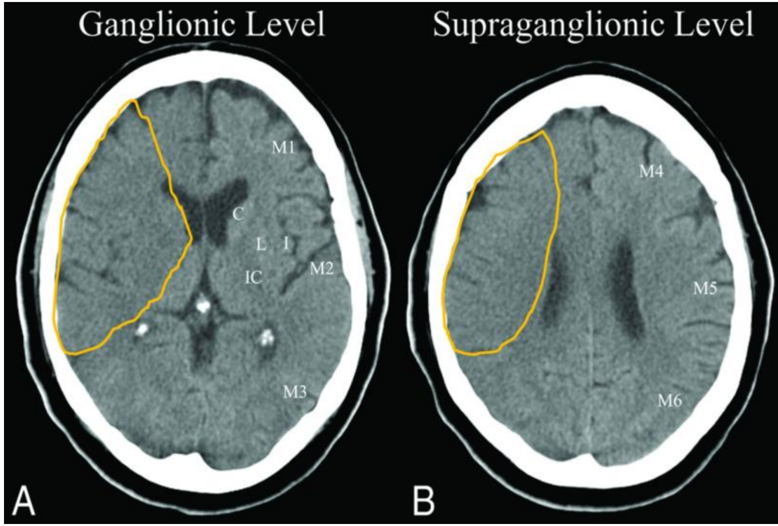
A Bigger Picture

Learning from annotated instances: Supervised learning

Obtaining annotations is among the most critical steps! Why?

Example: Ischemic stroke lesion identification in no-contrast CT (ASPECT scoring)

- Expert-level domain knowledge
- High variation (much data!)
- High dimensional (volumetric labels!)



Aviv, R. I., et al (2007). Alberta Stroke Program Early CT Scoring of CT perfusion in early stroke visualization and assessment. *Am J Neurorad*, *28*(10), 1975-1980.

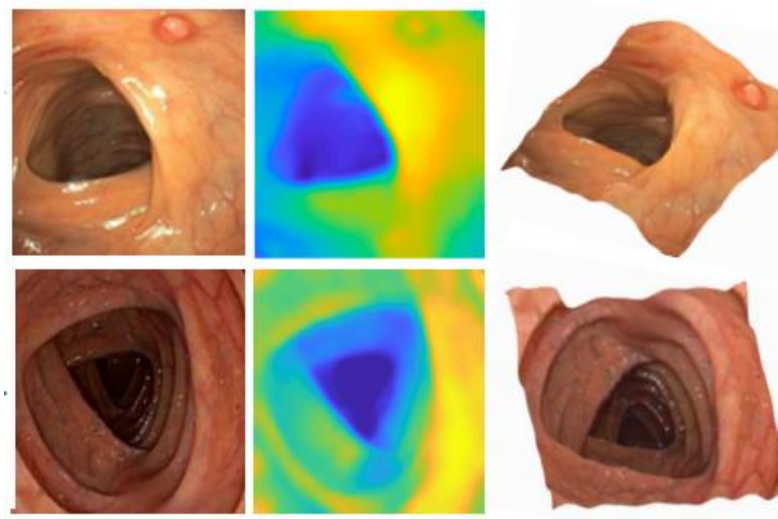
A Bigger Picture

Learning from annotated instances: Supervised learning

Obtaining annotations is among the most critical steps! Why?

Example: Monocular depth prediction from endoscopic video

- "True ground truth": Virtually impossible! Would require stereo or RGBD endoscopes
- High variation (much data!)



Mahmood, F., Chen, R., & Durr, N. J. (2018). Unsupervised reverse domain adaptation for synthetic medical images via adversarial training. IEEE TMI, 37(12), 2572-2581.

Today’s Lecture

If acquiring data is relatively easy …

… but annotating data is comparably difficult, …

… wouldn’t it be appealing if we could just learn from data?

Today's Lecture

If acquiring data is relatively easy …

… but annotating data is comparably difficult, …

… wouldn't it be appealing if we could just learn from data?

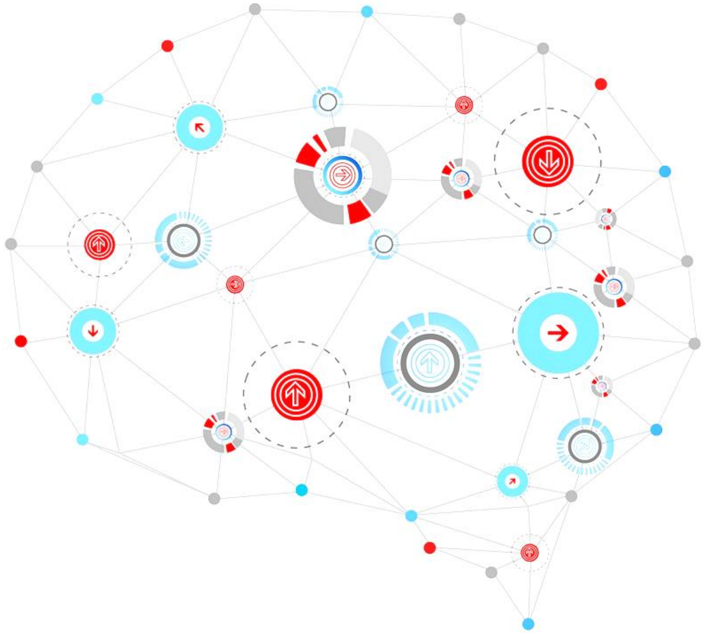
This is known as unsupervised and self-supervised learning!

Research in this area is rapidly growing!

→ Training is based on back-propagation of gradients

→ How can we define (un)happiness with the current parameters?

Today’s Lecture



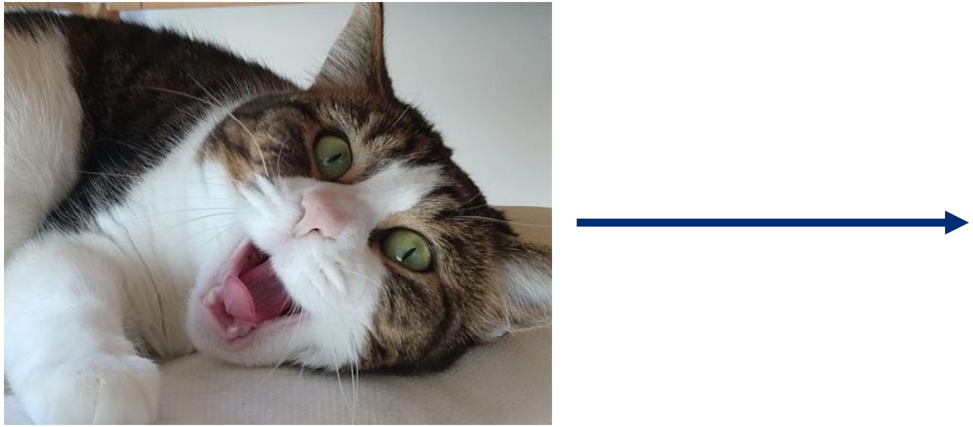
Unsupervised and self-supervised learning

Unsupervised and Self-supervised Learning

ImageNet Classification Without Labels

Image Classification

Assume set of discrete labels is known, e.g. {dog, cat, car, plane, …}
Classification describes a mapping of image onto label



cat

This is my cat.

Image Classification: A Mathematical Perspective

$$\min_{\theta,W} \frac{1}{N} \sum_i L\left(g_W\left(f_\theta \boxed{(x_i)}\right), \boxed{y_i}\right)$$

- Input x_i with label y_i

Caron, M., Bojanowski, P., Joulin, A., & Douze, M. (2018). Deep clustering for unsupervised learning of visual features. ECCV (pp. 132-149).

Image Classification: A Mathematical Perspective

$$\min_{\theta,W} \frac{1}{N} \sum_i L\left(g_W\left(\boxed{f_\theta\boxed{(x_i)}}\right), \boxed{y_i}\right)$$

- Input x_i with label y_i
- f_Θ computes high-level representations of x_i

Caron, M., Bojanowski, P., Joulin, A., & Douze, M. (2018). Deep clustering for unsupervised learning of visual features. ECCV (pp. 132-149).

Image Classification: A Mathematical Perspective

$$\min_{\theta,W} \frac{1}{N} \sum_i L\left(\boxed{g_W\left(\boxed{f_\theta\boxed{(x_i)}}\right)}, \boxed{y_i}\right)$$

- Input x_i with label y_i
- f_Θ computes high-level representations of x_i G
- g_W is a classifier that maps representations to labels

Caron, M., Bojanowski, P., Joulin, A., & Douze, M. (2018). Deep clustering for unsupervised learning of visual features. ECCV (pp. 132-149).

Image Classification: A Mathematical Perspective

$$\min_{\theta,W} \frac{1}{N} \sum_i \boxed{L\left(\boxed{g_W\boxed{\left(f_\theta\boxed{(x_i)}\right)}}, \boxed{y_i}\right)}$$

- Input x_i with label y_i
- f_Θ computes high-level representations of x_i G
- g_W is a classifier that maps representations to labels
- Loss function, e.g. logistic loss (negative log softmax)

Caron, M., Bojanowski, P., Joulin, A., & Douze, M. (2018). Deep clustering for unsupervised learning of visual features. ECCV (pp. 132-149).

Image Classification: A Mathematical Perspective

$$\boxed{\min_{\theta,W}} \frac{1}{N} \sum_i \boxed{L\left(\boxed{g_W\boxed{\left(f_\theta\boxed{(x_i))}\right)}}, \boxed{y_i}\right)}$$

Q: We said this was unsupervised.
What is y_i?

- Input x_i with label y_i
- f_Θ computes high-level representations of x_i G
- g_W is a classifier that maps representations to labels
- Loss function, e.g. logistic loss (negative log softmax)

→ **Task: Find parameters Θ,W such that loss is optimal**

Caron, M., Bojanowski, P., Joulin, A., & Douze, M. (2018). Deep clustering for unsupervised learning of visual features. ECCV (pp. 132-149).

From Embeddings to Pseudo-labels

- Let's assume Θ is known → Representations are informative!
- Then, we only need to find W for correct class assignments

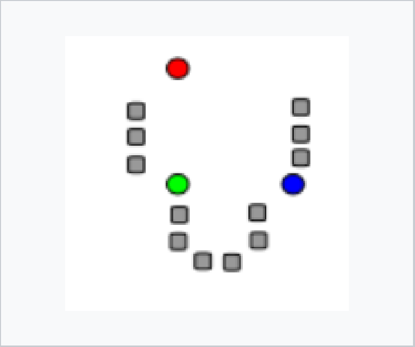
- Idea
  - Representations are informative
  - The embedding of images with similar appearance/content will be similar
  - The embedding of images with dissimilar appearance/content will be dissimilar
  - **Cluster these embeddings to assign pseudo-labels!**

- What clustering to use?

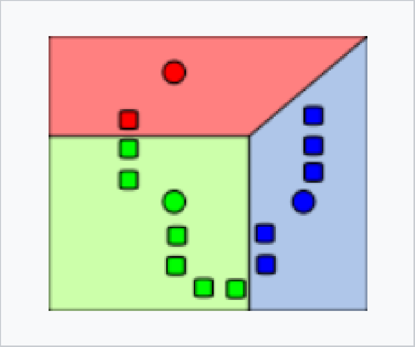
Caron, M., Bojanowski, P., Joulin, A., & Douze, M. (2018). Deep clustering for unsupervised learning of visual features. ECCV (pp. 132-149).

Clustering

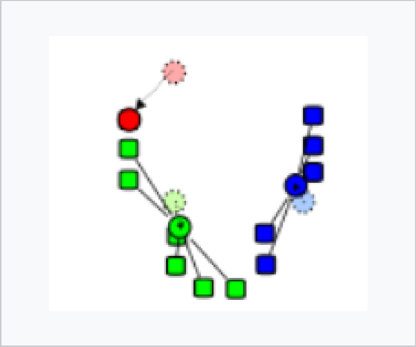
- Every image is embedded based on its content/appearance



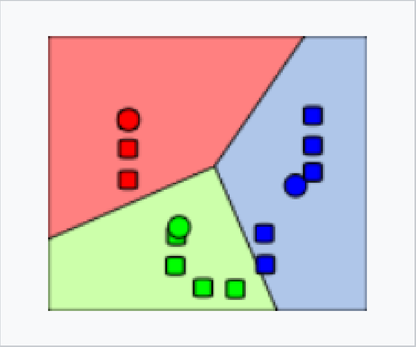
1. *k* initial "means" (in this case *k*=3) are randomly generated within the data domain (shown in color).



2. *k* clusters are created by associating every observation with the nearest mean. The partitions here represent the Voronoi diagram generated by the means.



3. The centroid of each of the *k* clusters becomes the new mean.

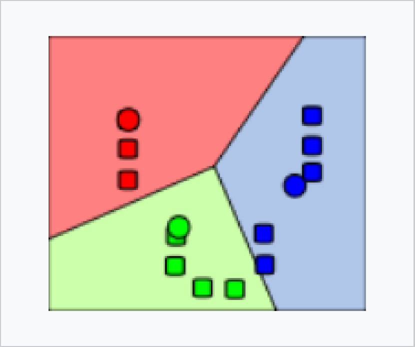


4. Steps 2 and 3 are repeated until convergence has been reached.

Caron, M., Bojanowski, P., Joulin, A., & Douze, M. (2018). Deep clustering for unsupervised learning of visual features. ECCV (pp. 132-149).

Clustering

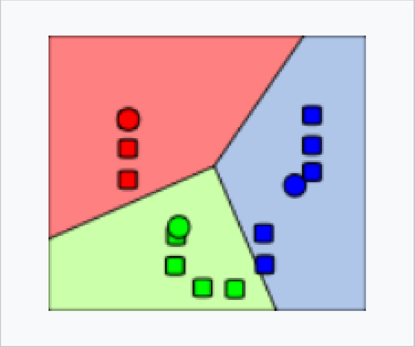
- Every image is embedded based on its content/appearance
- OK, we have pseudo-labels.
  But assumption was: Θ is already known!
- In reality, Θ is not known. Specifically, at the very first iteration/epoch **Θ** is not only not known but **random**!
- **Q: Problem?**



Caron, M., Bojanowski, P., Joulin, A., & Douze, M. (2018). Deep clustering for unsupervised learning of visual features. ECCV (pp. 132-149).

Clustering

- Every image is embedded based on its content/appearance



- OK, we have pseudo-labels.
  But assumption was: Θ is already known!

- In reality, Θ is not known. Specifically, at the very first iteration/epoch **Θ** is not only not known but **random**!

- **Q: Problem?**
- Turns out: **No problem.**
  An MLP classifier on top of random AlexNet achieves 12.5% accuracy!
  Random chance is at 0.1%!

Caron, M., Bojanowski, P., Joulin, A., & Douze, M. (2018). Deep clustering for unsupervised learning of visual features. ECCV (pp. 132-149).

Deep Clustering

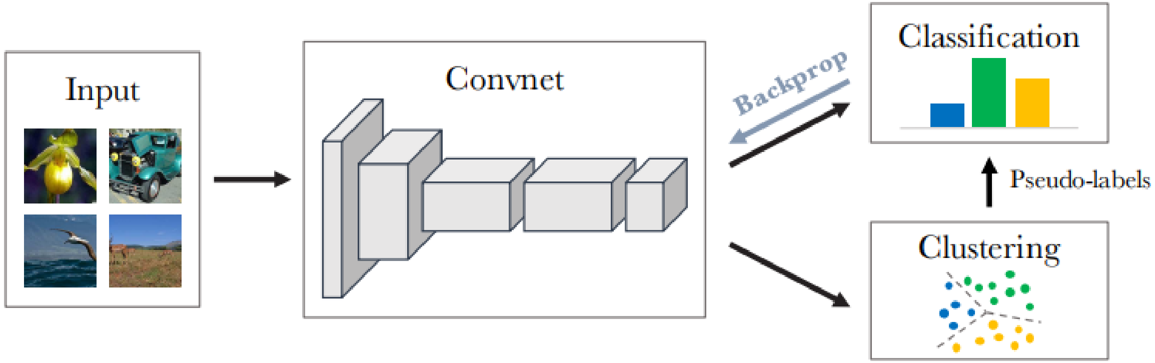


Fig. 1: Illustration of the proposed method: we iteratively cluster deep features and use the cluster assignments as pseudo-labels to learn the parameters of the convnet

- Alternates between
  1) clustering features to obtain pseudo-labels and
  2) updating the parameters of the ConvNet to predict these pseudo-labels.

