Joint Unsupervised Learning of Deep Representations and Image Clusters

Jianwei Yang, Devi Parikh, Dhruv Batra

[arXiv] [GitHub]

Image Processing Group
Signal Theory and Communications Department
Universitat Politècnica de Catalunya. BARCELONATECH

Slides by Albert Jiménez [GDoc]
Computer Vision Reading Group (23/09/2016)

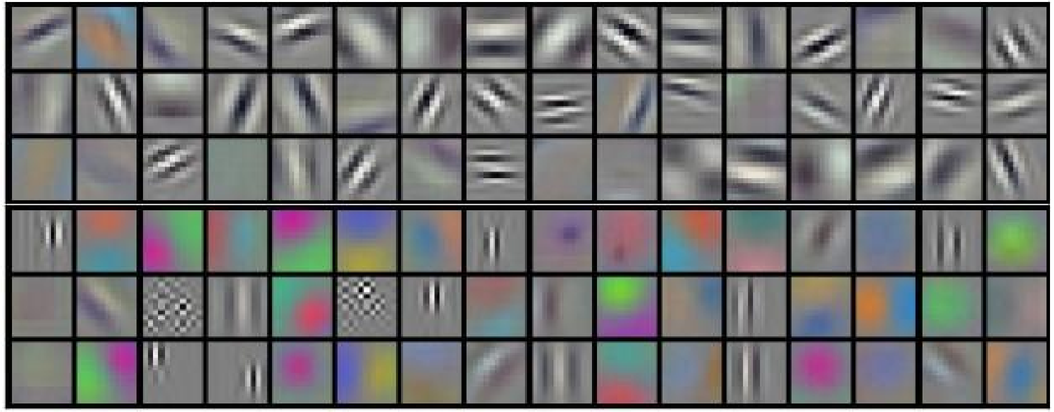
1.
Introduction

The main idea

Main Idea (1)

Joint Unsupervised Learning of representations and image clusters

Learning good image representations is beneficial to image clustering



Main Idea (2)

Joint Unsupervised Learning of representations and image clusters

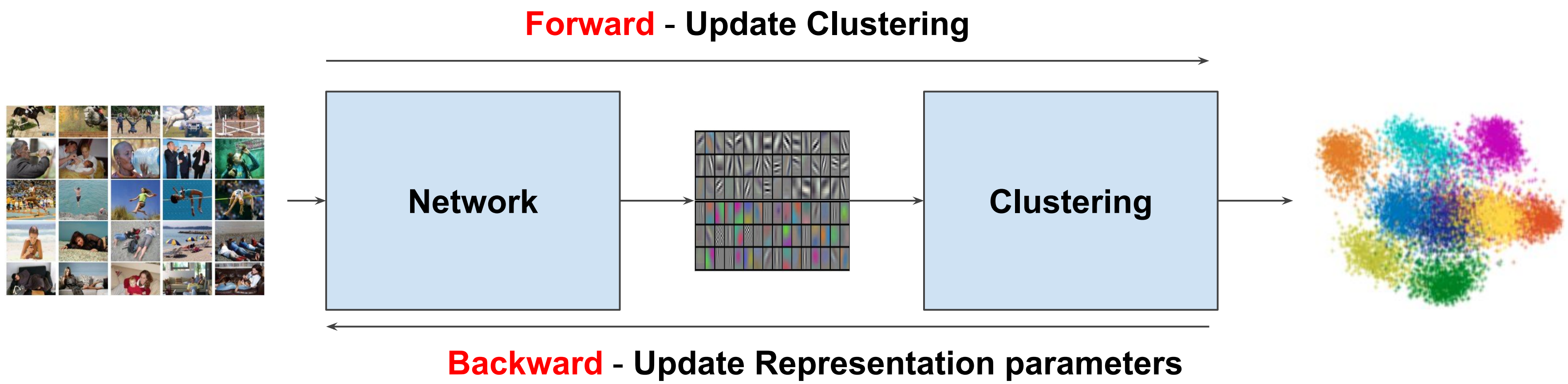
Clustering provide supervisory signals for image representation

Main Idea (3)

Joint Unsupervised Learning of representations and image clusters

Integrate both processes into a single model

- Unified triplet loss
- End-to-end training



2.
Clustering

Agglomerative Clustering

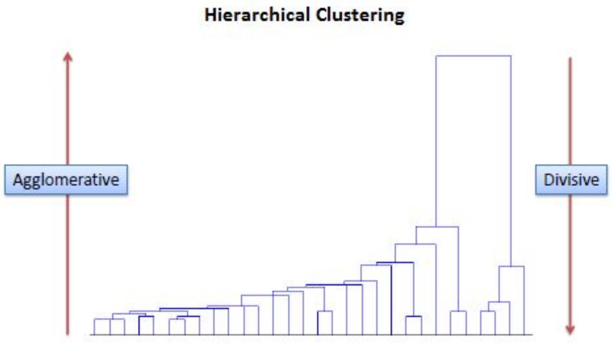
Agglomerative Clustering

What?

