Navee Project

Fashion products identification

Vassili Chesterkine
Haoyi Han
Carlos Santos García
Nicolas Vitor Yuki Obara Yamakoshi
Ke Zhang

Pôle Data Science - CentraleSupélec

Table of contents

Introduction

- **Few-shot learning (FSL)** : multiple applications e.g. character recognition, face detection for identification, etc.
  - Train models capable of classify objects with little data
- **Objective** : evaluate the possibility of using *pre-trained feature-extractors* to *categorize products* based on images with limited amounts of data
- **Procedure**
  1. Explored classical deep learning techniques to the *Kaggle Fashion Dataset*
  2. Few-shot learning techniques with few images per class

Problem statement - Few-shot classification task

Dataset :

$$D = (D_{support}, D_{query})$$

$$D_{support} = ((x_i, y_i)_{i \in I})$$

with $\forall i \in I$, x_i an image and y_i a label.

For every given value y of labels $\#\{i \in I \mid y_i = y\}$ is small, generally about 5.

N**-way** K**-shot classification problem** :

- N : number of classes
- K : number of samples per class
- **Objective** : find $\hat{h}_{N,K} : x_i \mapsto y_i$, based only on $D_{support}$.
- **Evaluation** : accuracy on D_{query}

Problem statement - Image retrieval task

- **Idea** : learn a representation of the images to cluster similar objects
- Given a *query image* and a *support set* of images from previously-unseen classes, retrieve all images of the support set that have the same label as the *query image*
- Train on D_{base} to learn the representation of images and use learned representation for FSL (similar to kNN)

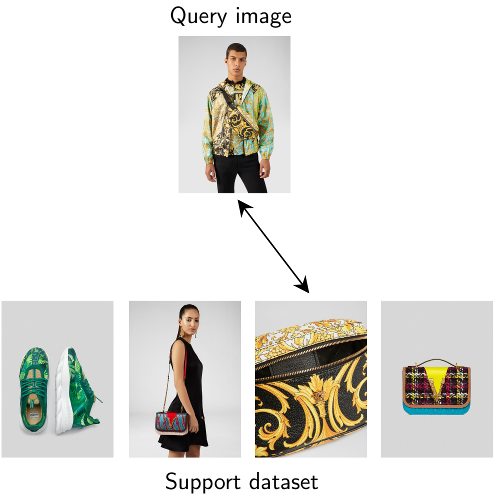


Figure – Image retrieval task

Problem statement - Image retrieval task evaluation

Retrieval task : Evaluation by **mean average precision (mAP)**

- **Average precision (AP)** : Compute for every sample (x_q, y_q) in D_{query} the *average precision* $AP(x_q)$ *of a binary classifier* that predicts whether or not each image x_s from the support dataset $D_{support}$ belongs to the same category y_q as x_q or not.

$$AP(x_q) = \sum_n (R_n - R_{n-1}) P_n$$

  where P_n and R_n are the precision and recall values at the n-th threshold.

- **Mean average precision** :

$$mAP = \frac{1}{\#D_{query}} \sum_{x_q \in D_{query}} AP(x_q)$$

Conventional computer vision task - Feature extraction

Feature extraction :

- Extract meaningful features from images for the process of classification
- 2 main approaches :
  - *Convolutional Neural Networks* : pre-trained models - *VGG16*, *ResNet-18* and *ResNet-50*
  - *Histograms of Oriented Gradients* : encode presence of edges and their direction by extracting location and orientation of gradients on the image

Figure – Oriented gradients obtained with a sock image from Kaggle dataset.

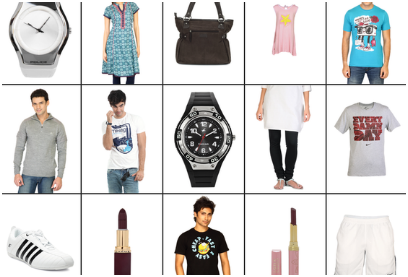
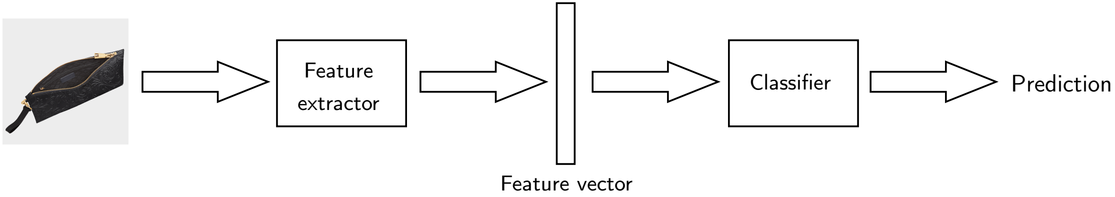


Figure – Kaggle Fashion Dataset

Conventional computer vision task - Multi-class classification

Multi-class classification :

- **On CNNs** : done automatically with fully-connected layers on top of the convolutional layers
- **On HOG extractors** : feature vectors pass through *k-nearest neighbors classifier* or *multi-layer perceptron*
- We also used ensemble learning techniques with multiple classifiers and finetuned pre-trained ResNet-50s