- **Problem**?

Caron, M., Bojanowski, P., Joulin, A., & Douze, M. (2018). Deep clustering for unsupervised learning of visual features. ECCV (pp. 132-149).

Deep Clustering

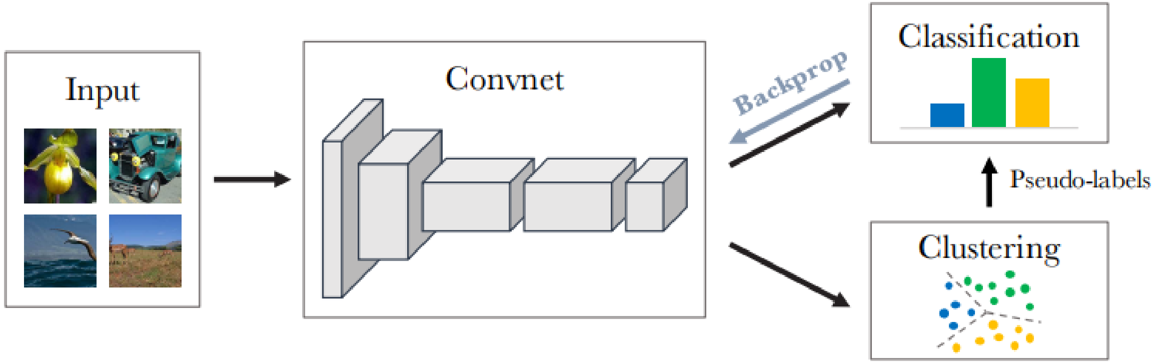


Fig. 1: Illustration of the proposed method: we iteratively cluster deep features and use the cluster assignments as pseudo-labels to learn the parameters of the convnet

- Alternates between
  1) clustering features to obtain pseudo-labels and
  2) updating the parameters of the ConvNet to predict these pseudo-labels.

- **Problem**? Assigning all points to a single cluster is an optimal solution.

Caron, M., Bojanowski, P., Joulin, A., & Douze, M. (2018). Deep clustering for unsupervised learning of visual features. ECCV (pp. 132-149).

Deep Clustering

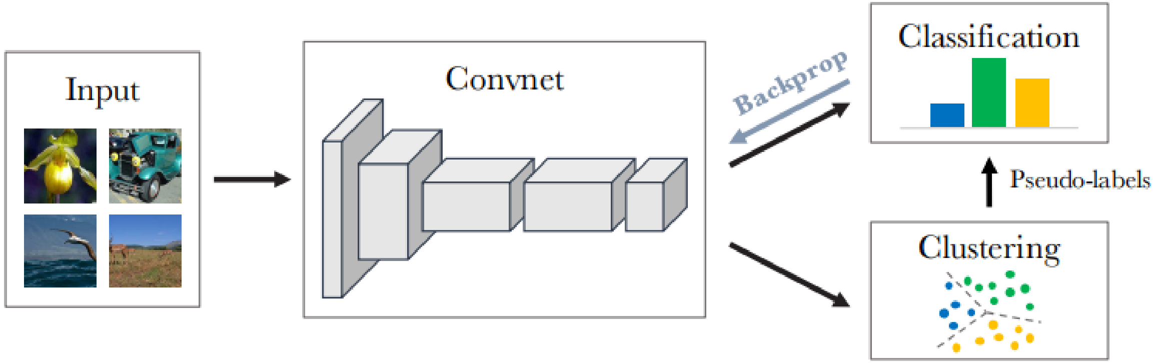


Fig. 1: Illustration of the proposed method: we iteratively cluster deep features and use the cluster assignments as pseudo-labels to learn the parameters of the convnet

- Avoiding trivial solutions
  - Avoiding empty clusters by re-assigning empty clusters
  - If most images are assigned to very few clusters, parameterization can become trivial
    → This is similar to highly unbalanced data
    → Sample images based on uniform distribution over the pseudo-labels
- Forward pass on full dataset, k-means clustering, backprop
- AlexNet with labels: 50.5% Top-1 – DeepCluster: 38.2% Top-1

Caron, M., Bojanowski, P., Joulin, A., & Douze, M. (2018). Deep clustering for unsupervised learning of visual features. ECCV (pp. 132-149).

Learning Good Representations from Data

- Often times, people make analogies to “how babies learn”
- No semantic information, but once provided, baby can identify the items

- An example:

Learning Good Representations from Data

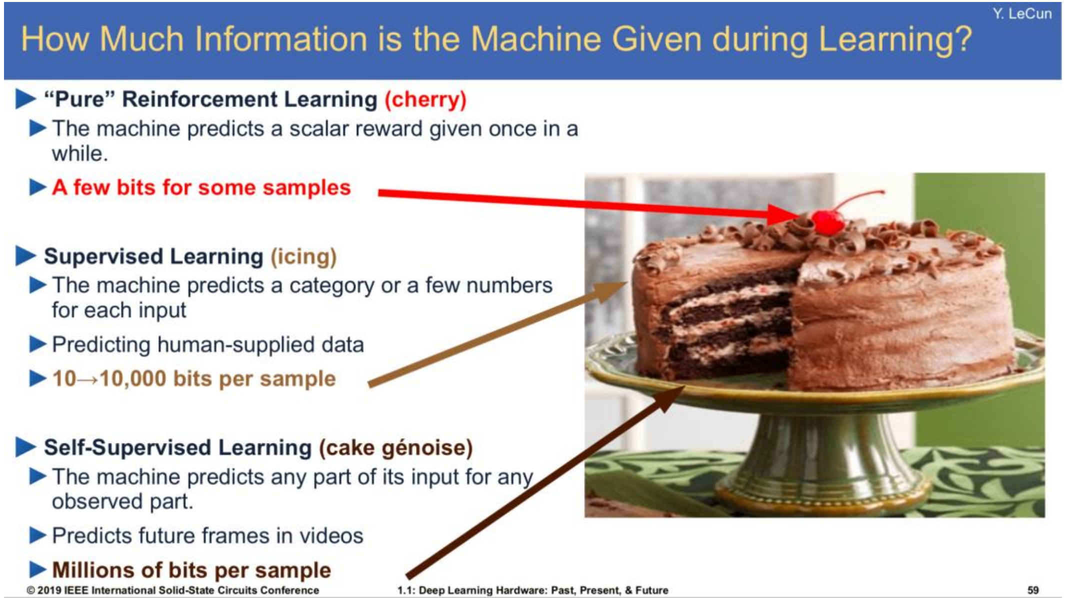
- Often times, people make analogies to “how babies learn”
- No semantic information, but once provided, baby can identify the items

- And another one: Conceptual model is violated

Learning Good Representations from Data

Derive auxiliary tasks from the data itself

Data is high-dimensional, these auxiliary tasks can be very informative!



Y. LeCun

How Much Information is the Machine Given during Learning?

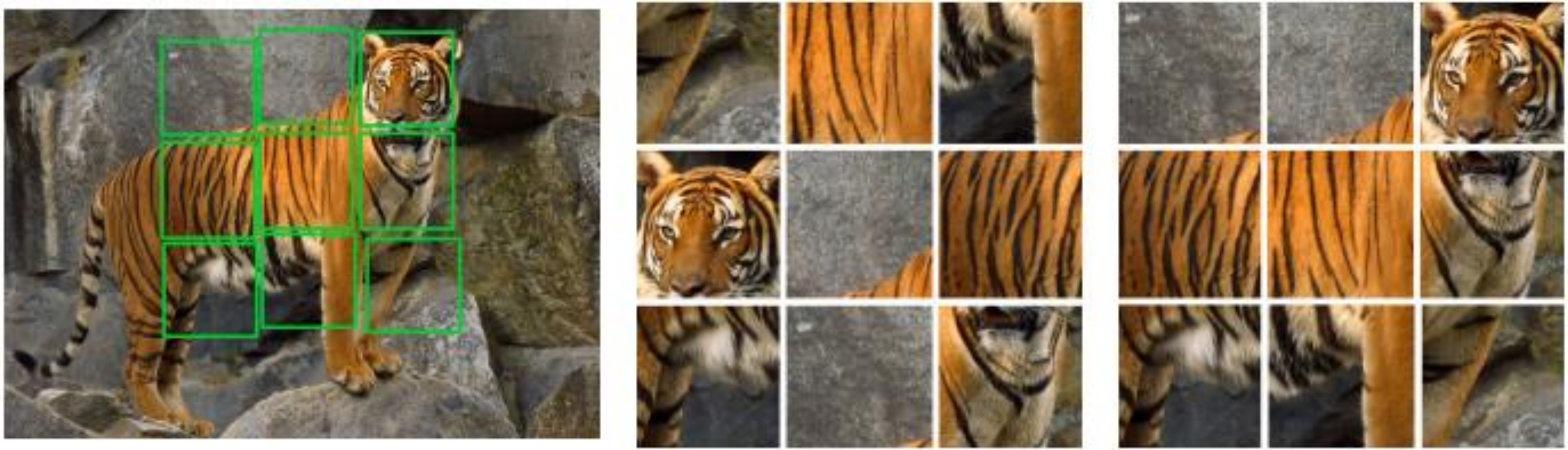
- **“Pure” Reinforcement Learning (cherry)**
  - The machine predicts a scalar reward given once in a while.
  - **A few bits for some samples**
- **Supervised Learning (icing)**
  - The machine predicts a category or a few numbers for each input
  - Predicting human-supplied data
  - **10→10,000 bits per sample**
- **Self-Supervised Learning (cake génoise)**
  - The machine predicts any part of its input for any observed part.
  - Predicts future frames in videos
  - **Millions of bits per sample**

© 2019 IEEE International Solid-State Circuits Conference 1.1: Deep Learning Hardware: Past, Present, & Future 59

Unsupervised and Self-supervised Learning

From Images

Jigsaw Puzzle Reassembly



- Introduced 1760 by John Spilsbury for children to learn geography
- Used to assess individual’s ability to organize visual stimuli

→ Use puzzles to develop a visuospatial representation of objects in context

Noroozi, M., & Favaro, P. (2016, October). Unsupervised learning of visual representations by solving jigsaw puzzles. ECCV (pp. 69-84).

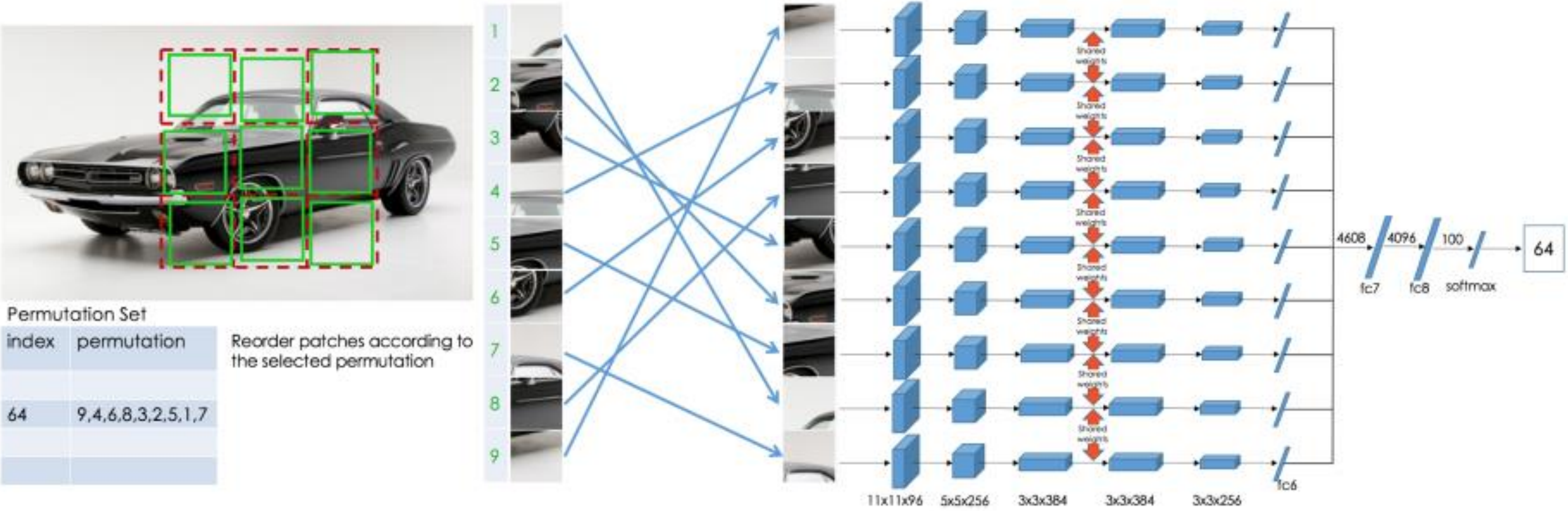
Jigsaw Puzzle Reassembly

- How to model jigsaw puzzle solving?

- Immediate approach: Stack puzzle tiles along channels (depth)
  - CNN may concentrate on low level statistics
    → Similar texture patterns at boundaries
  - This does not promote understanding of global object properties

- Idea: Delay computation of statistics across tiles
  - First, compute features/representations of every tile separately
  - Then, determine arrangement only based on this representation!

Noroozi, M., & Favaro, P. (2016, October). Unsupervised learning of visual representations by solving jigsaw puzzles. ECCV (pp. 69-84).



Permutation Set

index	permutation
64	9,4,6,8,3,2,5,1,7

Reorder patches according to the selected permutation

- Randomly crop 225x225 px window from image (red box)
- Divide into 3x3 cell grid and pick random 64x64 pixel tile from each cell
- 9 tiles are re-ordered randomly according to one of the 9! permutations
- Network predicts premutation index (one-hot)

Noroozi, M., & Favaro, P. (2016, October). Unsupervised learning of visual representations by solving jigsaw puzzles. ECCV (pp. 69-84).

Jigsaw Puzzle Reassembly

- Q: Any challenges that would prevent generalizable learning?

Noroozi, M., & Favaro, P. (2016, October). Unsupervised learning of visual representations by solving jigsaw puzzles. ECCV (pp. 69-84).

Jigsaw Puzzle Reassembly

- Q: Any challenges that would prevent generalizable learning? **Shortcuts!** Representations that are suitable for the pre-text **but not the target task.** → Features would carry no semantic information

- Shortcut 1: CNN may memorize absolute positions of tiles
  - → Feed in multiple puzzles at once to make absolute positions equally likely

- Shortcut 2: Edge continuity
  - → Random gap between tiles

- Shortcut 3: Pixel intensity / color distributions
  - → Jitter on color channels

Noroozi, M., & Favaro, P. (2016, October). Unsupervised learning of visual representations by solving jigsaw puzzles. ECCV (pp. 69-84).

Jigsaw Puzzle Reassembly

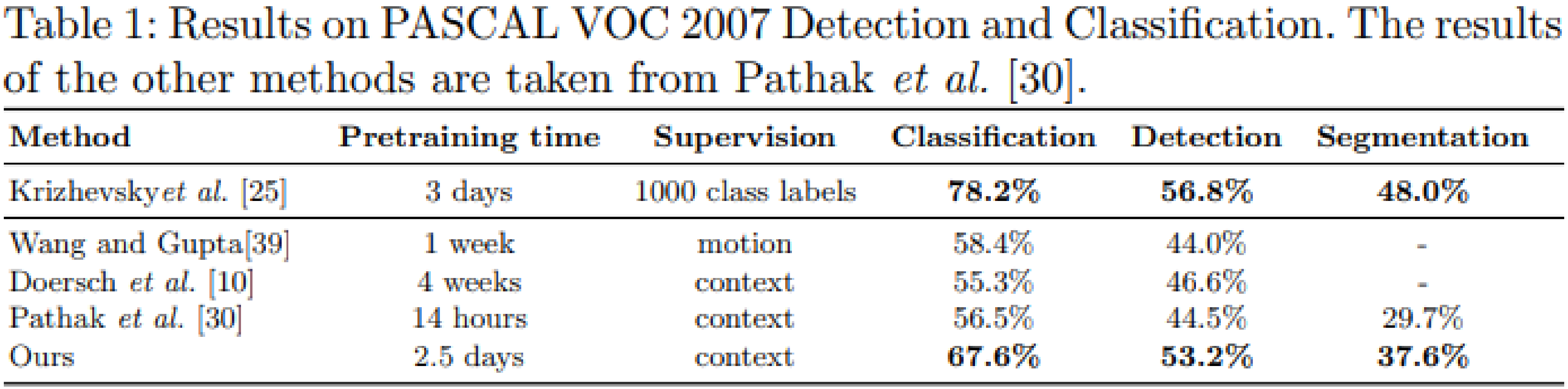


Table 1: Results on PASCAL VOC 2007 Detection and Classification. The results of the other methods are taken from Pathak *et al.* [30].