- At the beginning each image is a cluster
- Merge two clusters at each timestep

$$\{\mathcal{C}_a, \mathcal{C}_b\} = \underset{\mathcal{C}_i, \mathcal{C}_j \in \mathbf{C}, i \neq j}{\operatorname{argmax}} \boxed{\mathcal{A}(\mathcal{C}_i, \mathcal{C}_j)}$$

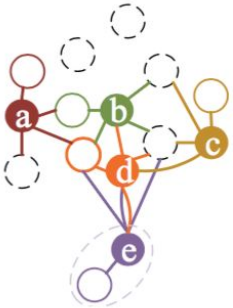
Affinity Measure



Agglomerative Clustering

Why?

- Recurrent process → Implemented in recurrent framework
- Begins with over-clustering (overcome bad initial representations)
- Clusters are merged as better representations are learned



Agglomerative Clustering

Affinity Measure

- Directed graph $G =< \mathcal{V}, \mathcal{E} >$,
- Affinity matrix corresponding to the edge set $\boldsymbol{W} \in \mathbb{R}^{n_s \times n_s}$

$$\boldsymbol{W}(i,j) = \begin{cases} exp(-\frac{||\boldsymbol{x}_i - \boldsymbol{x}_j||_2^2}{\sigma^2}), & \text{if } \boldsymbol{x}_j \in \mathcal{N}_i^{K_s} \\ 0, & \text{otherwise} \end{cases}$$

$$\sigma^2 = \frac{a}{n_s K_s} \sum_{\boldsymbol{x}_i \in \mathbf{X}} \sum_{\boldsymbol{x}_j \in \mathcal{N}_i^{K_s}} ||\boldsymbol{x}_i - \boldsymbol{x}_j||_2^2$$

K_s → # Nearest Neighbors

N_i^{Ks} → K_s NN of X_i

$x_{i,j}$ → Image Representation (vertex)

ns → # Samples

a → Design Parameter = 1

Agglomerative Clustering

Affinity Measure

- Affinity matrix corresponding to the edge set: $\boldsymbol{W} \in \mathbb{R}^{n_s \times n_s}$
- Affinity measure:

$$\begin{aligned}\mathcal{A}(\mathcal{C}_i, \mathcal{C}_j) &= \mathcal{A}(\mathcal{C}_j \rightarrow \mathcal{C}_i) + \mathcal{A}(\mathcal{C}_i \rightarrow \mathcal{C}_j) \\ &= \frac{1}{|\mathcal{C}_i|^2} \mathbf{1}_{|\mathcal{C}_i|}^{T} \boldsymbol{W}_{\mathcal{C}_i,\mathcal{C}_j} \boldsymbol{W}_{\mathcal{C}_j,\mathcal{C}_i} \mathbf{1}_{|\mathcal{C}_i|} \\ &+ \frac{1}{|\mathcal{C}_j|^2} \mathbf{1}_{|\mathcal{C}_j|}^{T} \boldsymbol{W}_{\mathcal{C}_j,\mathcal{C}_i} \boldsymbol{W}_{\mathcal{C}_i,\mathcal{C}_j} \mathbf{1}_{|\mathcal{C}_j|}\end{aligned}$$

