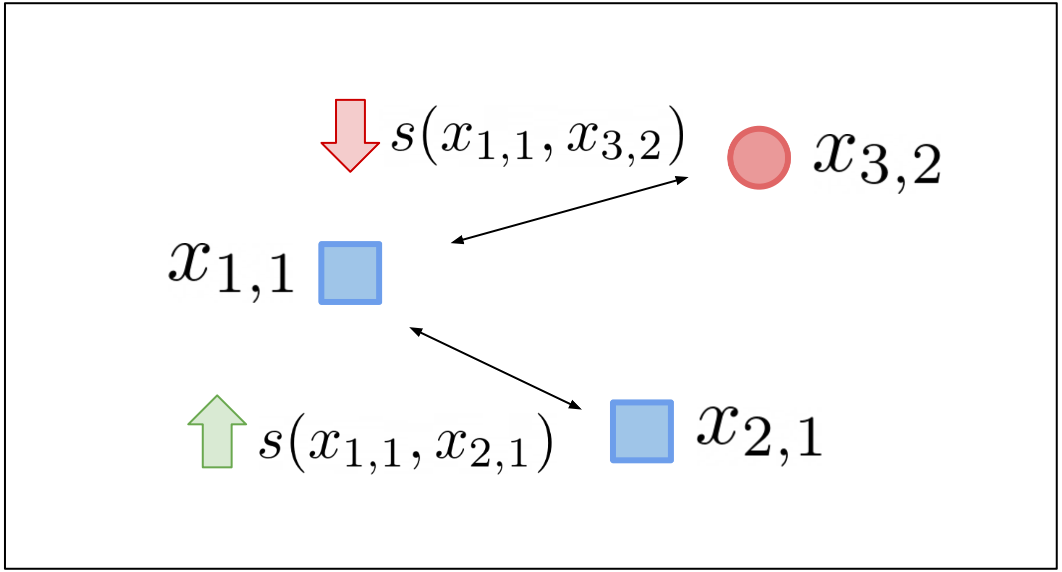
vilynx.

Smooth Proxy-Anchor Loss for Noisy Metric Learning

Carlos Roig, David Varas, Issey Masuda, Juan Carlos Riveiro, Elisenda Bou-Balust

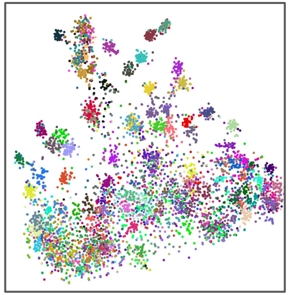
Metric Learning - Introduction



$x_{i,j}$: embedding i of class j

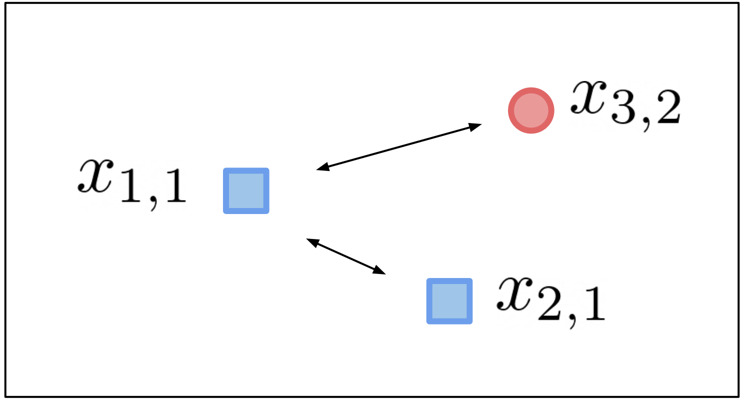
$s(\cdot,\cdot)$: similarity function (e.g. cosine similarity)