Method	Pretraining time	Supervision	Classification	Detection	Segmentation
Krizhevsky*et al.* [25]	3 days	1000 class labels	**78.2%**	**56.8%**	**48.0%**
Wang and Gupta[39]	1 week	motion	58.4%	44.0%	-
Doersch *et al.* [10]	4 weeks	context	55.3%	46.6%	-
Pathak *et al.* [30]	14 hours	context	56.5%	44.5%	29.7%
Ours	2.5 days	context	**67.6%**	**53.2%**	**37.6%**

- Performs well and is comparably fast to train

Noroozi, M., & Favaro, P. (2016, October). Unsupervised learning of visual representations by solving jigsaw puzzles. ECCV (pp. 69-84).

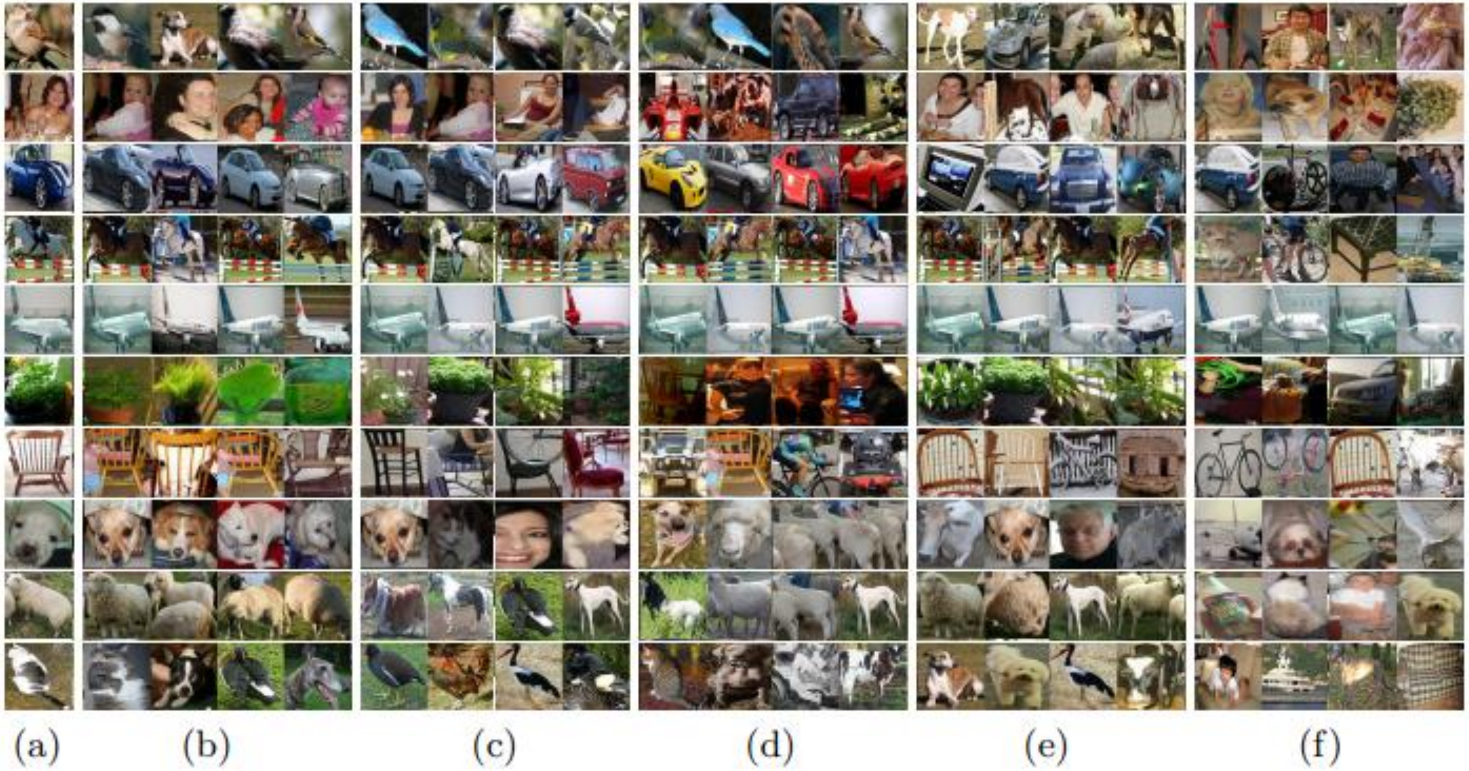


Fig. 5: Image retrieval (qualitative evaluation). (a) query images; (b) top-4 matches with AlexNet; (c) top-4 matches with the CFN trained without blocking chromatic aberration; (d) top-4 matches with Doersch *et al.* [10]; (e) top-4 matches with Wang and Gupta [39]; (f) top-4 matches with AlexNet with random weights.

Exemplar Networks

- **Q: When we perform data augmentation, what is our core assumption?**

Dosovitskiy, A., Springenberg, J. T., Riedmiller, M., & Brox, T. (2014). Discriminative unsupervised feature learning with convolutional neural networks. NeurIPS (pp. 766-774).

Exemplar Networks

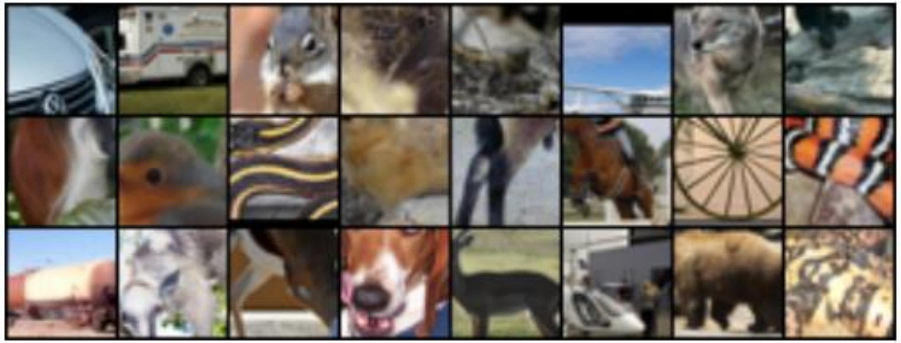


Figure 1: Exemplary patches sampled from the STL unlabeled dataset which are later augmented by various transformations to obtain surrogate data for the CNN training.

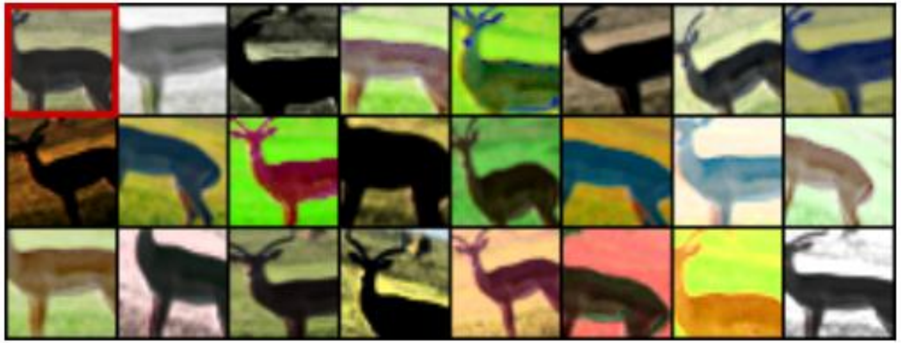


Figure 2: Several random transformations applied to one of the patches extracted from the STL unlabeled dataset. The original ('seed') patch is in the top left corner.

- Data augmentation to the max!
- Altering an image will produce an image of the same class! → Pseudo-labels!

Dosovitskiy, A., Springenberg, J. T., Riedmiller, M., & Brox, T. (2014). Discriminative unsupervised feature learning with convolutional neural networks. NeurIPS (pp. 766-774).

Exemplar Networks

- Family of “exemplar” transformations
  - **Translation**: vertical or horizontal translation by a distance within 0.2 of the patch size;
  - **Scaling**: multiplication of the patch scale by a factor between 0.7 and 1.4;
  - **Rotation**: rotation of the image by an angle up to 20 degrees;
  - **Contrast** 1: multiply the projection of each patch pixel onto the principal components of the set of all pixels by a factor between 0.5 and 2 (factors are independent for each principal component and the same for all pixels within a patch);
  - **Contrast** 2: raise saturation and value (S and V components of the HSV color representation) of all pixels to a power between 0.25 and 4 (same for all pixels within a patch), multiply these values by a factor between 0.7 and 1.4, add to them a value between −0.1 and 0.1;
  - **Color**: add a value between −0.1 and 0.1 to the hue (H component of the HSV color representation) of all pixels in the patch (the same value is used for all pixels within a patch).

- For each patch, sample 300 augmentation parameters

Dosovitskiy, A., Springenberg, J. T., Riedmiller, M., & Brox, T. (2014). Discriminative unsupervised feature learning with convolutional neural networks. NeurIPS (pp. 766-774).

Exemplar Networks

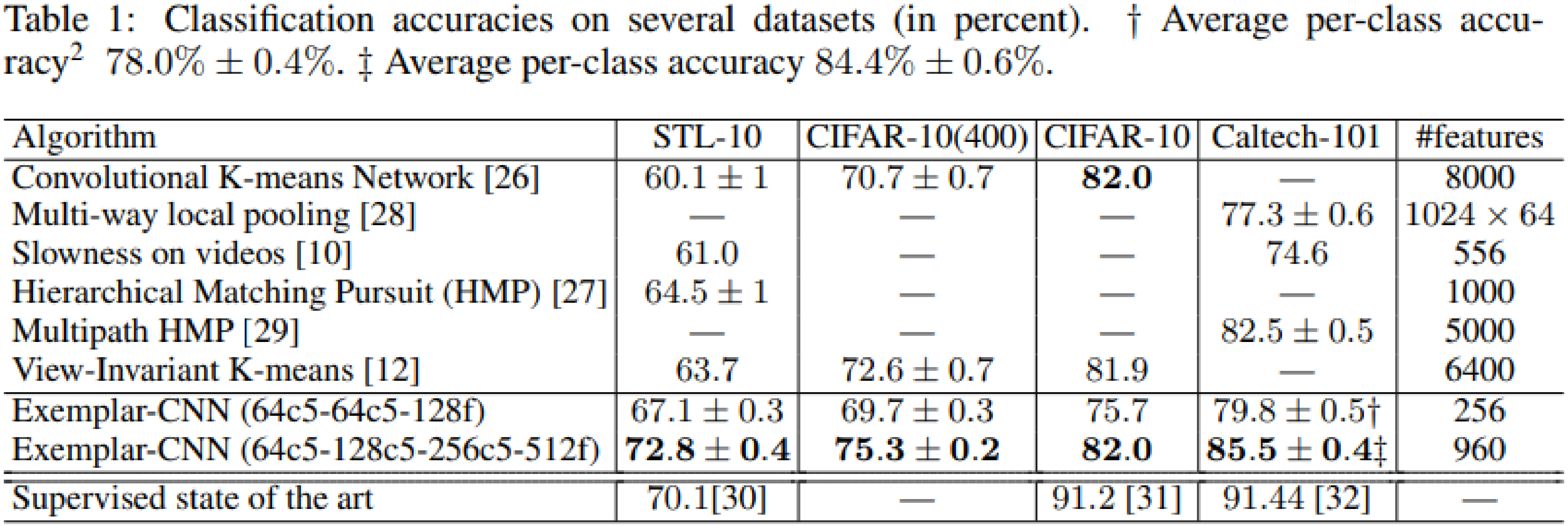


Table 1: Classification accuracies on several datasets (in percent). † Average per-class accuracy^2 78.0% ± 0.4%. ‡ Average per-class accuracy 84.4% ± 0.6%.

Algorithm	STL-10	CIFAR-10(400)	CIFAR-10	Caltech-101	#features
Convolutional K-means Network [26]	60.1 ± 1	70.7 ± 0.7	**82.0**	—	8000
Multi-way local pooling [28]	—	—	—	77.3 ± 0.6	1024×64
Slowness on videos [10]	61.0	—	—	74.6	556
Hierarchical Matching Pursuit (HMP) [27]	64.5 ± 1	—	—	—	1000
Multipath HMP [29]	—	—	—	82.5 ± 0.5	5000
View-Invariant K-means [12]	63.7	72.6 ± 0.7	81.9	—	6400
Exemplar-CNN (64c5-64c5-128f)	67.1 ± 0.3	69.7 ± 0.3	75.7	79.8 ± 0.5†	256
Exemplar-CNN (64c5-128c5-256c5-512f)	$\mathbf{72.8 \pm 0.4}$	$\mathbf{75.3 \pm 0.2}$	**82.0**	$\mathbf{85.5 \pm 0.4}$‡	960
Supervised state of the art	70.1[30]	—	91.2 [31]	91.44 [32]	—

- Q: Challenges with this approach?

Dosovitskiy, A., Springenberg, J. T., Riedmiller, M., & Brox, T. (2014). Discriminative unsupervised feature learning with convolutional neural networks. NeurIPS (pp. 766-774).

Exemplar Networks

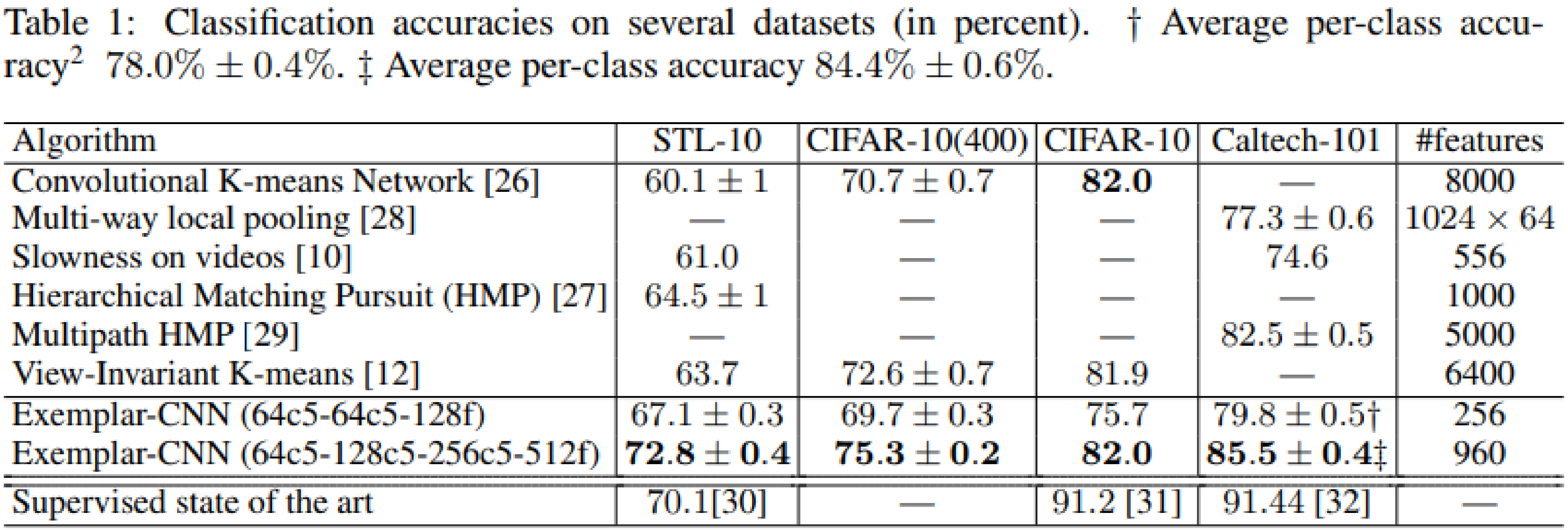


Table 1: Classification accuracies on several datasets (in percent). † Average per-class accuracy^2 78.0% ± 0.4%. ‡ Average per-class accuracy 84.4% ± 0.6%.