W. Zhang, X. Wang, D. Zhao, and X. Tang. Graph degree linkage: Agglomerative clustering on a directed graph.

3.
System Formulation

Joint Optimization

System Formulation

Recurrent Framework

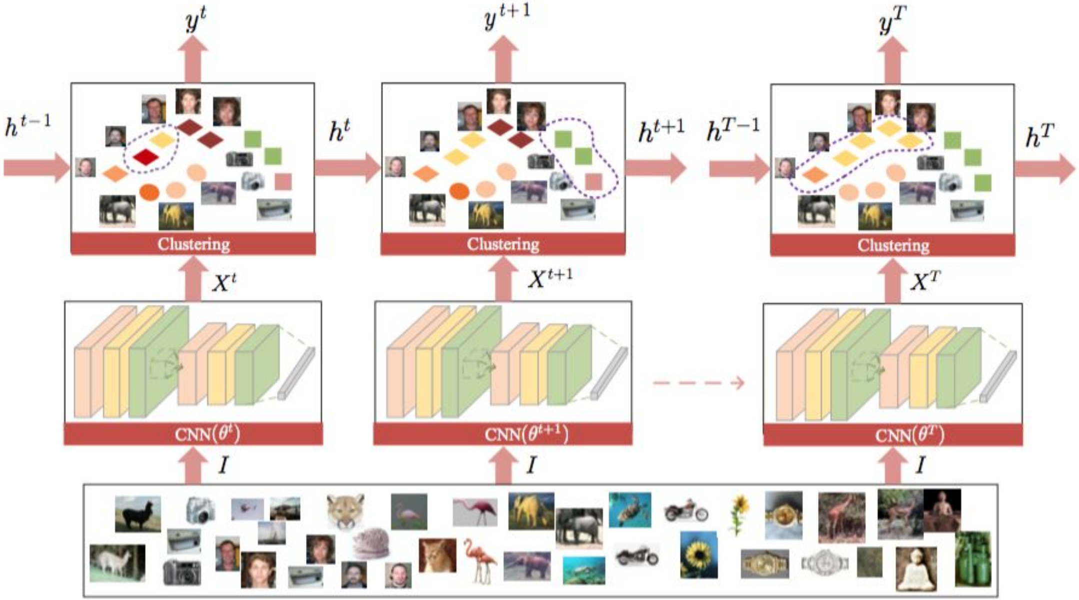
At timestep t:

$h^t \rightarrow$ Image cluster labels
$y^t \rightarrow$ Output
$X^t \rightarrow$ Image representations

$$\boldsymbol{X}^t = f_r(\boldsymbol{I}|\boldsymbol{\theta}^t)$$

$$\boldsymbol{h}^t = f_m(\boldsymbol{X}^t, \boldsymbol{h}^{t-1})$$

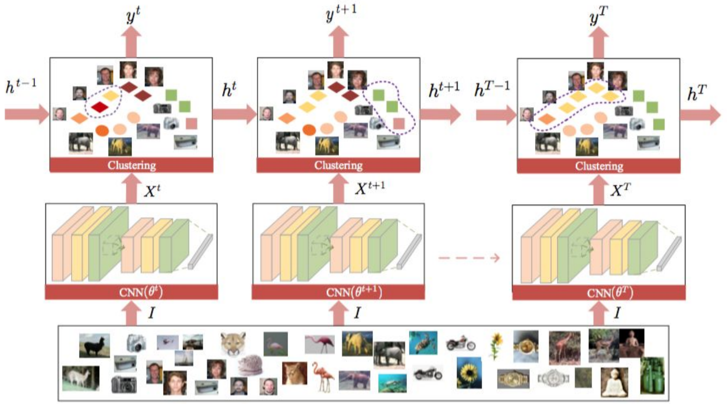
$$\boldsymbol{y}^t = f_o(\boldsymbol{h}^t) = \boldsymbol{h}^t$$



System Formulation

Recurrent Framework

- Unrolling strategy:
  - Complete
  - **Partial**
    - Split the overall T timesteps into multiple periods
    - In each period we merge a number of clusters and update CNN parameters



System Formulation

Algorithm

Input:

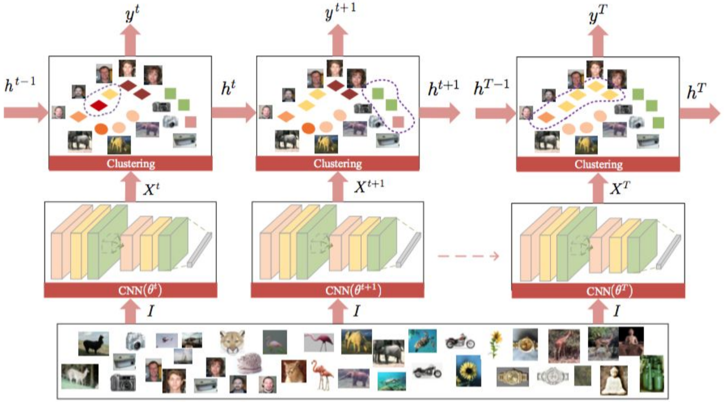
$\boldsymbol{I}$: = collection of image data;

n_c^*: = target number of clusters;

Output:

$\boldsymbol{y}^*, \boldsymbol{\theta}^*$: = final image labels and CNN parameters;

```
t ← 0; p ← 0
Initialize θ and y
repeat
    Update y^t to y^{t+1} by merging two clusters
    if t = t_p^e then
        Update θ^p to θ^{p+1} by training CNN
        p ← (p + 1)
    end if
    t ← t + 1
until Cluster number reaches n_c^*
y* ← y^t; θ* ← θ^p
```



