DeepPermNet: Visual Permutation Learning

Rodrigo Santa Cruz, Basura Fernando, Anoop Cherian and Stephen Gould
The Australian National University, Canberra, Australia
firstname.lastname@anu.edu.au

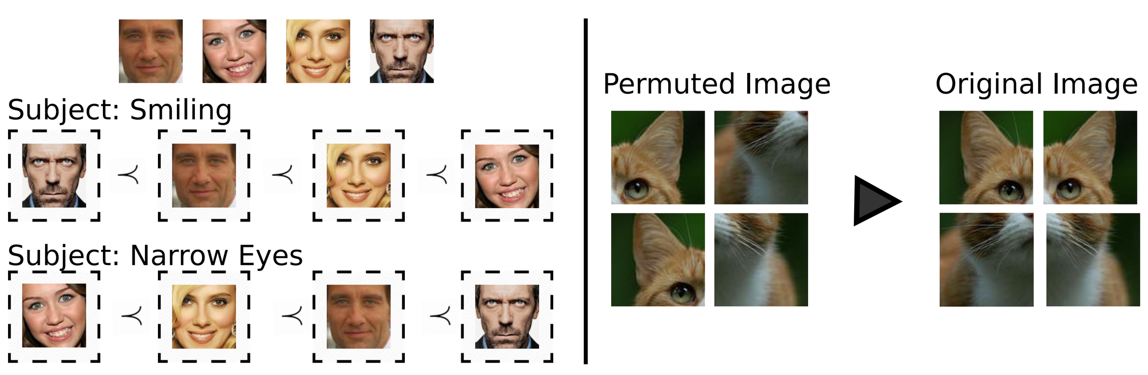
Australian Centre for Robotic Vision
Australian National University

1 - Introduction & Motivation

- Tasks in different fields involve learning a function that can recover the underlying structure of the data.
- Applications: Jigsaw puzzle in computer graphics, DNA and RNA modeling in biology, and re-assembling relics in archeology.
- Computer Vision: image ranking and self-supervised representation learning.
- **We propose the Visual Permutation Learning task as a generic formulation to learn structural concepts intrinsic to natural images and ordered image sequences.**

2 - Visual Permutation Learning

Can we assign a meaningful order to a given collection of images ?



We hypothesize that learning machines need to understand semantic concepts, visual patterns and image features in order to solve these tasks.

Task: Given a permuted image sequence $\tilde{X}$, predict the permutation matrix P such that $P^{-1} = P^T$ recovers the ordered sequence X.

Learning: We propose to learn a parametrized function $f_\theta(\cdot)$ that maps from an image sequence to a doubly stochastic matrix,

$$f_\theta : \tilde{X} \in \mathcal{S}^c \times \mathcal{P}^l \mapsto Q \in \mathcal{B}^l$$

by minimizing the regularized empirical risk,

$$\underset{\theta}{\text{minimize}} \sum_{(X,P)\in\mathcal{D}} \Delta\left(P, f_\theta(\tilde{X})\right) + R(\theta)$$

where $\mathcal{D} = \{(X,P) \mid X \in \mathcal{S}^c \text{ and } P \in \mathcal{P}^l\}$ is a synthetically created training set.

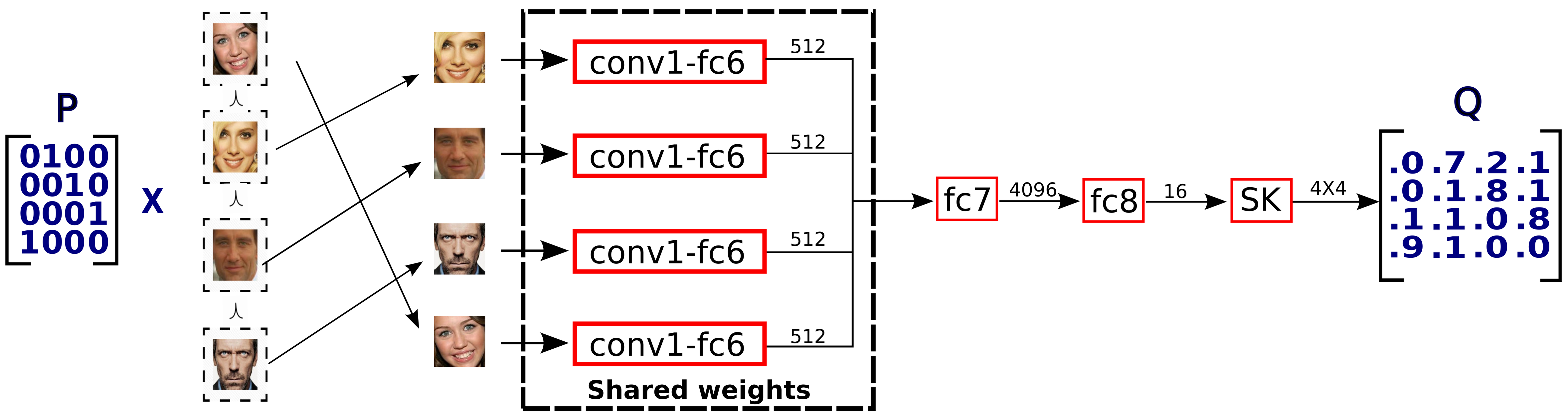
Inference: $X = \hat{P}^T\tilde{X}$

$$\hat{P} \in \underset{\hat{P}\in\mathcal{P}^l}{\text{argmin}} \left\|\hat{P} - Q\right\|_F$$

Note:

- Doubly-stochastic matrices as differentiable relaxation of permutation matrices.
- $\mathcal{D}$ can be generated on-the-fly providing a huge amount of data.
- End-to-End Learning: image representation + permutation problem.