Embedding space



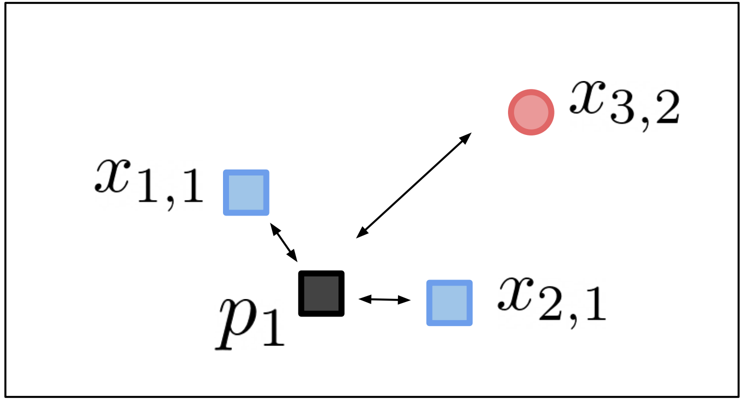
Metric Learning - Pairs vs Proxy methods

Pair-based methods



$$s(x_{1,1}, x_{2,1}) > s(x_{1,1}, x_{3,2})$$

Proxy-based methods



$$s(x_{1,1}, p_1) > s(x_{3,2}, p_1)$$

vilynx.

Metric Learning - Applications

Face Verification

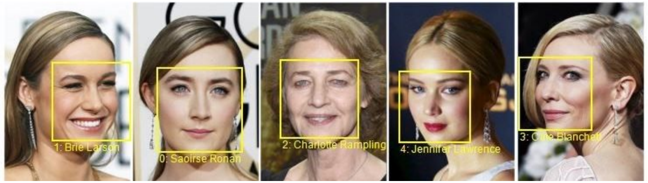
Person Re-Identification

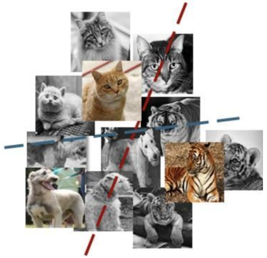
Few-Shot Learning

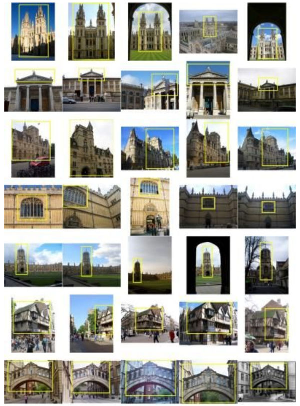
Content Based Image Retrieval

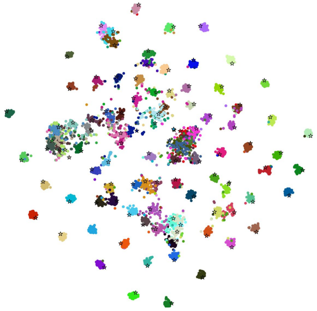
Representation Learning

Require clean data!









Proxy Anchor Loss

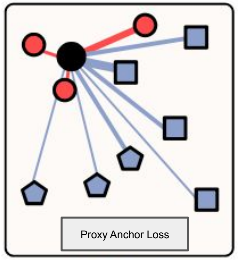


Image batch

Positive proxies

Negative proxies

$$\mathcal{L}(X) = \overbrace{\frac{1}{|P^+|}\sum_{p\in P^+}\log\left(1+\sum_{x\in X_p^+} e^{-\alpha(s(x,p)-\delta)}\right)}^{\text{Positive proxies}} + \overbrace{\frac{1}{|P|}\sum_{p\in P}\log\left(1+\sum_{x\in X_p^-} e^{\alpha(s(x,p)+\delta)}\right)}^{\text{Negative proxies}}$$

Positive proxies of a given sample

Samples corresponding to proxy p

Cosine similarity

Hyperparams

$$X_p^- = X - X_p^+$$

Sungyeon Kim, Dongwon Kim, Minsu Cho, and Suha Kwak. *Proxy anchor loss for deep metric learning*, CVPR 2020.

vilynx.