Unrolling rate η: control the number of timesteps

n_c^s : number of clusters at the start of p

Number of timesteps: $n_p = ceil(\eta \times n_c^s)$

System Formulation

Objective Function

- Accumulate the losses from all the timesteps

$$\mathcal{L}(\{\boldsymbol{y}^1, ..., \boldsymbol{y}^T\}, \{\boldsymbol{\theta}^1, ..., \boldsymbol{\theta}^T\}|\boldsymbol{I}) = \sum_{t=1}^{T} \mathcal{L}^t(\boldsymbol{y}^t, \boldsymbol{\theta}^t|\boldsymbol{y}^{t-1}, \boldsymbol{I})$$

- For y^0 we take each image as a cluster, at time t we merge 2 clusters given y^{t-1}
- In conventional agglomerative clustering
  - 2 clusters are selected by maximal Affinity measure over pairs of clusters $\{\mathcal{C}_a, \mathcal{C}_b\} = \underset{\mathcal{C}_i, \mathcal{C}_j \in \boldsymbol{\mathcal{C}}, i \neq j}{\operatorname{argmax}} \boldsymbol{\mathcal{A}}(\mathcal{C}_i, \mathcal{C}_j)$
- In this approach
  - Consider the local structure surrounding the clusters

System Formulation

Objective Function

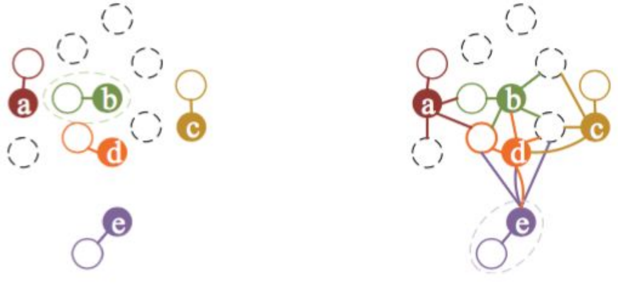
- Assuming that from y^{t-1} to y^t we have merged a cluster C_i^t and its NN

$$\mathcal{L}^t(\boldsymbol{y}^t, \boldsymbol{\theta}^t | \boldsymbol{y}^{t-1}, \boldsymbol{I}) = \underbrace{-\boldsymbol{\mathcal{A}}(\mathcal{C}_i^t, \mathcal{N}_{\mathcal{C}_i^t}^{K_c}[1])}_{\text{Affinity between cluster } C_i \text{ and NN}} \underbrace{- \frac{\lambda}{(K_c - 1)} \sum_{k=2}^{K_c} \left(\boldsymbol{\mathcal{A}}(\mathcal{C}_i^t, \mathcal{N}_{\mathcal{C}_i^t}^{K_c}[1]) - \boldsymbol{\mathcal{A}}(\mathcal{C}_i^t, \mathcal{N}_{\mathcal{C}_i^t}^{K_c}[k]) \right)}_{\text{Difference of affinity between cluster } C_i \text{ and NN and affinities of } C_i \text{ with its Kc neighbor clusters}}$$

System Formulation

Objective Function

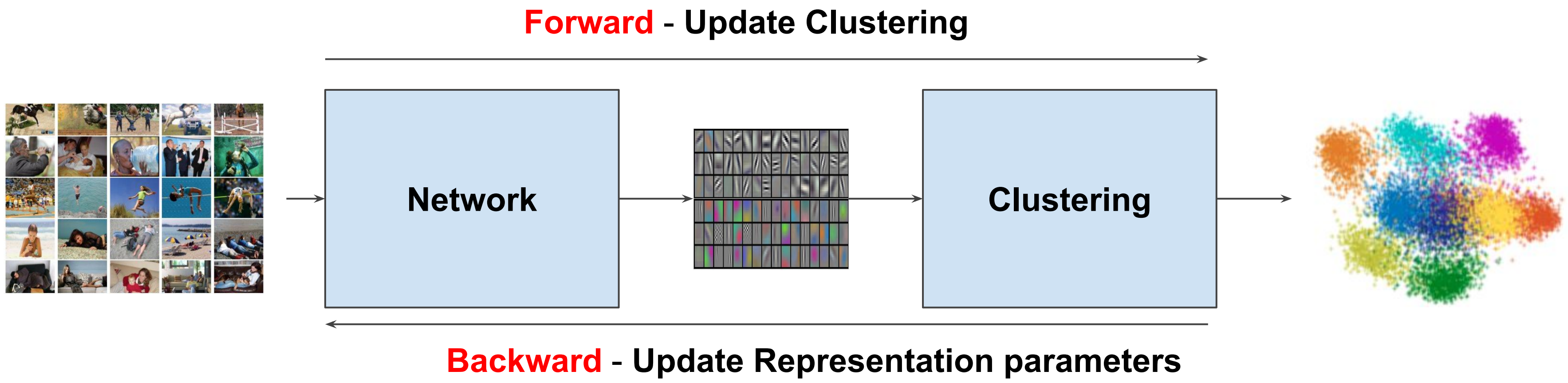
- Introducing this loss term: $$-\frac{\lambda}{(K_c-1)}\sum_{k=2}^{K_c}\left(\mathcal{A}(C_i^t, \mathcal{N}_{C_i^t}^{K_c}[1]) - \mathcal{A}(C_i^t, \mathcal{N}_{C_i^t}^{K_c}[k])\right)$$
    - Consider the local structure surrounding the clusters
    - Allows to write the loss in terms of triplets