Algorithm	STL-10	CIFAR-10(400)	CIFAR-10	Caltech-101	#features
Convolutional K-means Network [26]	60.1 ± 1	70.7 ± 0.7	**82.0**	—	8000
Multi-way local pooling [28]	—	—	—	77.3 ± 0.6	1024 × 64
Slowness on videos [10]	61.0	—	—	74.6	556
Hierarchical Matching Pursuit (HMP) [27]	64.5 ± 1	—	—	—	1000
Multipath HMP [29]	—	—	—	82.5 ± 0.5	5000
View-Invariant K-means [12]	63.7	72.6 ± 0.7	81.9	—	6400
Exemplar-CNN (64c5-64c5-128f)	67.1 ± 0.3	69.7 ± 0.3	75.7	79.8 ± 0.5†	256
Exemplar-CNN (64c5-128c5-256c5-512f)	**72.8 ± 0.4**	**75.3 ± 0.2**	**82.0**	**85.5 ± 0.4‡**	960
Supervised state of the art	70.1[30]	—	91.2 [31]	91.44 [32]	—

- Q: Challenges with this approach?
- Scalability! Exemplars are created via augmentation. How to deal with more complex variation, such as viewpoint changes?

Dosovitskiy, A., Springenberg, J. T., Riedmiller, M., & Brox, T. (2014). Discriminative unsupervised feature learning with convolutional neural networks. NeurIPS (pp. 766-774).

Why does this work?

Re-colorization

- In order to attribute color to the object, we must recognize the object first

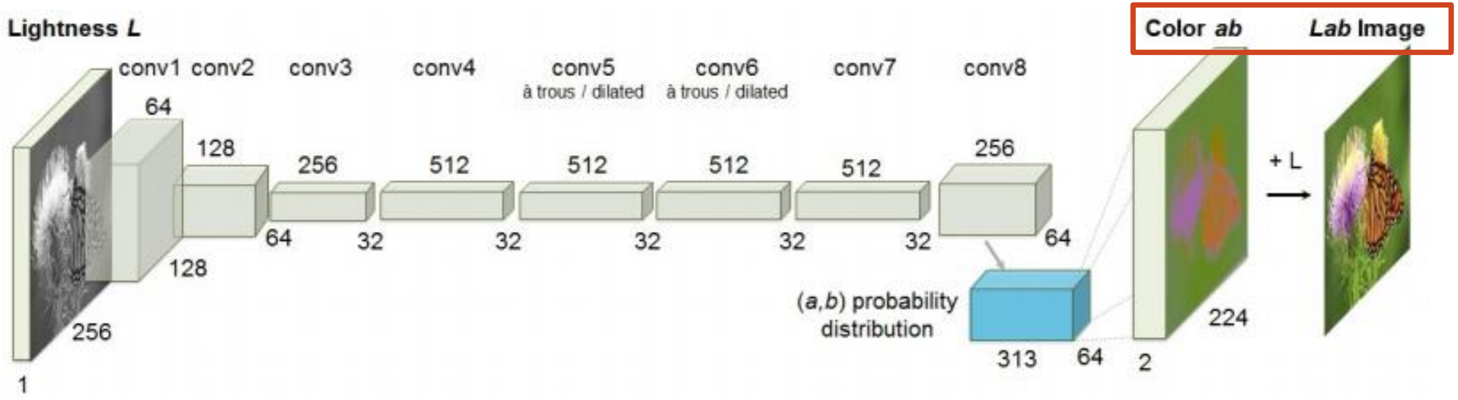


Fig. 2. Our network architecture. Each **conv** layer refers to a block of 2 or 3 repeated **conv** and **ReLU** layers, followed by a **BatchNorm** [30] layer. The net has no **pool** layers. All changes in resolution are achieved through spatial downsampling or upsampling between **conv** blocks.

Zhang, R., Isola, P., & Efros, A. A. (2016, October). Colorful image colorization. ECCV (pp. 649-666).

Re-colorization

- Why “CIE Lab” color space? Distances in Lab model perceptual distance

- But using L2 is not a good idea. **Q: Why?**

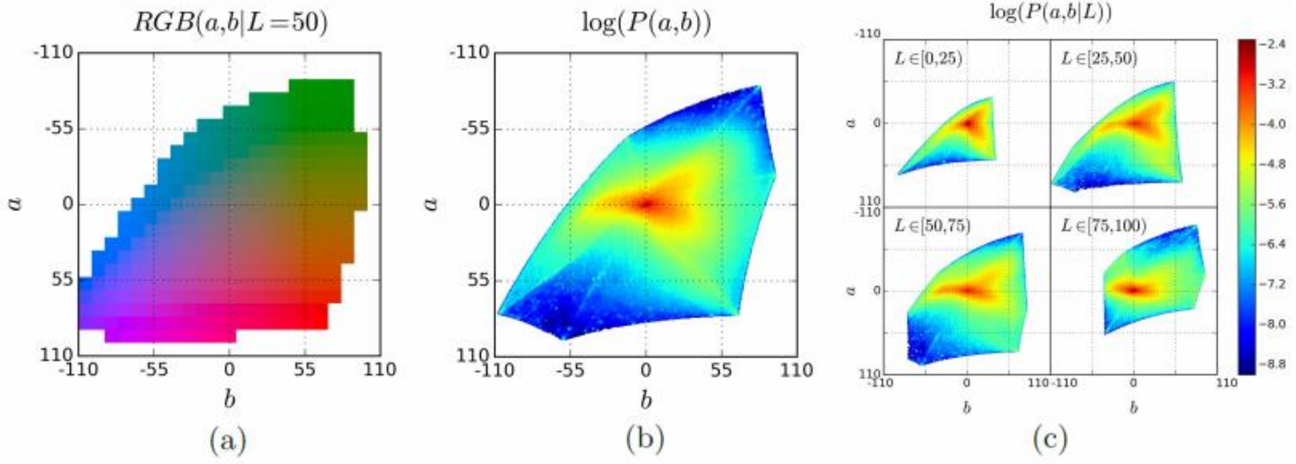


Fig. 3. (a) Quantized *ab* color space with a grid size of 10. A total of 313 *ab* pairs are in gamut. (b) Empirical probability distribution of *ab* values, shown in log scale. (c) Empirical probability distribution of *ab* values, conditioned on *L*, shown in log scale.

Zhang, R., Isola, P., & Efros, A. A. (2016, October). Col

Re-colorization

- Think about the chameleon. Objects can take multiple different colors!
- Rather than regression → Classification!

- Rebalancing needed! See Fig. 3b. Due to background, desaturated pixels are much more likely!
  → Weighting!

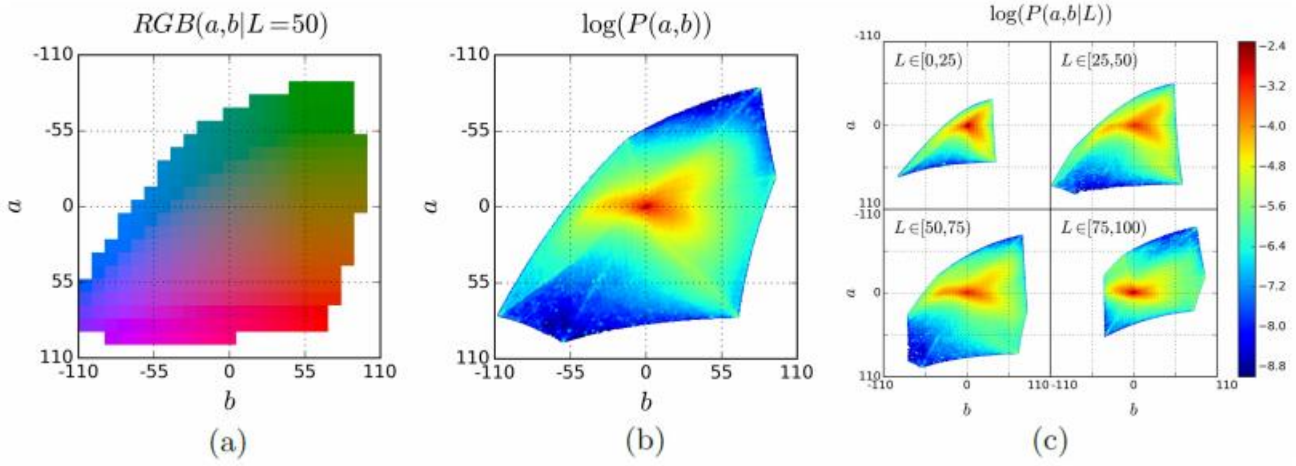
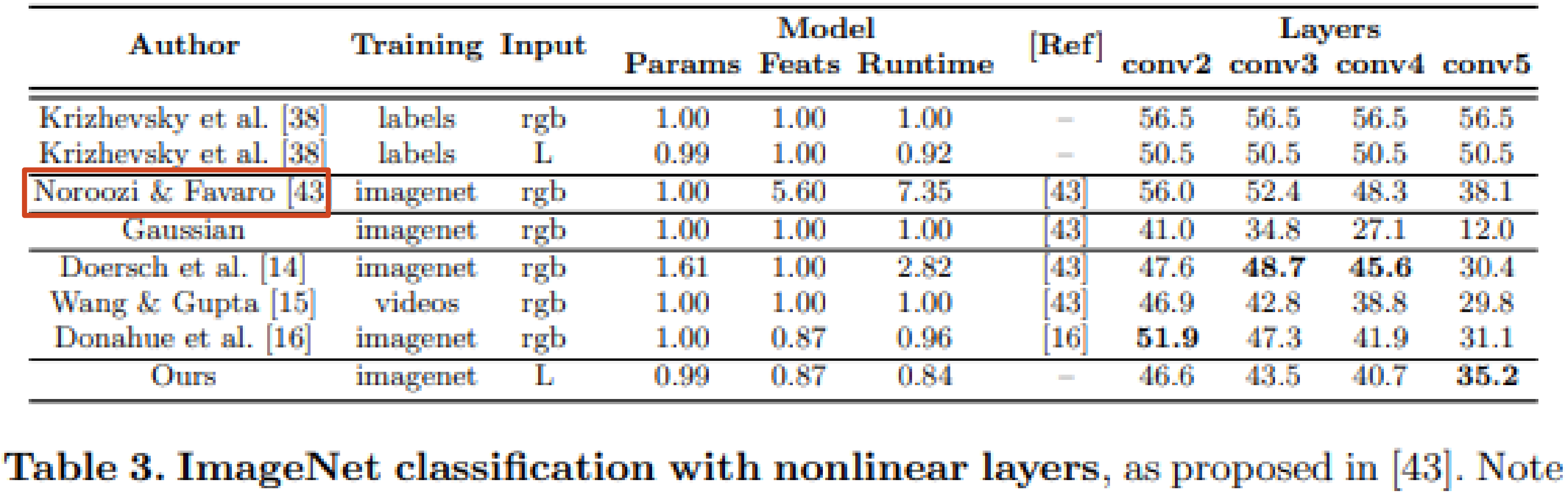


Fig. 3. (a) Quantized *ab* color space with a grid size of 10. A total of 313 *ab* pairs are in gamut. (b) Empirical probability distribution of *ab* values, shown in log scale. (c) Empirical probability distribution of *ab* values, conditioned on *L*, shown in log scale.

Zhang, R., Isola, P., & Efros, A. A. (2016, October). Col

Re-colorization

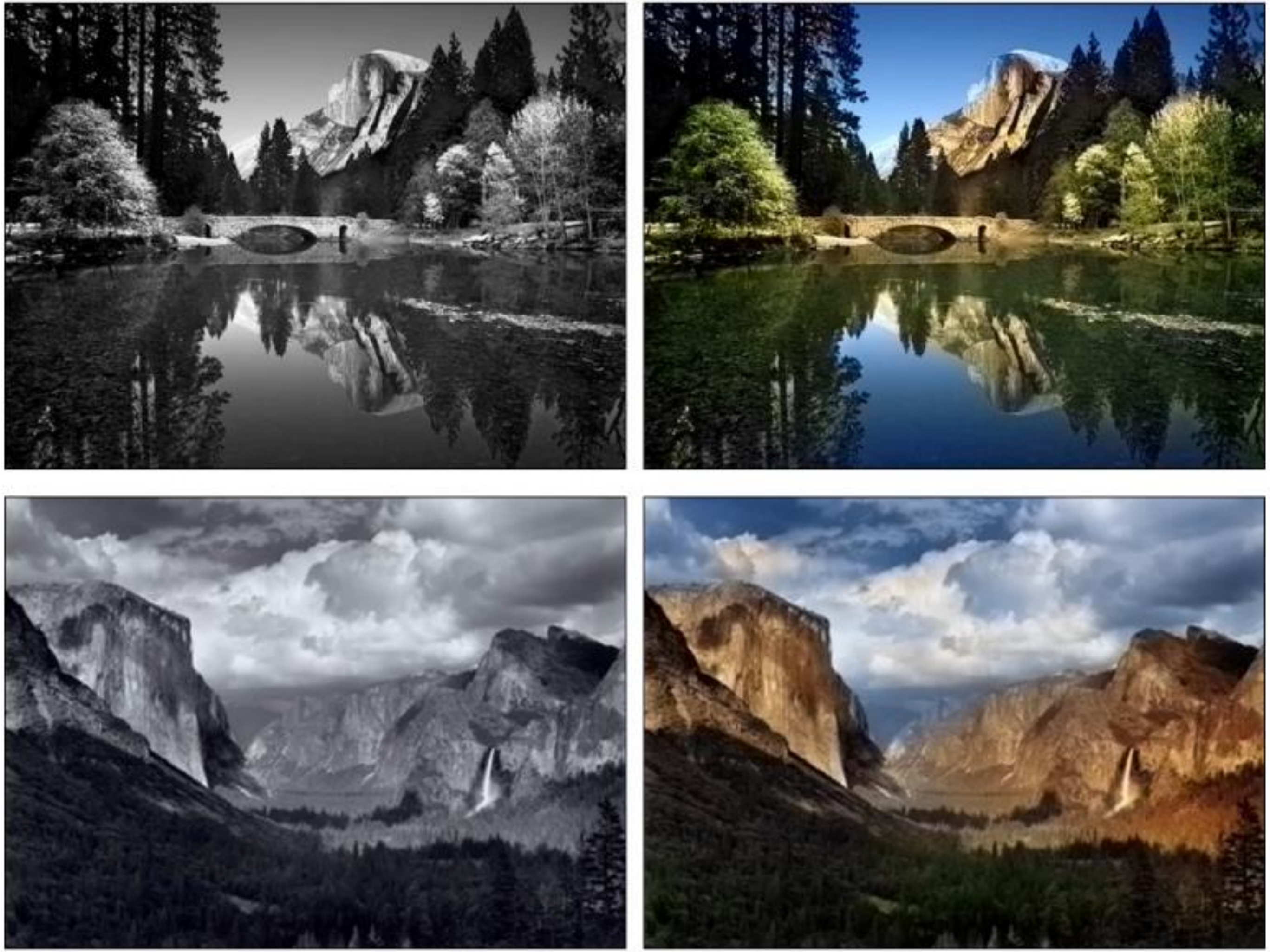


Author	Training	Input	Model			[Ref]	Layers			
			Params	Feats	Runtime		conv2	conv3	conv4	conv5
Krizhevsky et al. [38]	labels	rgb	1.00	1.00	1.00	–	56.5	56.5	56.5	56.5
Krizhevsky et al. [38]	labels	L	0.99	1.00	0.92	–	50.5	50.5	50.5	50.5
Noroozi & Favaro [43]	imagenet	rgb	1.00	5.60	7.35	[43]	56.0	52.4	48.3	38.1
Gaussian	imagenet	rgb	1.00	1.00	1.00	[43]	41.0	34.8	27.1	12.0
Doersch et al. [14]	imagenet	rgb	1.61	1.00	2.82	[43]	47.6	**48.7**	**45.6**	30.4
Wang & Gupta [15]	videos	rgb	1.00	1.00	1.00	[43]	46.9	42.8	38.8	29.8
Donahue et al. [16]	imagenet	rgb	1.00	0.87	0.96	[16]	**51.9**	47.3	41.9	31.1
Ours	imagenet	L	0.99	0.87	0.84	–	46.6	43.5	40.7	**35.2**

Table 3. ImageNet classification with nonlinear layers, as proposed in [43]. Note

Jigsaw puzzles

Zhang, R., Isola, P., & Efros, A. A. (2016, October). Colorful image colorization. ECCV (pp. 649-666).