Smooth Proxy Anchor Loss

$$\mathcal{L}_{smooth}(X) = \frac{1}{|P^+|}\sum_{p\in P^+}\log\left(1+\sum_{x\in X_p^+} w_{x,p}\cdot e^{-\alpha(s(x,p)-\delta)}\right) + \frac{1}{|P|}\sum_{p\in P}\log\left(1+\sum_{x\in X_p^-}(1-w_{x,p})\cdot e^{\alpha(s(x,p)+\delta)}\right)$$

Smoothing function

$$w_{x,p} = \frac{1}{1+e^{-\beta\cdot(c_{x,p}-\lambda)}}$$

Controls the position of the function

Controls the sharpness of the function

Confidence value for sample x of belonging to proxy p

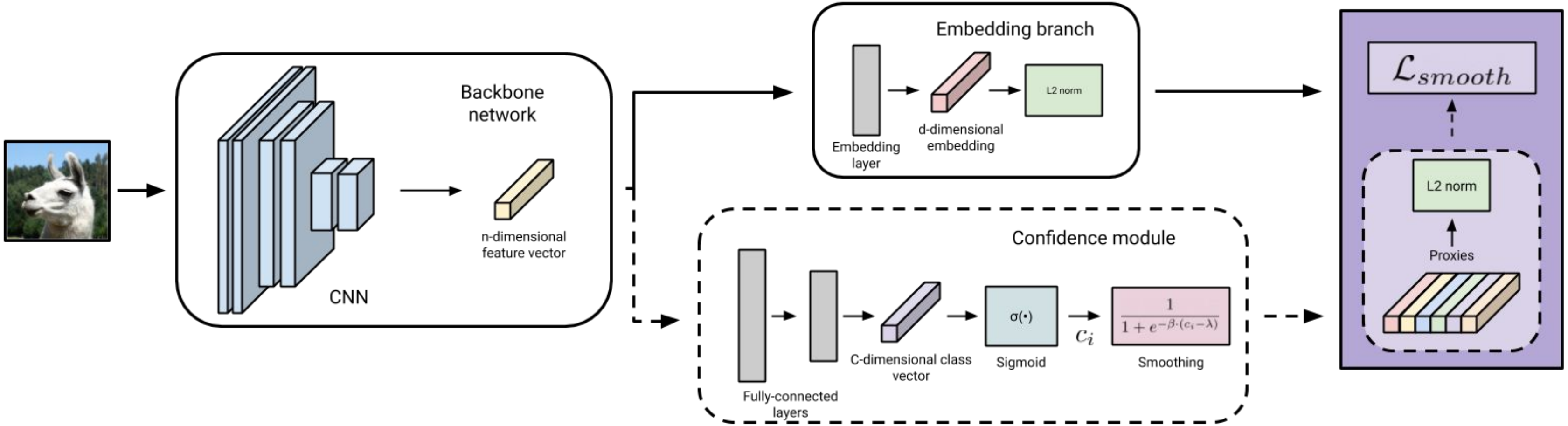
The positive samples X_p^+ corresponding to a proxy are selected if

$$c_{x,p} > \lambda$$

otherwise, they are negative.

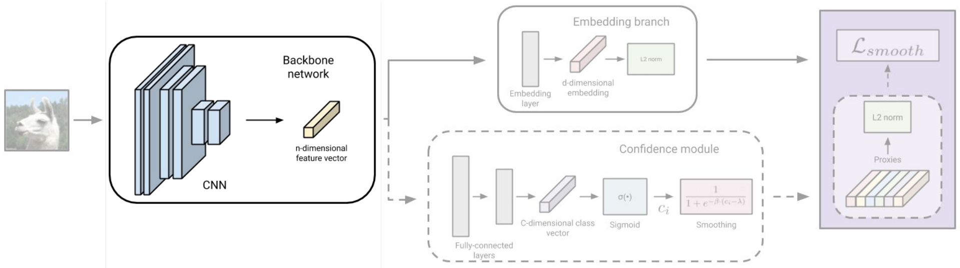
vilynx.

Our Method



Backbone network
CNN
n-dimensional feature vector
Embedding branch
Embedding layer
d-dimensional embedding
L2 norm
Confidence module
Fully-connected layers
C-dimensional class vector
σ(•)
Sigmoid
c_i
$\frac{1}{1+e^{-\beta\cdot(c_i-\lambda)}}$
Smoothing
$\mathcal{L}_{smooth}$
L2 norm
Proxies