(a) (b)

System Formulation

Forward/Backward Pass



System Formulation

Forward Pass

- In the forward pass of the *p-th* period $p \in (1, \dots, P)$, update the clusters with the network parameters fixed

$$\mathcal{L}^p(\mathcal{Y}^p|\boldsymbol{\theta}^p, \boldsymbol{I}) = \sum_{t=t_p^s}^{t_p^e} \mathcal{L}^t(\boldsymbol{y}^t|\boldsymbol{\theta}^p, \boldsymbol{y}^{t-1}, \boldsymbol{I})$$

$[t_p^s, t_p^e]$ is the corresponding timesteps in period p.

Overall loss for period p

System Formulation

Backward Pass

- In the forward pass of the *p-th* period $p \in (1, \dots, P)$, we have merged a number of clusters $\{[\mathcal{C}_*^t, \mathcal{N}_{\mathcal{C}_*^t}^{K_c}[1]]\}, t \in \{t_p^s, \dots, t_p^e\}$
- In the backward pass we aim to derive the optimal parameters $\boldsymbol{\theta}$ to minimize the losses generated in the forward pass.

$$\mathcal{L}(\boldsymbol{\theta}|\{\mathcal{Y}_*^1, \dots, \mathcal{Y}_*^p\}, \boldsymbol{I}) = \sum_{k=1}^{p} \mathcal{L}^k(\boldsymbol{\theta}|\mathcal{Y}_*^k, \boldsymbol{I}) = -\frac{\lambda}{K_c - 1} \sum_{t=1}^{t_p^e} \sum_{k=2}^{K_c} \left(\lambda' \boldsymbol{\mathcal{A}}(\mathcal{C}_*^t, \mathcal{N}_{\mathcal{C}_*^t}^{K_c}[1]) - \boldsymbol{\mathcal{A}}(\mathcal{C}_*^t, \mathcal{N}_{\mathcal{C}_*^t}^{K_c}[k])\right)$$

We accumulate the losses of p periods

$$\lambda' = (1 + 1/\lambda)$$

System Formulation

From cluster based loss to sample based

- Loss defined on clusters
  - Need the entire dataset to estimate
  - Difficult to use batch-based optimization

$$\mathcal{L}(\boldsymbol{\theta}|\{\boldsymbol{\mathcal{Y}}_*^1, ..., \boldsymbol{\mathcal{Y}}_*^p\}, \boldsymbol{I}) = \sum_{k=1}^{p} \mathcal{L}^k(\boldsymbol{\theta}|\boldsymbol{\mathcal{Y}}_*^k, \boldsymbol{I}) = -\frac{\lambda}{K_c - 1} \sum_{t=1}^{t_p^e} \sum_{k=2}^{K_c} \left(\lambda' \boldsymbol{\mathcal{A}}(\mathcal{C}_*^t, \mathcal{N}_{\mathcal{C}_*^t}^{K_c}[1]) - \boldsymbol{\mathcal{A}}(\mathcal{C}_*^t, \mathcal{N}_{\mathcal{C}_*^t}^{K_c}[k])\right)$$

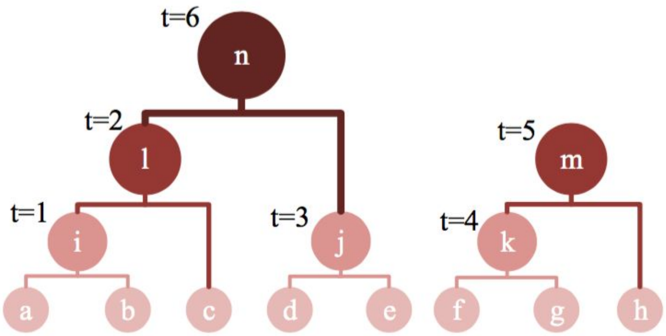
- Sample based loss approximation

$$\mathcal{L}(\boldsymbol{\theta}|\boldsymbol{y}_*^{t_p^e}, \boldsymbol{I}) = -\frac{\lambda}{K_c - 1} \sum_{i,j,k} (\gamma \boldsymbol{\mathcal{A}}(\boldsymbol{x}_i, \boldsymbol{x}_j) - \boldsymbol{\mathcal{A}}(\boldsymbol{x}_i, \boldsymbol{x}_k))$$