Representation learning - Introduction

Representation learning :

- Usage of "real" imperfect datasets of various product images from 7 brands, provided by Navee
- 3967 classes across 7 brands
- Learn a representation of images in an embedding space
- The chosen strategy should favor the creation of clusters of images from the same category

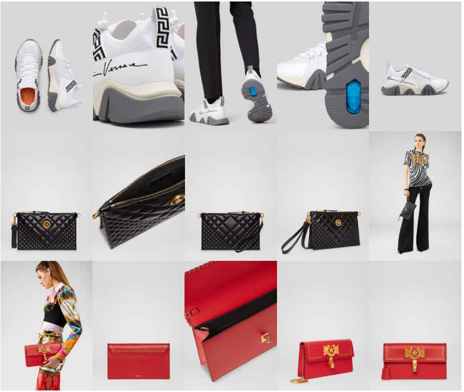
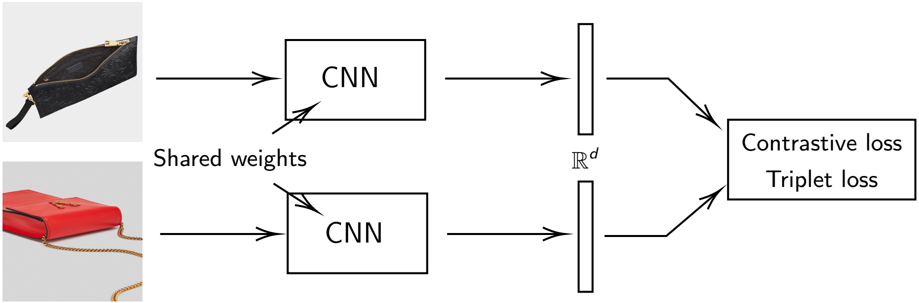


Figure – Examples of images in three Versace categories

Representation learning - Siamese Networks

Siamese Networks :

- Neural networks that contain two or more identical sub-networks, that share same characteristics and parameters
- Each sub-network is used to compute an embedding for an input, then all embeddings are compared with a loss function
- In sum, they learn a similarity function between inputs, which can be used to classify images with classes that were not in the training set



Representation learning - Loss functions

Contrastive loss :

- Uses pairs of inputs, positive (same class : $y = 1$) or negative (different classes : $y = 0$)

Contrastive loss expression

$$L = y \times d(x, x') + (1 - y) \times \max(0, m - d(x, x'))$$

with (x, x') an input pair of representations, d the Euclidean distance and m the margin.

Triplet loss :

- Uses triplets of inputs, an anchor sample x^a, a positive sample x^p (same class as the anchor) and a negative sample x^n (different class)

Triplet loss expression

$$L(x^a, x^p, x^n) = \left[d(x^a, x^p) + m - d(x^a, x^n)\right]_+$$

Idea : push similar images close together and dissimilar images far from another in the embedding space

Representation learning - Triplet selection

Triplet selection :

- Only a selection of useful triplets is used for computing gradients, not all of them (too computationally intensive)
- Selecting semi-hard or hard triplets has a huge impact on performance
- **Implementation** : Given a mini-batch, we would compute all losses and select the top k % of greatest losses to compute gradients on, where k is a hyper-parameter that we selected
- Used in lieu of proper online selection because of computational power constraints

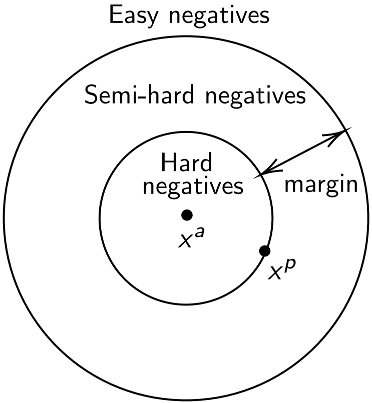


Figure – Types of triplets

Representation learning - Classification using embeddings

Classification using embeddings : using the outputs of the embedding functions, we can use these representations to classify unseen images, using 2 different methods :

- **k-Nearest Neighbors (kNN)** : Use the embeddings of all images to identify the class of a given query image, based on kNN strategy
- **Perceptron** : Use an perceptron to classify images based on the embeddings, with only 1 hidden layer

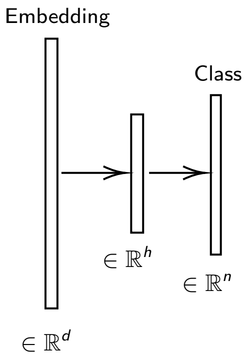


Figure – Structure of the MLP : d is the dimension of the embeddings, h is the dimension of the hidden layer and n is the number of classes of the query set

Other techniques - Model-Agnostic Meta-Learning

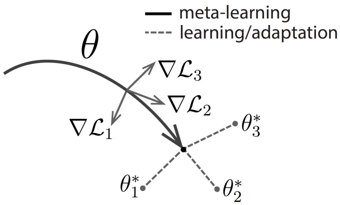


Figure – Illustration of MAML training.