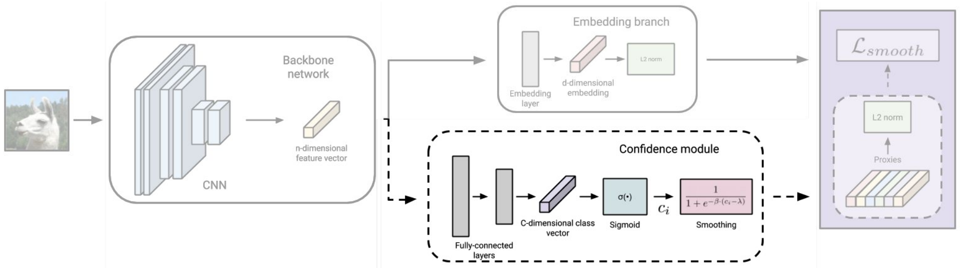
Our Method



- Backbone: **ResNet50.**
- Pretrained on the **ImageNet** dataset.
- Without the classification layer.
- Frozen for all experiments.

vilynx.

Our Method



Trained with Binary Cross Entropy loss:

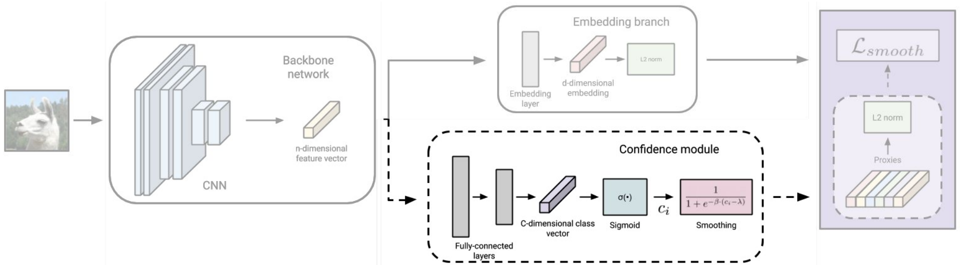
$$L(y,\hat{y}) = -\frac{1}{N}\sum_{i=0}^{N}(y * log(\hat{y}_i) + (1-y) * log(1-\hat{y}_i))$$

The confidence module generates the class confidences and the smoothing function balances each contribution.

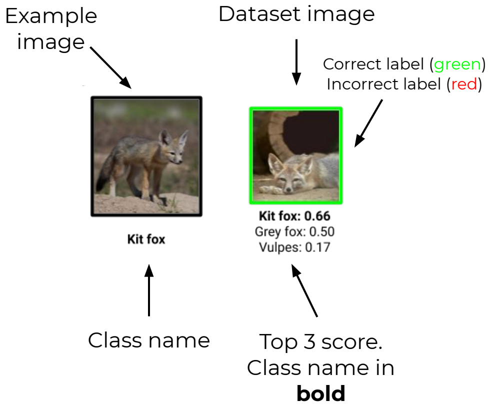
Dataset is a partition of the Webvision dataset*