Unsupervised and Self-supervised Learning

From Videos

Propagating Colors

- So far: Image colorization
  - Learn sematic features about objects to assign colors
  - Allows learning of generalizable representations

- Now: Video colorization
  - Naively done, this is just the same as image colorization
  - **But, can you think of an interesting twist?**

Vondrick, C., Shrivastava, A., Fathi, A., Guadarrama, S., & Murphy, K. (2018). Tracking emerges by colorizing videos. *ECCV* (pp. 391-408).

Propagating Colors

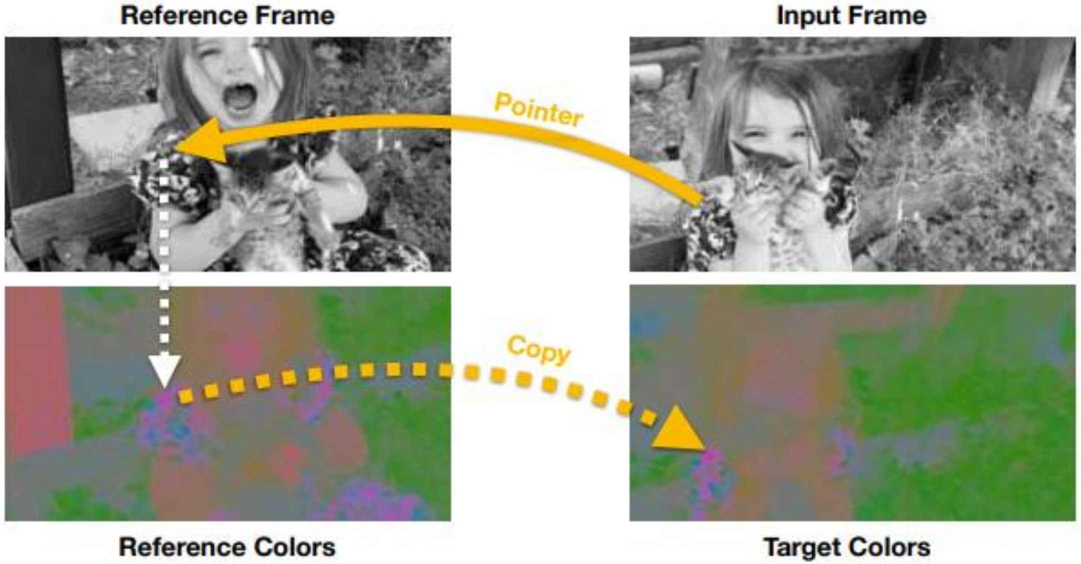


Fig. 1. Self-supervised Tracking: We capitalize on large amounts of unlabeled video to learn a self-supervised model for tracking. The model learns to predict the target colors for a gray-scale input frame by pointing to a colorful reference frame, and copying the color channels. Although we train without ground-truth labels, experiments and visualizations suggest that tracking emerges automatically in this model.

Vondrick, C., Shrivastava, A., Fathi, A., Guadarrama, S., & Murphy, K. (2018). Tracking emerges by colorizing videos. *ECCV* (pp. 391-408).

Propagating Colors

- Colorization then is performed using soft assign
- **Interestingly, this mechanism discovers object tracking!**

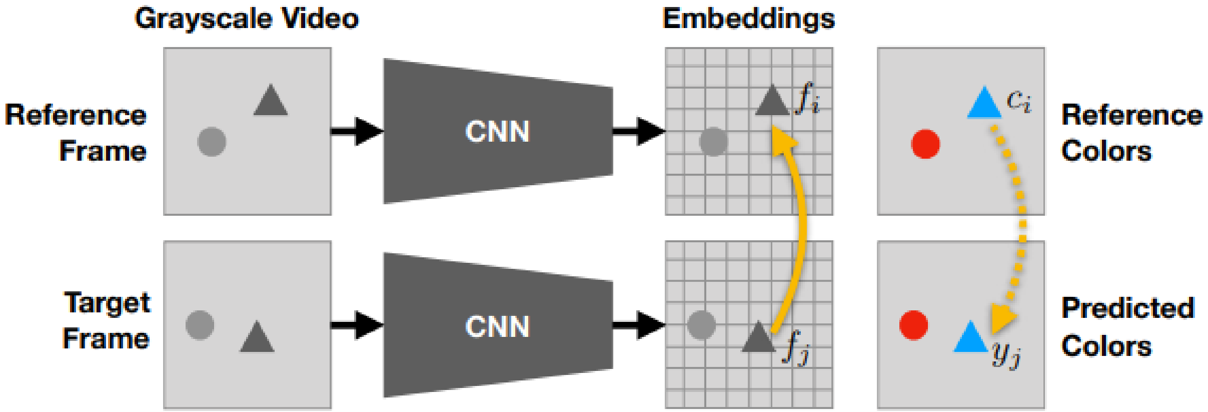
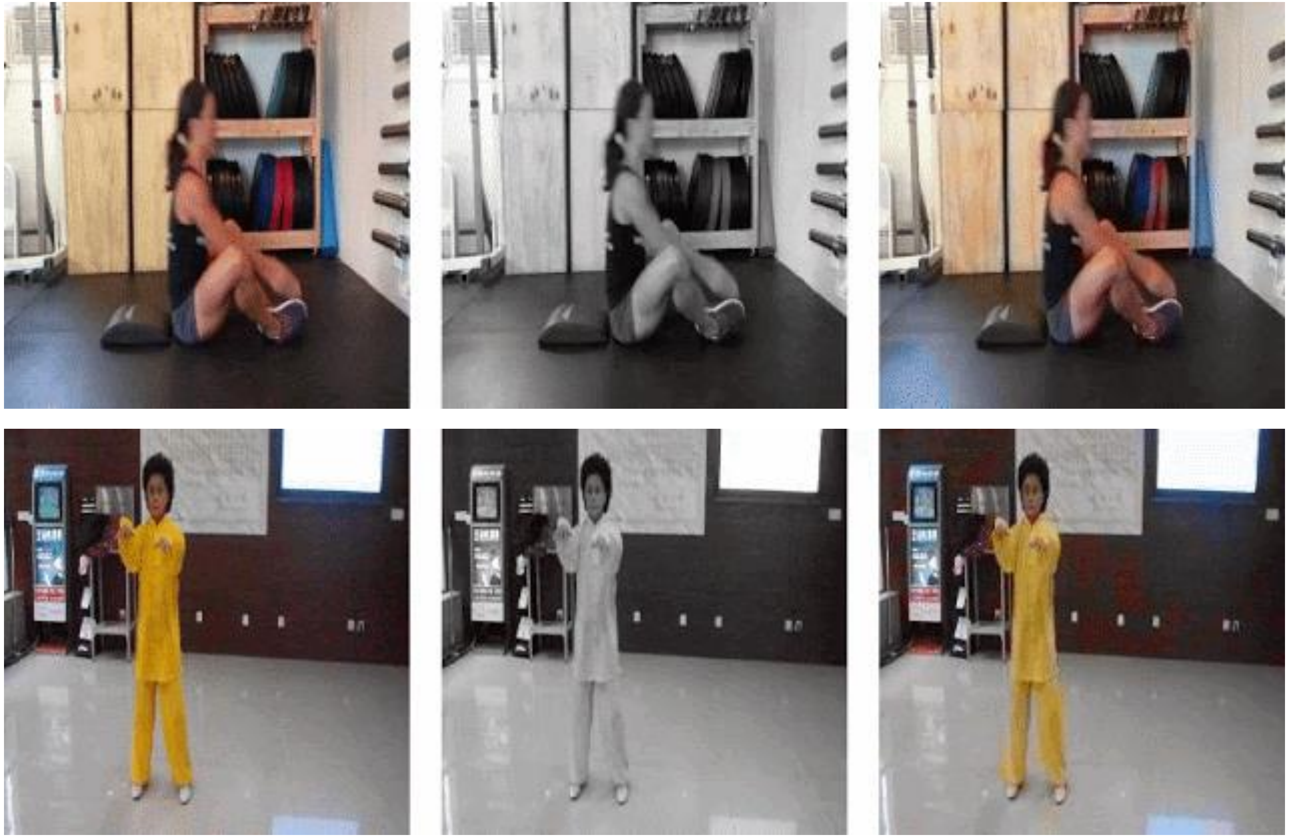


Fig. 2. Model Overview: Given gray-scale frames, the model computes low-dimensional embeddings for each location with a CNN. Using softmax similarity, the model points from the target frame into the reference frame embeddings (solid yellow arrow). The model then copies the color back into the predicted frame (dashed yellow arrow). After learning, we use the pointing mechanism as a visual tracker. Note that the model's pointer is soft, but for illustrations purposes we draw it as a single arrow.

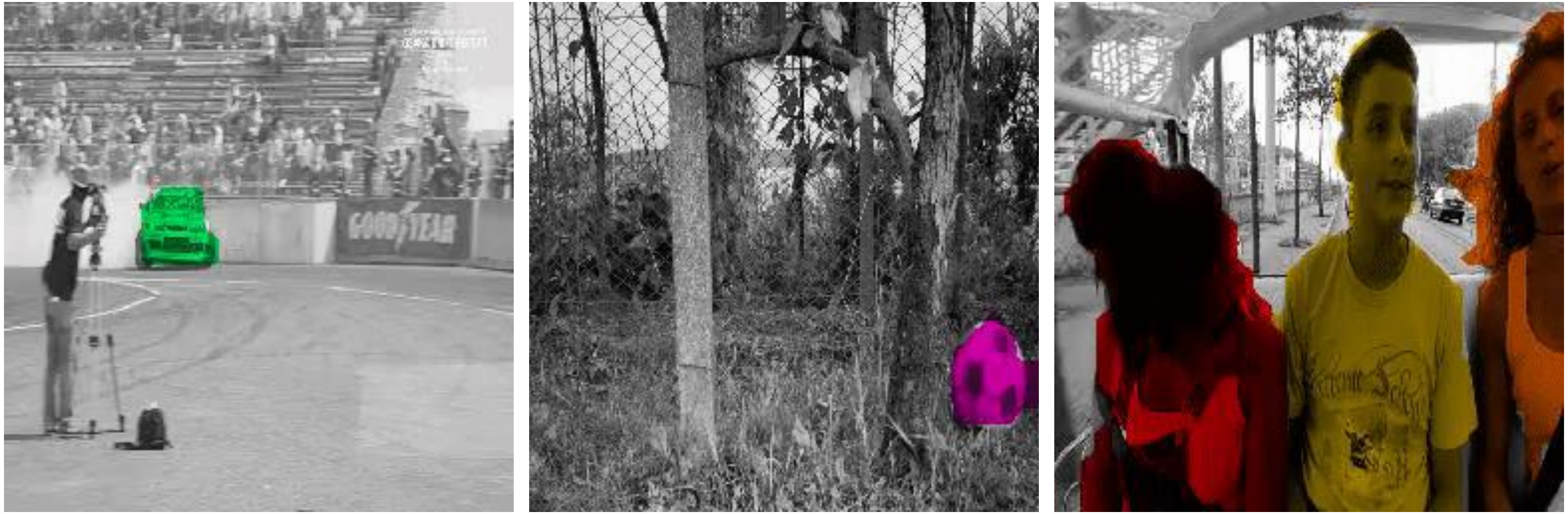
Vondrick, C., Shrivastava, A., Fathi, A., Guadarrama,

Propagating Colors



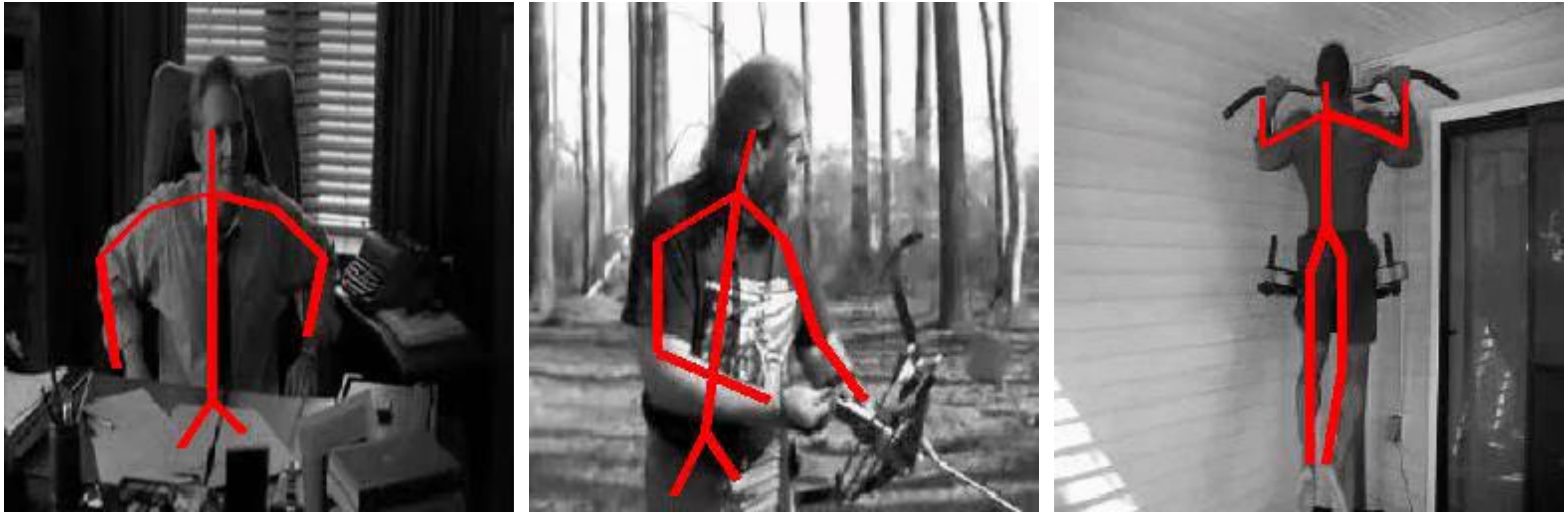
Vondrick, C., Shrivastava, A., Fathi, A., Guadarrama, S., & Murphy, K. (2018). Tracking emerges by colorizing videos. *ECCV* (pp. 391-408).

Propagating Colors



Vondrick, C., Shrivastava, A., Fathi, A., Guadarrama, S., & Murphy, K. (2018). Tracking emerges by colorizing videos. *ECCV* (pp. 391-408).

Propagating Colors



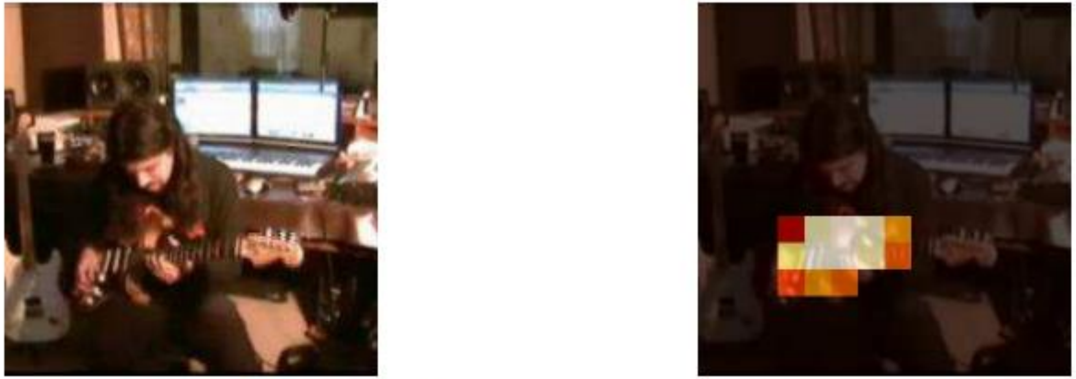
Vondrick, C., Shrivastava, A., Fathi, A., Guadarrama, S., & Murphy, K. (2018). Tracking emerges by colorizing videos. *ECCV* (pp. 391-408).