System Formulation

From cluster based loss to sample based

- Sample based loss intuition
  - Agglomerative clustering starts with each datapoint as a cluster
  - Clusters at higher level hierarchy are formed by merging lower level clusters



System Formulation

From cluster based loss to sample based

- Affinities between clusters can be expressed in terms of affinities with datapoints

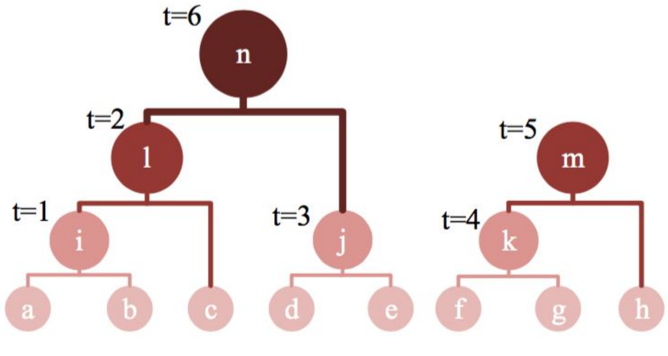
- **t=2**: $\mathcal{C}_*^2 = \mathcal{C}_i, \mathcal{N}_{\mathcal{C}_*^2}^2[1] = \mathcal{C}_c$ and $\mathcal{N}_{\mathcal{C}_*^2}^2[2] = \mathcal{C}_d$. We have

$$\begin{aligned}\mathcal{L}(\boldsymbol{\theta}|\{\boldsymbol{y}_*^1, \boldsymbol{y}_*^2\}, I) &= \mathcal{L}(\boldsymbol{\theta}|\boldsymbol{y}_*^1, I) \\ &\quad - (\lambda' \boldsymbol{\mathcal{A}}(\mathcal{C}_i, \mathcal{C}_c) - \boldsymbol{\mathcal{A}}(\mathcal{C}_i, \mathcal{C}_d))\end{aligned} \tag{24}$$

Since $\mathcal{C}_i = \mathcal{C}_a \cup \mathcal{C}_b$, we base on Eq. (19) for approximation

$$\begin{aligned}\boldsymbol{\mathcal{A}}(\mathcal{C}_i, \mathcal{C}_c) &= \boldsymbol{\mathcal{A}}(\mathcal{C}_a \to \mathcal{C}_c) + \boldsymbol{\mathcal{A}}(\mathcal{C}_b \to \mathcal{C}_c) \\ &\quad + \frac{1}{2}\boldsymbol{\mathcal{A}}(\mathcal{C}_c \to \mathcal{C}_a) + \frac{1}{2}\boldsymbol{\mathcal{A}}(\mathcal{C}_c \to \mathcal{C}_b)\end{aligned} \tag{25}$$

$$\begin{aligned}\boldsymbol{\mathcal{A}}(\mathcal{C}_i, \mathcal{C}_d) &= \boldsymbol{\mathcal{A}}(\mathcal{C}_a \to \mathcal{C}_d) + \boldsymbol{\mathcal{A}}(\mathcal{C}_b \to \mathcal{C}_d) \\ &\quad + \frac{1}{2}\boldsymbol{\mathcal{A}}(\mathcal{C}_d \to \mathcal{C}_a) + \frac{1}{2}\boldsymbol{\mathcal{A}}(\mathcal{C}_d \to \mathcal{C}_b)\end{aligned} \tag{26}$$



[For more info check supplement of the paper]

System Formulation

From cluster based loss to sample based

- Finally it has got the form of a weight triplet loss

$$\mathcal{L}(\boldsymbol{\theta}|\boldsymbol{y}_*^{t_p^e}, \boldsymbol{I}) = -\frac{\lambda}{K_c - 1} \sum_{i,j,k} (\gamma \boldsymbol{\mathcal{A}}(\boldsymbol{x}_i, \boldsymbol{x}_j) - \boldsymbol{\mathcal{A}}(\boldsymbol{x}_i, \boldsymbol{x}_k))$$

γ is a weight whose value depends on $\lambda' = (1 + 1/\lambda)$

x_i and x_j are from the same cluster

x_k is from the K_c nearest neighbouring clusters

System Formulation

Optimization

- Given a dataset with n_s samples and n_c number of clusters, $T = n_s - n_c$ timesteps
  - Time consuming optimization in large datasets

- Run a fast algorithm to determine initial clustering
  - Agglomerative clustering via maximum incremental path integral

4.
Experiments

Experiments

Experimental Setup

- They use Caffe/Torch
- Convolutional layers of 50 channels, 5x5 filters, stride = 1, padding = 0
- Pooling layers, 2x2 kernel, stride = 2
- Final size of the output feature map is 10x10