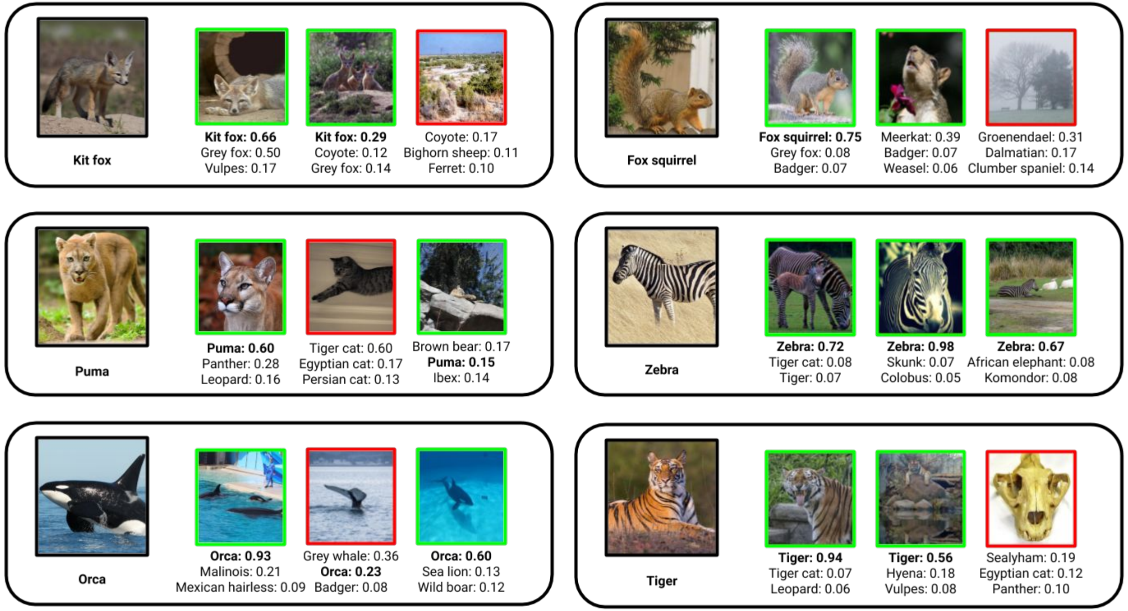
* More details in the paper

Our Method



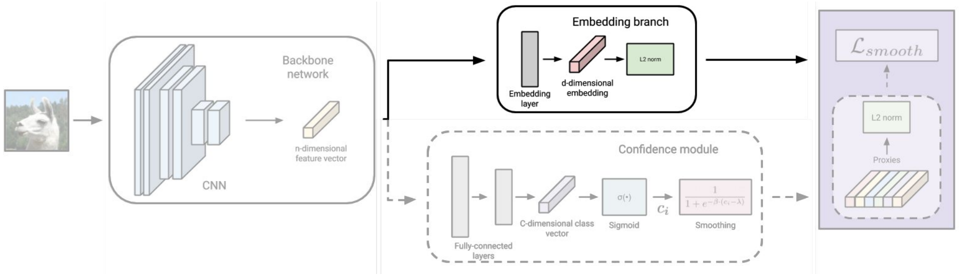
Top 3 confidence scores

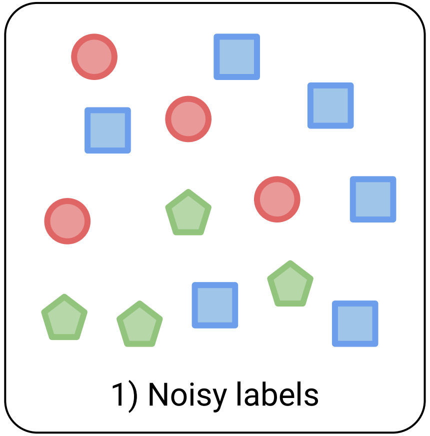




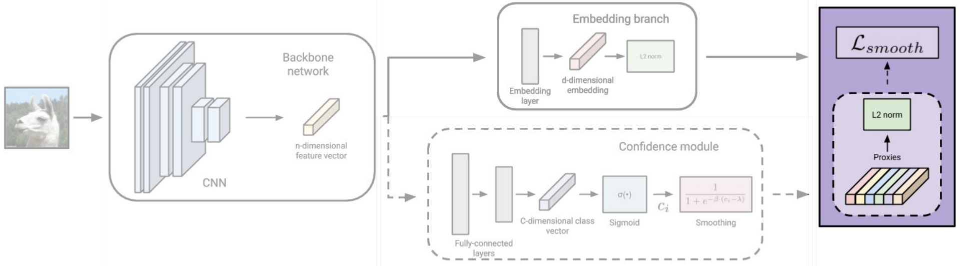
vilynx.

Our Method

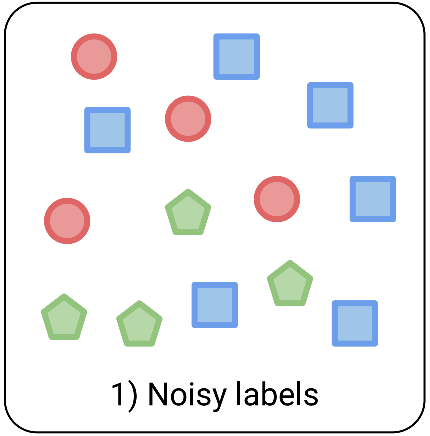

Backbone network
CNN
n-dimensional feature vector
Embedding branch
Embedding layer
d-dimensional embedding
L2 norm
Confidence module
Fully-connected layers
C-dimensional class vector
Sigmoid
Smoothing
Proxies
L2 norm