Unsupervised and Self-supervised Learning

From Multi-modal Data

Exploiting Multi-modal Information

- Can we localize sources of sound in images / a video sequence?



(a) Input image with sound

(b) Where is the sound?

Fig. 1. Where is the sound? Given an input image and sound clip, our method learns, without a single labelled example, to localize the object that makes the sound.

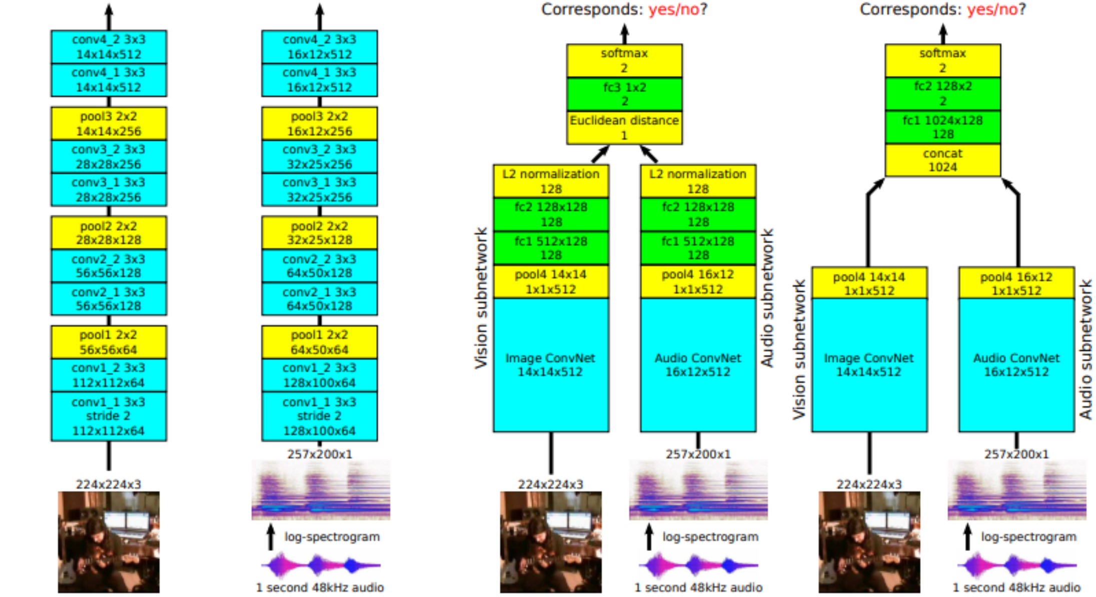
Arandjelovic, R., & Zisserman, A. (2018). Objects that sound. ECCV (pp. 435-451).

Exploiting Multi-modal Information

- In order to do so, we first need a network architecture that can process both
  - Video sequences / images
  - Audio

- Then, supervision comes for free from **audio-visual synchronization**

Arandjelovic, R., & Zisserman, A. (2018). Objects that sound. ECCV (pp. 435-451).

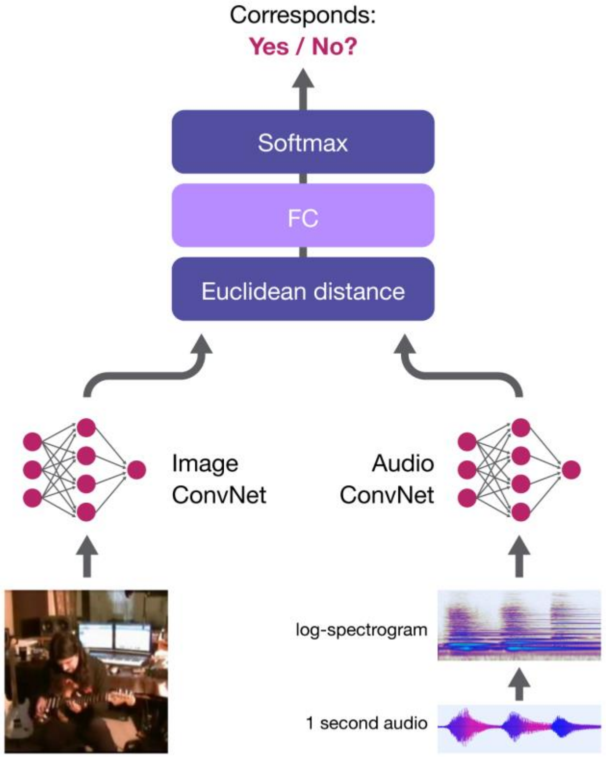


(a) Vision ConvNet (b) Audio ConvNet (c) AVE-Net (d) L^3-Net [4]

Fig. 2. ConvNet architectures. Each blocks represents a single layer with text providing more information – first row: layer name and optional kernel size, second row: output feature map size. Each convolutional layer is followed by batch normalization [25] and a ReLU nonlinearity, and the first fully connected layer (`fc1`) is followed by ReLU. All pool layers perform max pooling and their strides are equal to the kernel sizes. (a) and (b) show the vision and audio ConvNets which perform initial feature extraction from the image and audio inputs, respectively. (c) Our AVE-Net is designed to produce aligned vision and audio embeddings as the only information, a single scalar, used to decide whether the two inputs correspond is the Euclidean distance between the embeddings. (d) In contrast, the L^3-Net [4] architecture combines the two modalities by concatenation and a couple of fully connected layers which produce the corresponds or not classification scores.

Exploiting Multi-modal Information

- Does the 1s sound clip match my query image?
- Aligns latent spaces
- Can be used for audio clip retrieval
- Shortcuts! **Q: Can you think of one?**

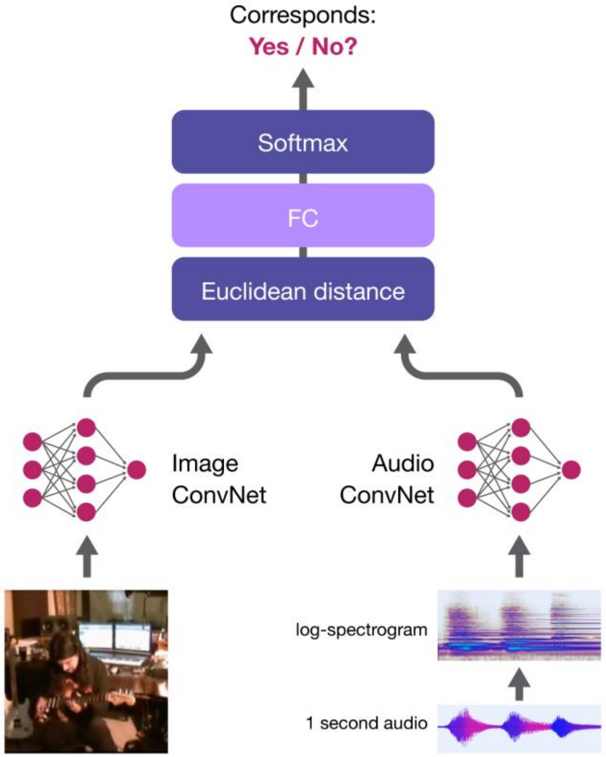


https://deepmind.com/blog/article/objects-that-sound

Arandjelovic, R., & Zisserman, A. (2018). Objects that sound. ECCV (pp. 435-451).

Exploiting Multi-modal Information

- Does the 1s sound clip match my query image?
- Aligns latent spaces
- Can be used for audio clip retrieval
- Shortcuts!
  - Positives are always sampled at N*0.04s (25fps)
  - Negatives must also be sampled that way!
    Network can identify otherwise, likely due to encoding artifact



https://deepmind.com/blog/article/objects-that-sound

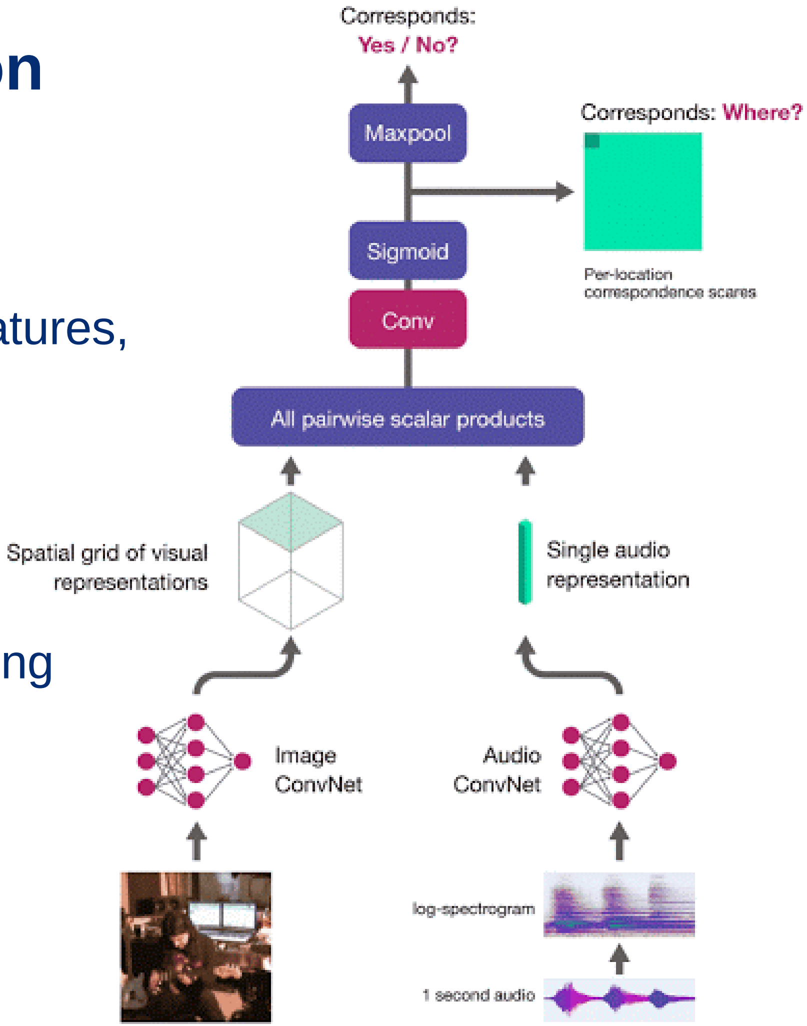
Arandjelovic, R., & Zisserman, A. (2018). Objects that sound. ECCV (pp. 435-451).

Exploiting Multi-modal Information

- Vision network does not globally pool the features, but operates on 14x14 resolution

- FC-layers are converted to fully conv. layers

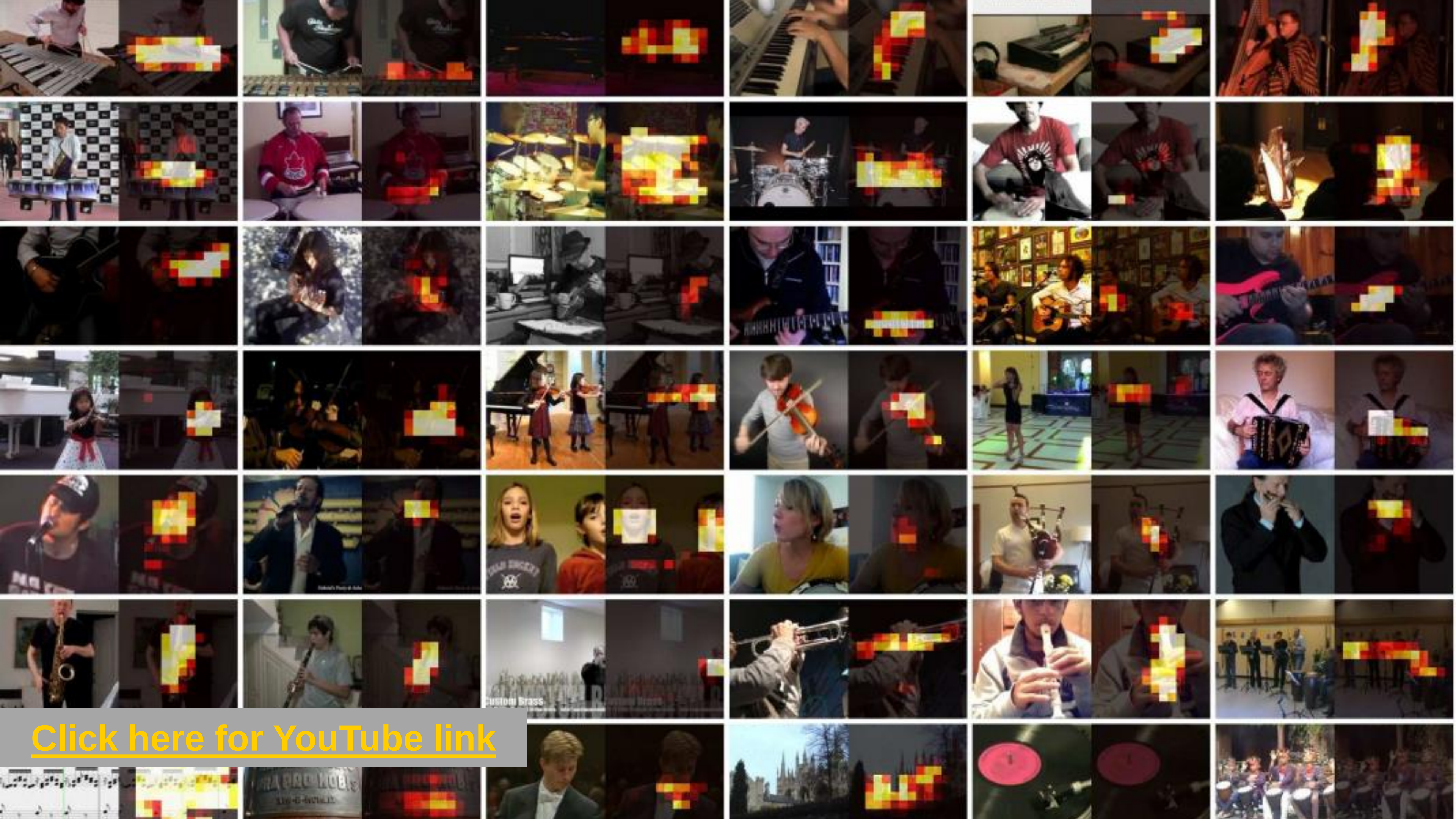
- Similarities between audio and visual encoding reveal location of objects that sound



Corresponds:
Yes / No?
Maxpool
Corresponds: Where?
Sigmoid
Per-location correspondence scores
Conv
All pairwise scalar products
Spatial grid of visual representations
Single audio representation
Image ConvNet
Audio ConvNet
log-spectrogram
1 second audio

https://deepmind.com/blog/article/objects-that-sound

Arandjelovic, R., & Zisserman, A. (2018). Objects that sound. ECCV (pp. 435-451).



Click here for YouTube link

Unsupervised and Self-supervised Learning

Exploiting Domain Knowledge

Extracting Weak Annotations

- In previous applications, we …
  - … solved for a surrogate task
  - … and then transferred features to our target task

- In some cases, we may be able to target this problem more directly

- Rather than solving for a surrogate task, maybe we can get sparse / weak annotations for our target task?