3 - DeepPermNet



Naive Approach: l^2 multi-label classification (i.e. sigmoid outputs + cross entropy loss).

4 - Sinkhorn Norm. Layer

Sinkhorn's theorem: Any non-negative square matrix can be converted to a DSM by alternating between rescaling its rows and columns to one.

$$R_{i,j}(Q) = \frac{Q_{i,j}}{\sum_{k=0}^{d} Q_{i,k}}; \; C_{i,j}(Q) = \frac{Q_{i,j}}{\sum_{k=0}^{d} Q_{k,j}}$$

$$S^n(Q) = \begin{cases} Q, & \text{if } n = 0 \\ C\left(R\left(S^{n-1}(Q)\right)\right), & \text{otherwise.} \end{cases}$$

Note that $S^n(Q)$ is differentiable!

5 - Applications

Permutation Prediction

Method	Length	KT	HS	NE
Naive App.	4	0.859	0.893	0.062
	8	0.774	0.832	0.1
Sinkhorn Norm.	4	0.884	0.906	0.019
	8	0.963	0.973	0.022

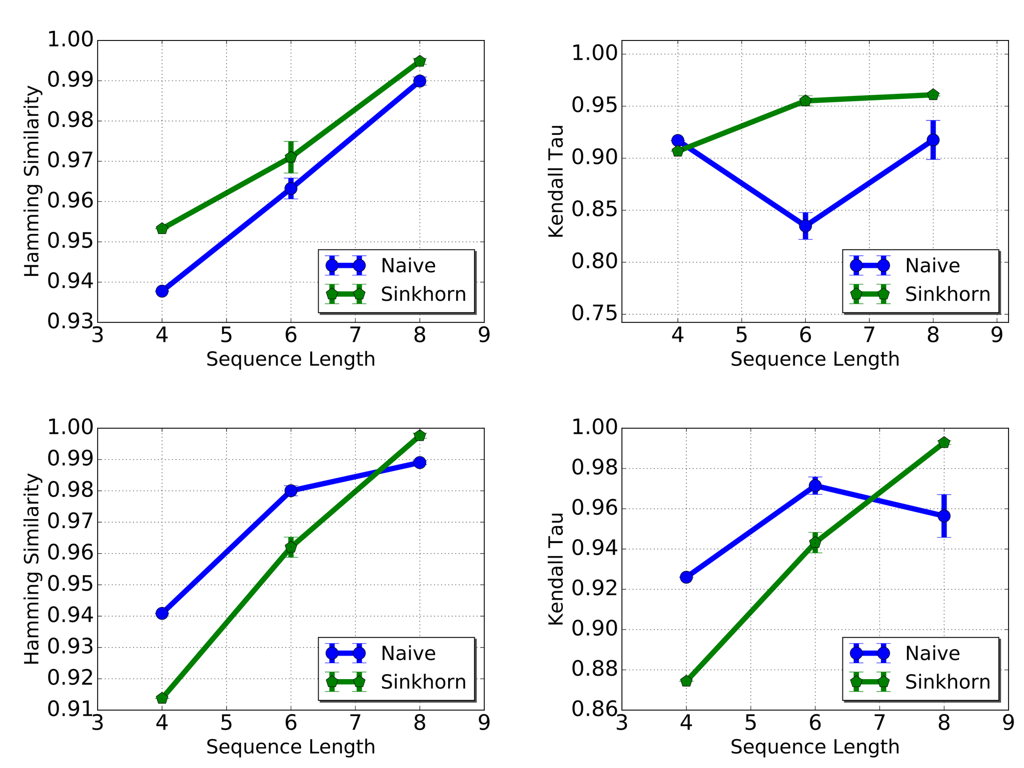


Image Ranking Based on Attributes

Method	Public Figures	OSR
Relative Att.	80.56	88.80
Relative Forest	83.37	90.41
Local Learning	89.72	92.37
End-to-End Loc. Rank.	-	97.02
Deep Relative Att.	94.52	97.77
DeepPermNet	**98.14**	**98.48**

Self-Supervised Repr. Learning

Method	Classification (mAP%)	FRCN Detection (mAP%)	FCN Segmentation (%mIU)
ImageNet	78.2	56.8	48.0
Random Gaussian	53.3	43.4	19.8
Context Prediction	55.3	46.6	-
Temporal coherence	58.4	44.0	-
In-painting	56.5	44.5	29.7
Colorization	65.6	47.9	35.6
Jigsaw Puzzle	68.6	**51.8**	36.1
DeepPermNet	**69.4**	49.5	**37.9**

Ranking Examples & Saliency Maps