1) Noisy labels

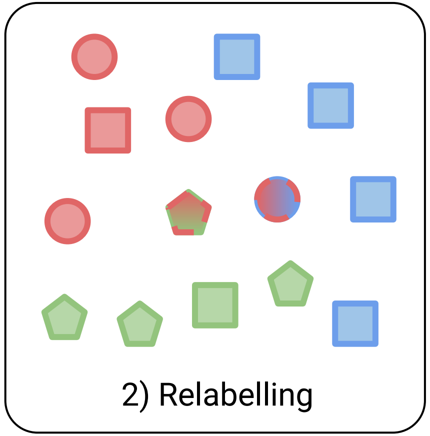
Our Method



Backbone network
CNN
n-dimensional feature vector
Embedding branch
Embedding layer
d-dimensional embedding
L2 norm
Confidence module
Fully-connected layers
C-dimensional class vector
Sigmoid
Smoothing
$\mathcal{L}_{smooth}$
L2 norm
Proxies

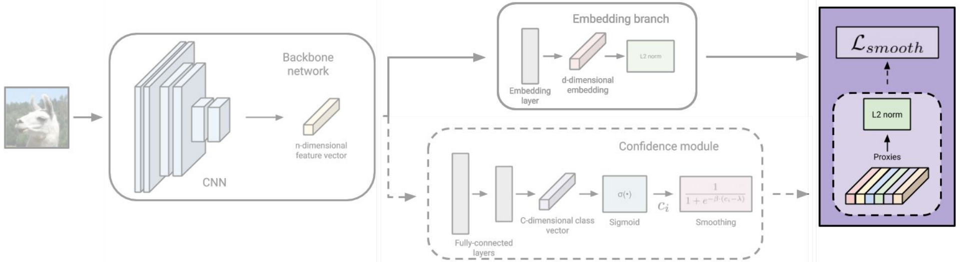


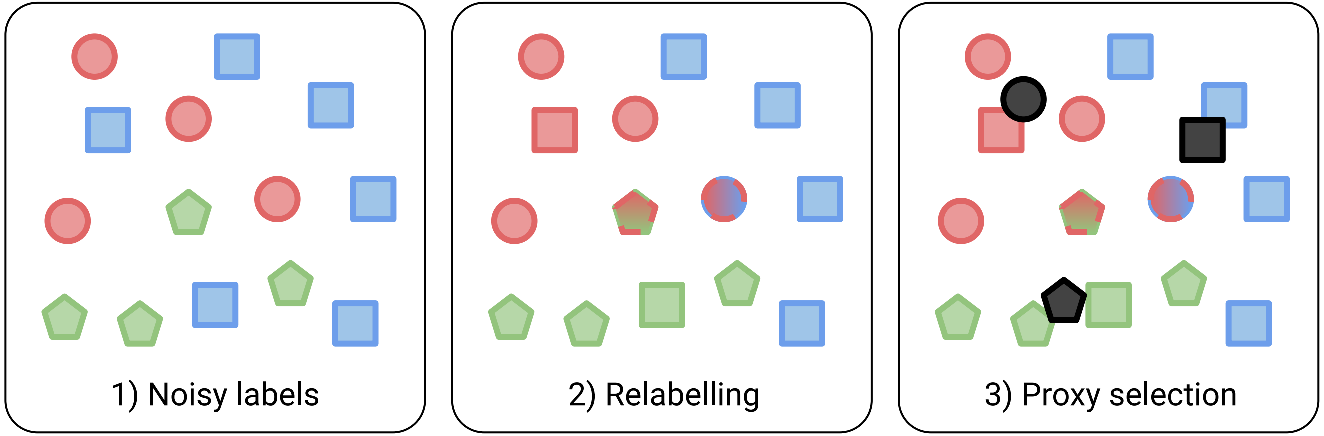
1) Noisy labels



2) Relabelling

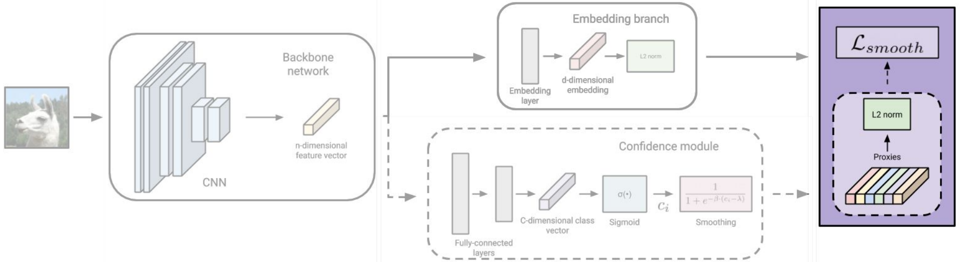
Our Method

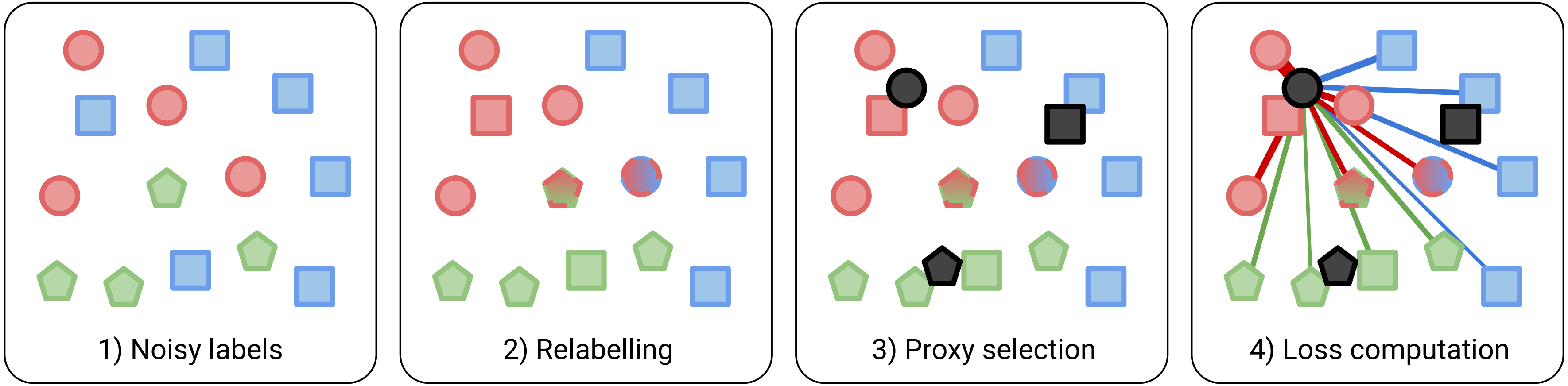




vilynx.

Our Method





vilynx.

Results

Recall@K	1	2	4	8	16
Proxy-NCA [1]	65.89	75.70	82.36	87.51	91.56
Proxy-Anchor [2]	67.95	77.47	84.50	89.33	93.09
MultiSimilarity [3]	68.61	70.08	85.04	89.95	93.42
Ours	**71.24**	**79.83**	**86.10**	**90.30**	**93.66**

Table 3. Comparison of Recall@K for different methods against our proposed loss on the WebVision dataset partition.

[1] Yair Movshovitz-Attias, Alexander Toshev, Thomas K. Leung, Sergey Ioffe, and Saurabh Singh. *No fuss distance metric learning using proxies*, 2017
[2] Sungyeon Kim, Dongwon Kim, Minsu Cho, and Suha Kwak. *Proxy anchor loss for deep metric learning*, 2020
[3] Xun Wang, Xintong Han, Weilin Huang, Dengke Dong, and Matthew R. Scott. *Multi-similarity loss with general pair weighting for deep metric learning*, 2019

Conclusions

- Two branch system for noisy metric learning
  - Confidence module
  - Embedding
- We propose a Smooth Proxy Anchor Loss that weights the contribution of noisy samples
- Our method improves 2.63 and 3.29 in Recall@1 with respect to MultiSimilarity and Proxy-Anchor loss respectively

vilynx.

Thanks!

carlos@vilynx.com

Get the paper!