- **Q: Can you find examples for this?**

“Error-corrective Boosting”

- Brain segmentation is possible using FreeSurfer
  - Reasonably accurate but takes a very long time (> hours)
  - Manual labels: Even more accurate but even more time consuming

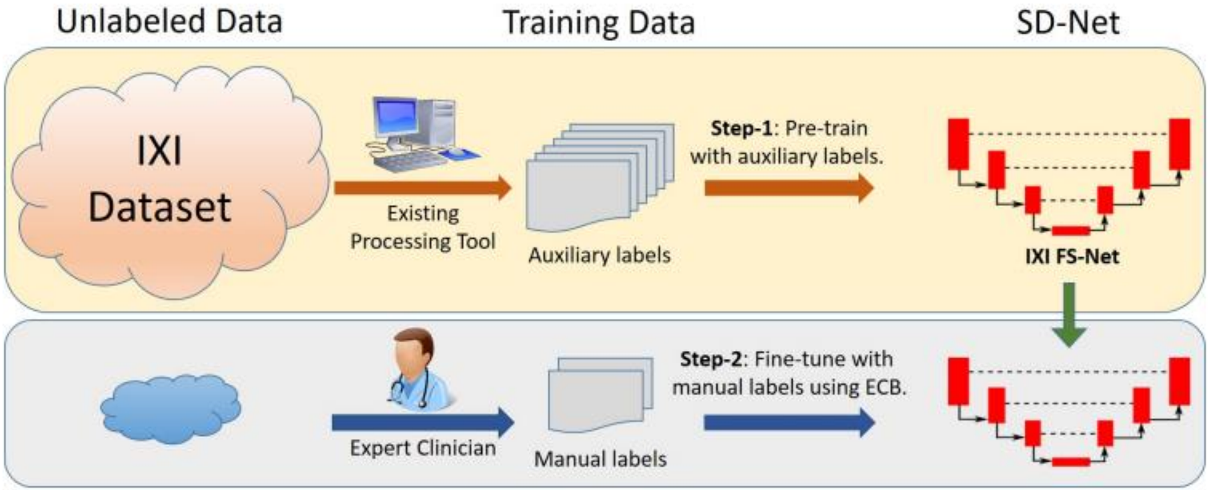
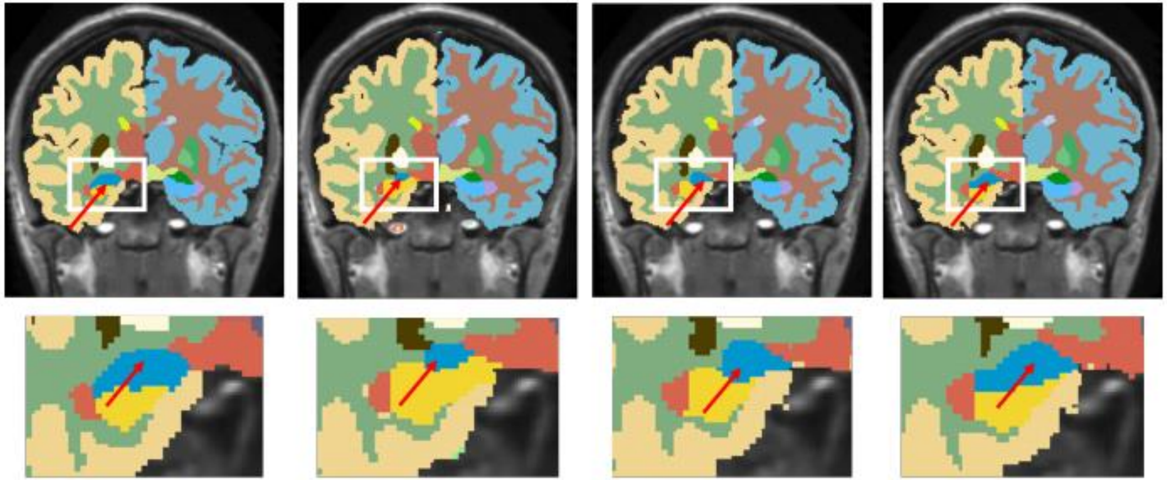


Fig. 1: Illustration of the different steps involved in training of F-CNNs with surplus auxiliary labeled data and limited manually labeled data.

Roy, A. G., Conjeti, S., Sheet, D., Katouzian, A., Navab, N., & Wachinger, C. (2017). Error corrective boosting for learning fully convolutional networks with limited data. MICCAI.

“Error-corrective Boosting”



(a) Ground Truth (b) SD-Net (only manual) (c) SD-Net (Normal FT) (d) SD-Net (ECB)

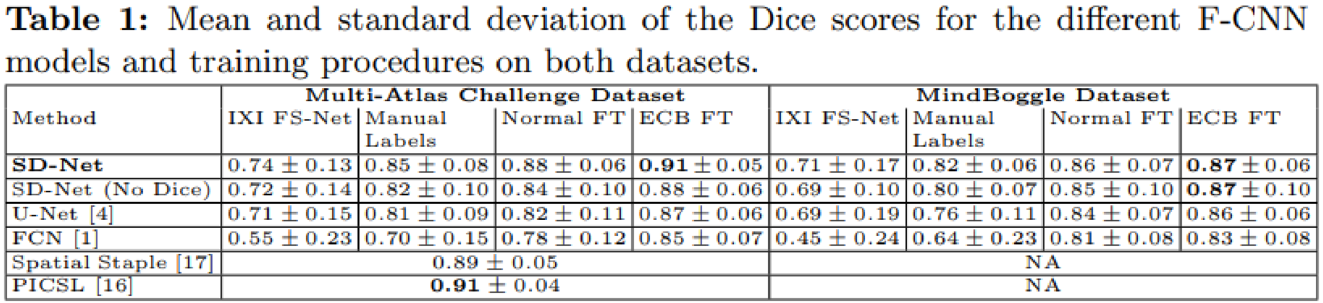


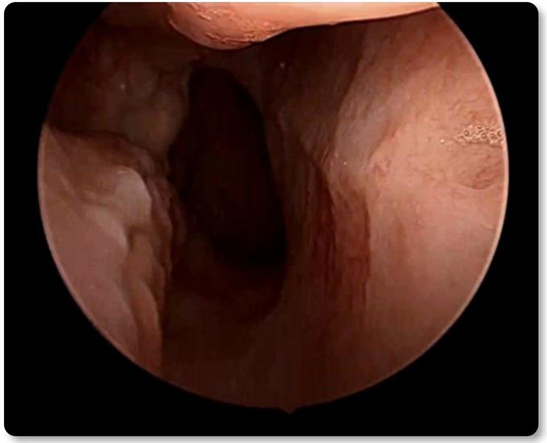
Table 1: Mean and standard deviation of the Dice scores for the different F-CNN models and training procedures on both datasets.

Method	Multi-Atlas Challenge Dataset				MindBoggle Dataset			
	IXI FS-Net	Manual Labels	Normal FT	ECB FT	IXI FS-Net	Manual Labels	Normal FT	ECB FT
SD-Net	0.74 ± 0.13	0.85 ± 0.08	0.88 ± 0.06	**0.91** ± 0.05	0.71 ± 0.17	0.82 ± 0.06	0.86 ± 0.07	**0.87** ± 0.06
SD-Net (No Dice)	0.72 ± 0.14	0.82 ± 0.10	0.84 ± 0.10	0.88 ± 0.06	0.69 ± 0.10	0.80 ± 0.07	0.85 ± 0.10	**0.87** ± 0.10
U-Net [4]	0.71 ± 0.15	0.81 ± 0.09	0.82 ± 0.11	0.87 ± 0.06	0.69 ± 0.19	0.76 ± 0.11	0.84 ± 0.07	0.86 ± 0.06
FCN [1]	0.55 ± 0.23	0.70 ± 0.15	0.78 ± 0.12	0.85 ± 0.07	0.45 ± 0.24	0.64 ± 0.23	0.81 ± 0.08	0.83 ± 0.08
Spatial Staple [17]	0.89 ± 0.05				NA			
PICSL [16]	**0.91** ± 0.04				NA			

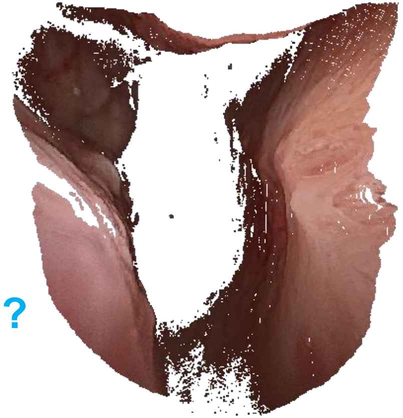
Roy, A. G., Conjeti, S., Sheet, D., Katouzian, A., Navab, N., & Wachinger, C. (2017). Error corrective boosting for learning fully convolutional networks with limited data. MICCAI.

Estimating 3D Geometry from Endoscopic Video

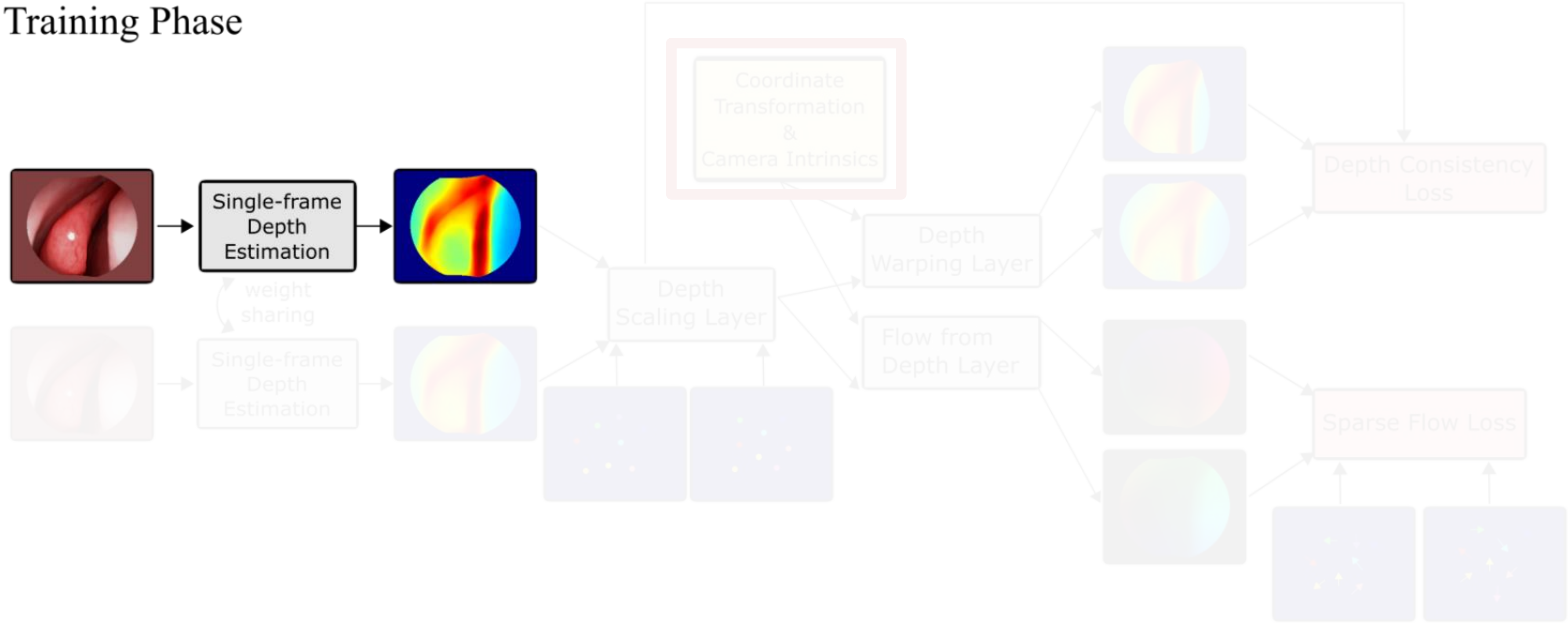
How to obtain 3D representations from endoscopic video?



… some AI maybe?



Training Phase



Single-frame Depth Estimation

weight sharing

Single-frame Depth Estimation

Depth Scaling Layer

Coordinate Transformation & Camera Intrinsics

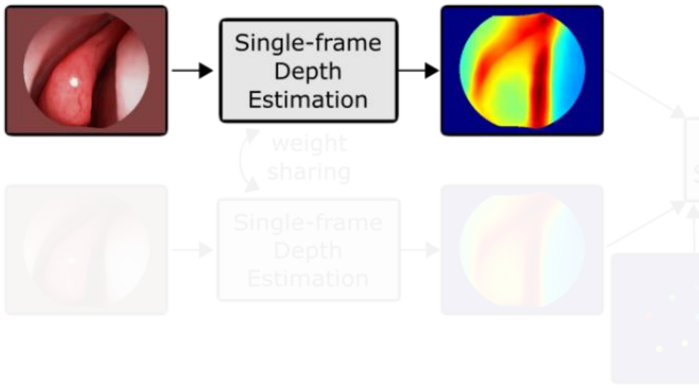
Depth Warping Layer

Flow from Depth Layer

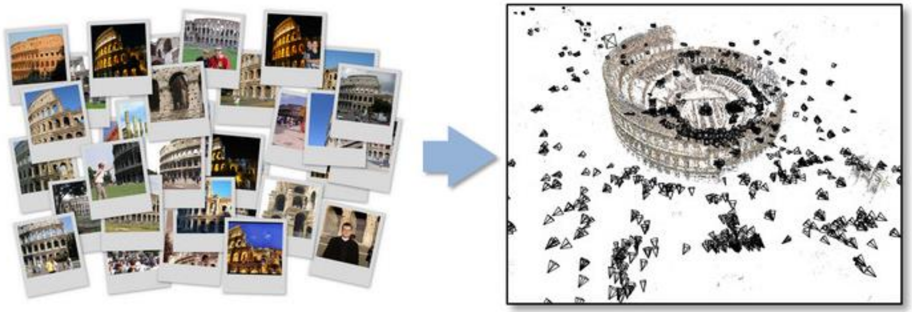
Depth Consistency Loss

Sparse Flow Loss

Training Phase


Single-frame Depth Estimation
weight sharing
Single-frame Depth Estimation
Coordinate Transformation & Camera Intrinsics
Depth Consistency