Dataset	*MNIST*	*USPS*	*COIL20*	*COIL100*	*UMist*	*FRGC-v2.0*	*CMU-PIE*	*YTF*
#Samples	70000	11000	1440	7200	575	2462	2856	10000
#Categories	10	10	20	100	20	20	68	41
Image Size	28×28	16×16	128×128	128×128	112×92	32×32	32×32	55×55

Hyper-parameter	K_s	a	K_c	λ	γ	η
Value	20	1.0	5	1.0	2.0	0.9 or $\boxed{0.2}$

Face datasets

Experiments

Experimental Setup

- On top of all the CNNs they append a IP layer (dimension 160)
- Followed by l2 norm layer
- wt-loss = weight triplet loss

Dataset	*COIL20*	*COIL100*	*USPS*	*MNIST-test*	*MNIST-full*	UMist	*FRGC*	*CMU-PIE*	*YTF*
conv1	✓	✓	✓	✓	✓	✓	✓	✓	✓
bn1	✓	✓	✓	✓	✓	✓	✓	✓	✓
relu1	✓	✓	✓	✓	✓	✓	✓	✓	✓
pool1	✓	✓	✓	✓	✓	✓	✓	✓	✓
conv2	✓	✓	–	✓	✓	✓	✓	✓	✓
bn2	✓	✓	–	✓	✓	✓	✓	✓	✓
relu2	✓	✓	–	✓	✓	✓	✓	✓	✓
pool2	✓	✓	–	–	–	✓	✓	✓	✓
conv3	✓	✓	–	–	–	✓	–	–	–
bn3	✓	✓	–	–	–	✓	–	–	–
relu3	✓	✓	–	–	–	✓	–	–	–
pool3	✓	✓	–	–	–	✓	–	–	–
conv4	✓	✓	–	–	–	–	–	–	–
bn4	✓	✓	–	–	–	–	–	–	–
relu4	✓	✓	–	–	–	–	–	–	–
pool4	✓	✓	–	–	–	–	–	–	–
ip1	✓	✓	✓	✓	✓	✓	✓	✓	✓
l2-norm	✓	✓	✓	✓	✓	✓	✓	✓	✓
wt-loss	✓	✓	✓	✓	✓	✓	✓	✓	✓

Experiments

Robustness Analysis

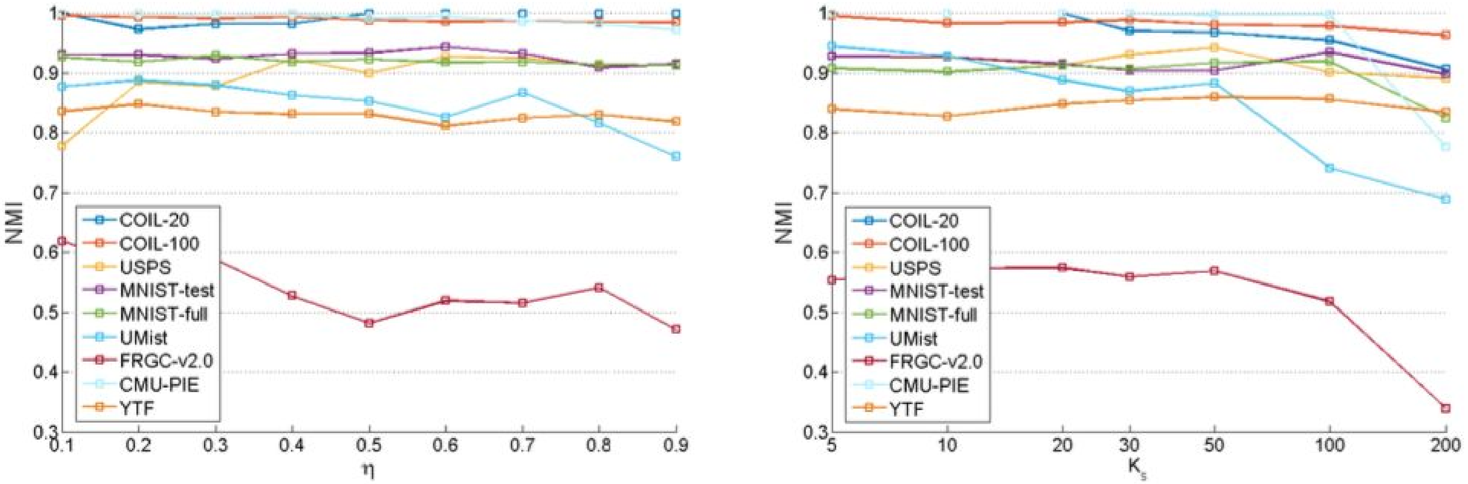


Figure 7: Clustering performance (NMI) with different η (left) and K_s (right).