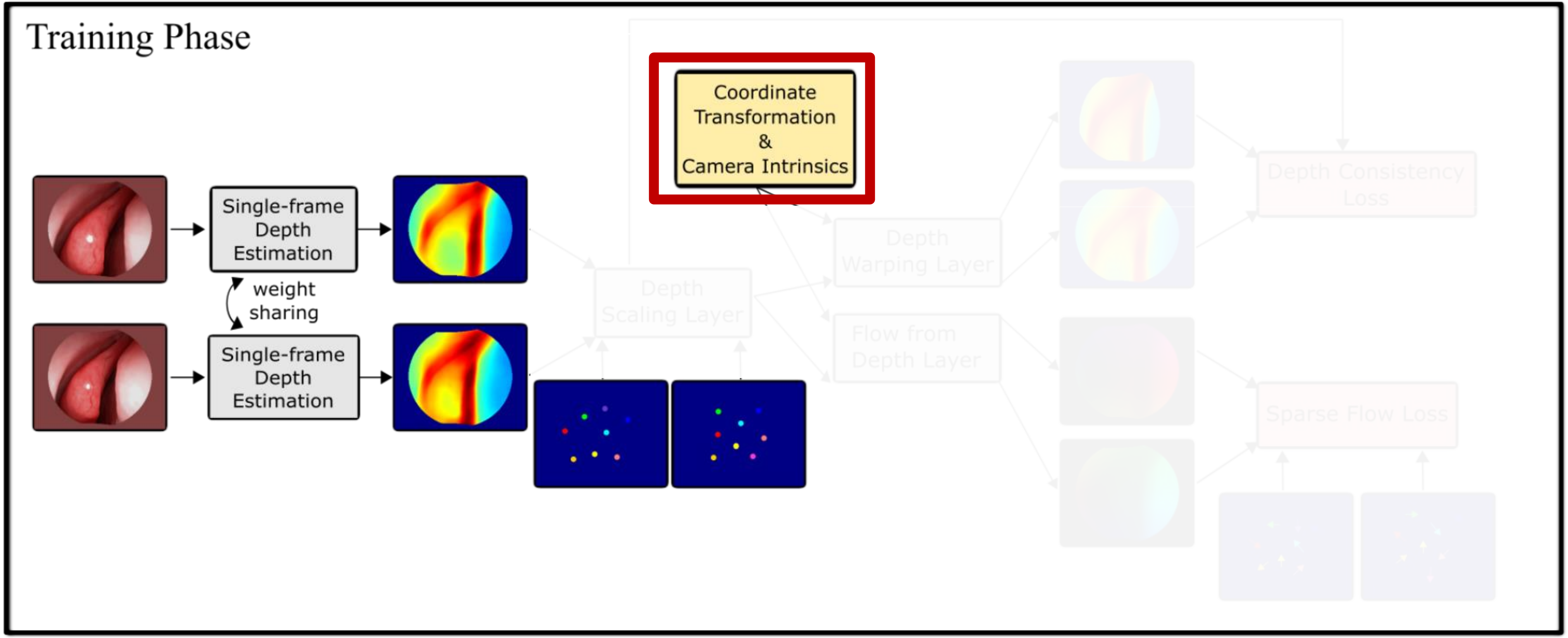
Classical – Structure from Motion



- Feature matching
- Triangulation and bundle adjustment

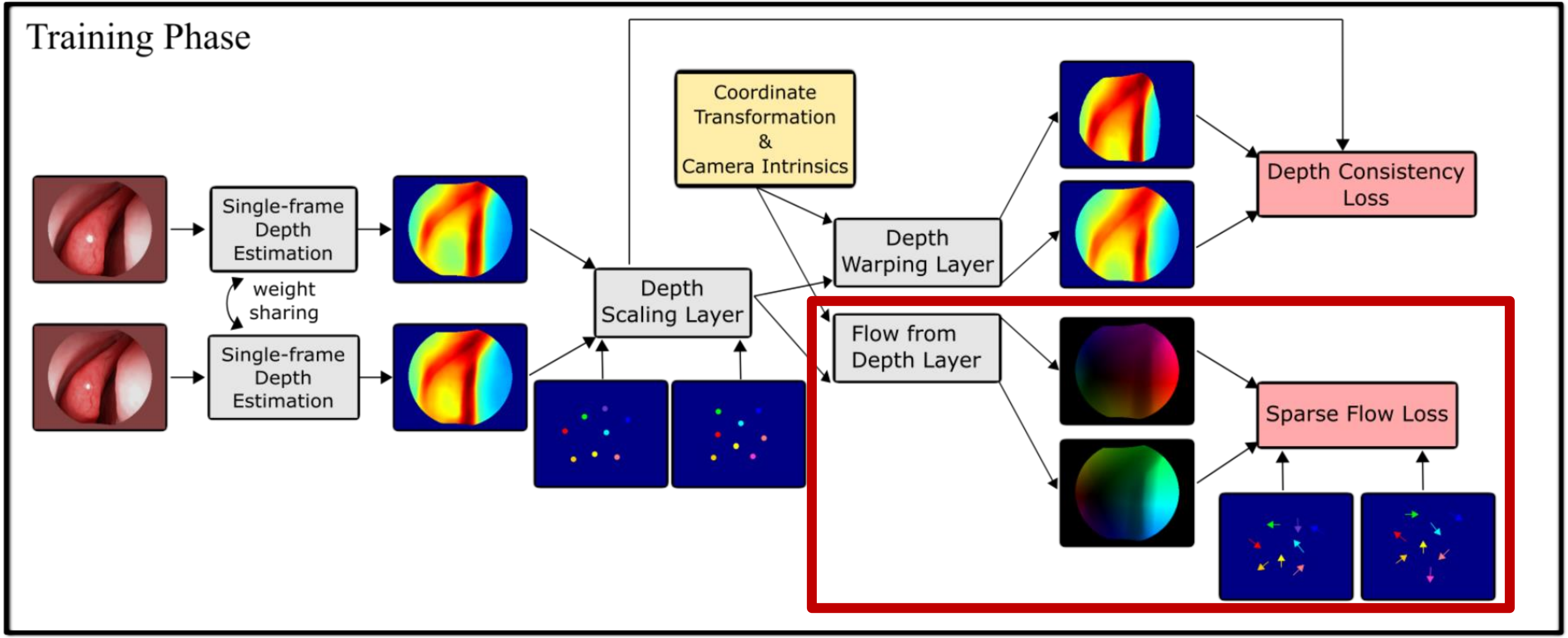
→ Reconstruction from acquired images

Snavely, N., Seitz, S. M., & Szeliski, R. (2006, July). Photo tourism: exploring photo collections in 3D. In ACM transactions on graphics (TOG) (Vol. 25, No. 3, pp. 835-846). ACM.



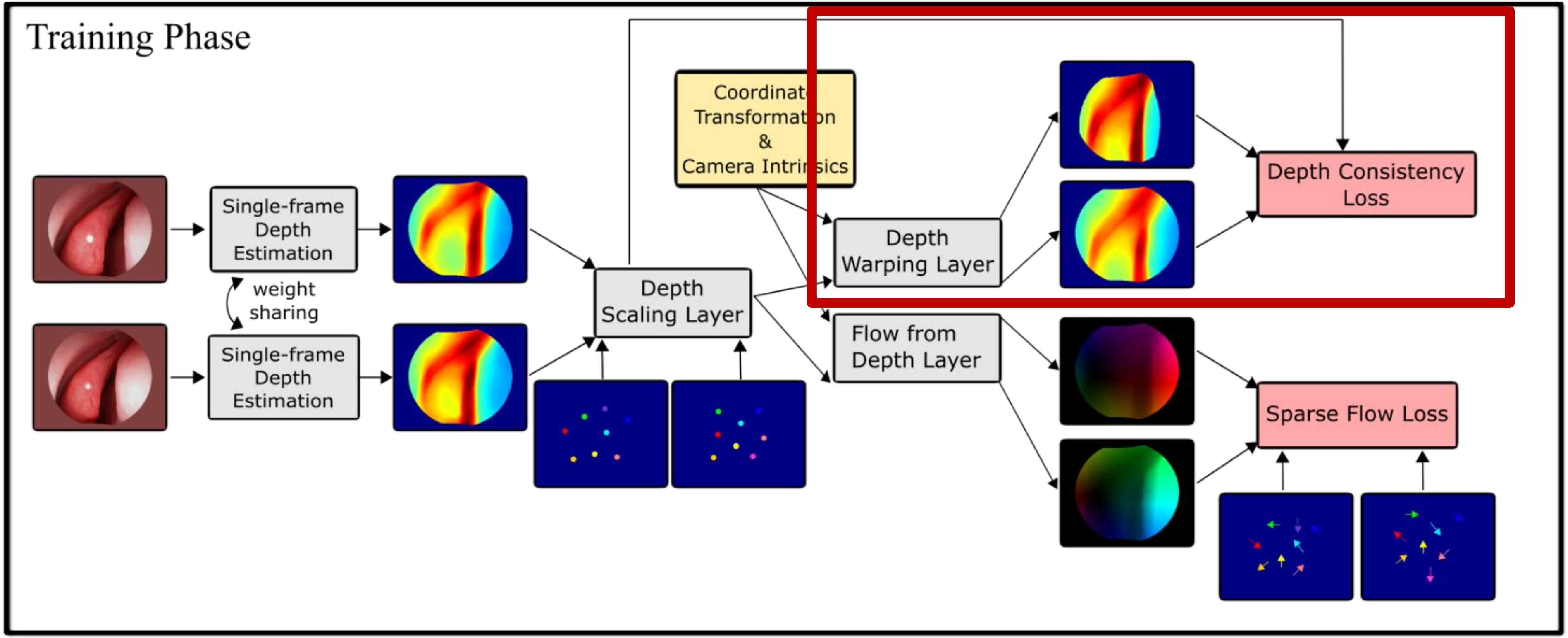
Structure from motion (SfM)-based self-supervision

- Run SfM on short video sequence (15 to 30 frames)
- Siamese network → Process multiple frames



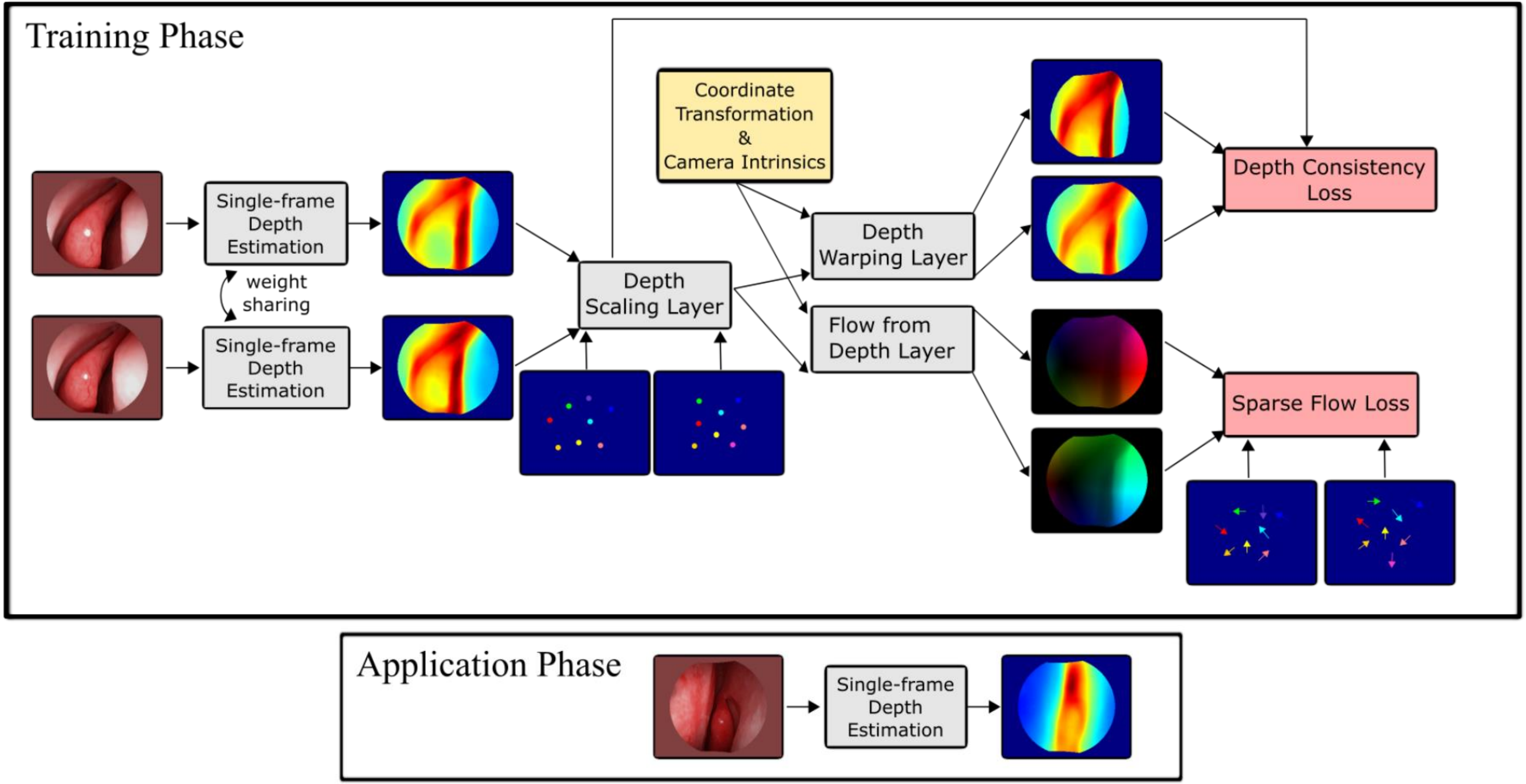
Sparse Flow Loss

- True 2D optical flow from 3D reconstructed points (SfM)
- Estimated optical flow from depth prediction



Depth Consistency Loss

- Differentiable warping operation to warp estimated depth into neighbor frame
- Enforces consistency among predictions



Training Phase

Single-frame Depth Estimation

weight sharing

Single-frame Depth Estimation

Depth Scaling Layer

Coordinate Transformation & Camera Intrinsics

Depth Warping Layer

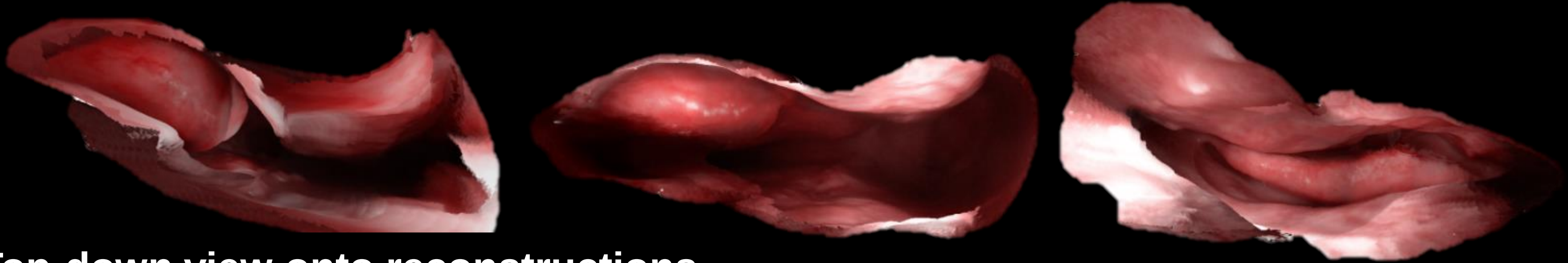
Flow from Depth Layer

Depth Consistency Loss

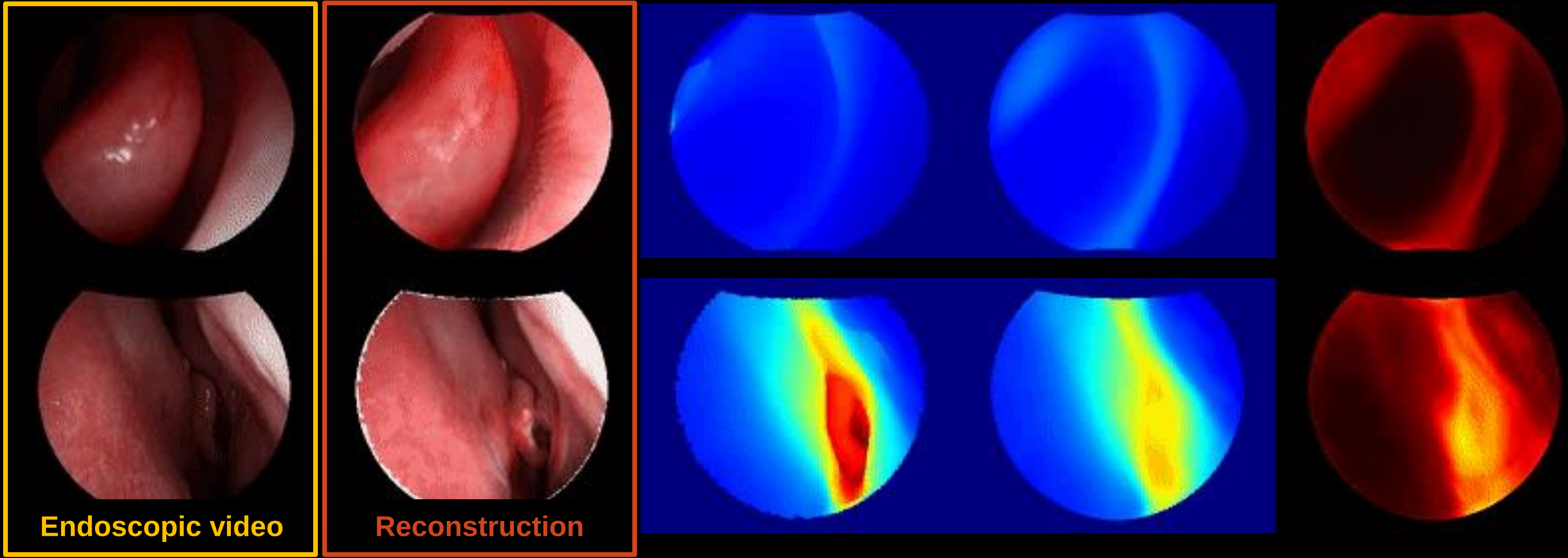
Sparse Flow Loss

Application Phase

Single-frame Depth Estimation



Top-down view onto reconstructions



Fly-through the reconstructions

Unsupervised and Self-supervised Learning

Questions?