Experiments

Quantitative Comparison Using Image Intensities as an Input

Dataset	*COIL20*	*COIL100*	*USPS*	*MNIST-test*	*MNIST-full*	*UMist*	*FRGC*	*CMU-PIE*	*YTF*
K-means [39]	0.775	0.822	0.447	0.528	0.500	0.609	0.389	0.549	0.761
SC-NJW [43]	0.860/0.889	0.872/0.854	0.409/0.690	0.528/0.755	0.476	0.727	0.186	0.543	0.752
SC-ST [67]	0.673/0.895	0.706/0.858	0.342/0.726	0.445/0.756	0.416	0.611	0.431	0.581	0.620
SC-LS [3]	0.877	0.833	0.681	0.756	0.706	0.810	0.550	0.788	0.759
N-Cuts [52]	0.768/0.884	0.861/0.823	0.382/0.675	0.386/0.753	0.411	0.782	0.285	0.411	0.742
AC-Link [25]	0.512	0.711	0.579	0.662	0.686	0.643	0.168	0.545	0.738
AC-Zell [70]	0.954/0.911	0.963/0.913	0.774/0.799	0.810/0.768	0.017	0.755	0.351	0.910	0.733
AC-GDL [68]	0.945/0.937	0.954/0.929	0.854/0.824	0.864/0.844	0.017	0.755	0.351	0.934	0.622
AC-PIC [69]	0.950	0.964	0.840	0.853	0.017	0.750	0.415	0.902	0.697
NMF-LP [1]	0.720	0.783	0.435	0.467	0.452	0.560	0.346	0.491	0.720
NMF-D [57]	0.692	0.719	0.286	0.243	0.148	0.500	0.258	0.983/0.910	0.569
TSC-D [61]	-/0.928	-	-	-	-/0.651	-	-	-	-
OURS-SF	**1.000**	0.978	0.858	0.876	0.906	**0.880**	0.566	0.984	**0.848**
OURS-RC	**1.000**	**0.985**	**0.913**	**0.915**	**0.913**	0.877	**0.574**	**1.00**	**0.848**

Measure = NMI = Normalized Mutual Information

Experiments

Quantitative Comparison of the Other State-of-the-Art Using their Representations as Input

Dataset	*COIL20*	*COIL100*	*USPS*	*MNIST-test*	*MNIST-full*	*UMist*	*FRGC*	*CMU-PIE*	*YTF*
K-means [39]	0.926	0.919	0.758	0.908	0.927	0.871	0.636	0.956	0.835
SC-NJW [43]	0.915	0.898	0.753	0.878	0.931	0.833	0.625	0.957	0.789
SC-ST [67]	0.959	0.922	0.741	0.911	0.906	0.847	**0.651**	0.938	0.741
SC-LS [3]	0.950	0.905	0.780	0.912	**0.932**	**0.879**	0.639	0.950	0.802
N-Cuts [52]	0.963	0.900	0.705	0.910	0.930	0.877	0.640	0.995	0.823
AC-Link [25]	0.896	0.884	0.783	0.901	0.918	0.872	0.621	0.990	0.803
AC-Zell [70]	**1.000**	0.989	0.910	0.893	0.919	0.870	0.551	**1.000**	0.821
AC-GDL [68]	**1.000**	0.985	0.913	**0.915**	0.913	0.870	0.574	**1.000**	**0.842**
AC-PIC [69]	**1.000**	**0.990**	**0.914**	0.909	0.907	0.870	0.553	**1.000**	0.829
NMF-LP [1]	0.855	0.834	0.729	0.905	0.926	0.854	0.575	0.690	0.788

Experiments

Transfer Learning

Layer	*data*	top(*ip*)	top-1	top-2
COIL20 → COIL100	0.924	0.927	**0.939**	0.934
COIL100 → COIL20	0.944	0.949	**0.957**	0.951

Layer	*data*	top(*ip*)	top-1	top-2
MNIST-test → USPS	0.874	0.892	0.907	**0.908**
USPS → MNIST-test	0.872	0.873	**0.886**	-

Experiments

Face Verification

- Representations learn from Youtube-Face dataset evaluated on LFW

#Samples	10k	20k	30k	50k	100k
Supervised	**0.737**	**0.746**	0.748	**0.764**	0.770
OURS	0.728	0.743	**0.750**	0.762	0.767

5.
Conclusions

Conclusions

- They combine agglomerative clustering with CNNs and optimize jointly in a recurrent framework.
- Proposal of a partial unrolling strategy to divide the timesteps into multiple periods.
- Obtain more precise image clusters and discriminative representations.
- Able to generalize well across many datasets and tasks.