Model-Agnostic Meta-Learning :

- Algorithm to train a model on multiple learning tasks, for it to solve new learning tasks using only few training samples and few gradient descent steps
- **Model-agnostic** : any model that uses *gradient descent* can be utilized
- **Training dataset** : composed of *Tasks*, where each *Task* $\mathcal{T}_i$ is a N-way K-shot classification problem on its own, with a query image and a support set
- **Training** :
  1. For each *Task* $\mathcal{T}_i$, copy current parameter vector θ and do gradient descent to achieve θ_i'
  2. Update θ using gradient descent to minimize sum of losses $\mathcal{L}_{\mathcal{T}_i}(f_{\theta_i'})$

Other techniques - Prototypical Networks

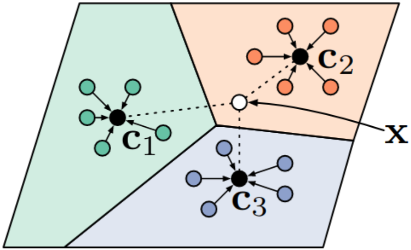


Figure – Illustration of prototypical networks in the embedding space.

Prototypical Networks :

- Learns to attribute a "prototypical" central point to each class in the embedding space

$$c_k = \frac{1}{|S_k|} \sum_{(x_i,y_i)\in S_k} f_\phi(x_i),$$

with S_k as the subset of support set for k-th class and f_ϕ as the encoder function.

- Classifies the query image by choosing the class whose prototype has minimal distance to the query's embedding

Preliminary results

Preliminary results :

- Pre-trained embeddings and HOG extractors had excellent performance on Kaggle data (especially HOG + MLP)
- Ensemble of 3 ResNet-50s also improved performance

	masterCategory	subCategory	articleType
ResNet-50	98%	94%	82%
HOG + kNN	98%	91%	80%

Table – Test accuracies obtained with the two main methods for Kaggle dataset images.

ŷ: Sandals
y: Sports Sandals

ŷ: Tops
y: Shirts

ŷ: Mobile Pouch
y: Handbags

ŷ: Sport Shoes
y: Casual Shoes

ŷ: Ring
y: Bangle

ŷ: Jeans
y: Jeggings

ŷ: Flat
y: Heels

Experiments and results - Methodology

Methodology :

- Training on all brands images besides 50 articles from Versace and all images from Givenchy
- Checking retrieval performances on Versace support dataset (N=50)
- Checking retrieval performances on Givenchy support dataset (N=309)
- Data augmentation was applied

Experiments and results - Hyperparameter search

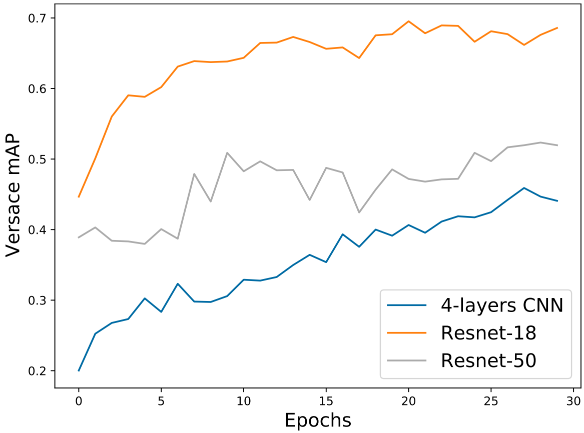


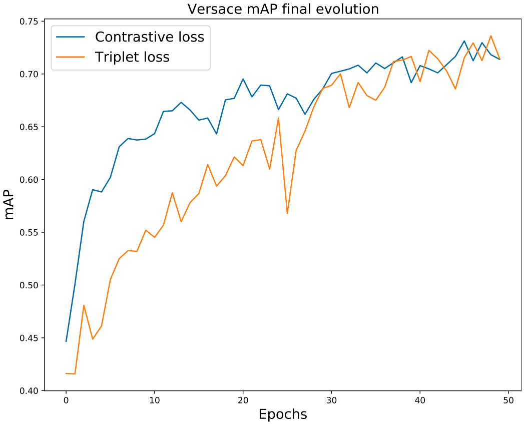
Figure – mAP on Versace suggests the relevance of using middle-sized models - an example of the explorations made.

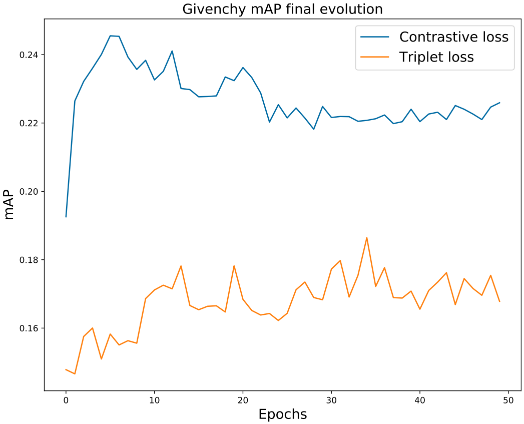
Hyperparameter search :

- After testing, an optimal value was found for learning rate λ, p_{mining} and d on Siamese networks
- p_{same} and m were also explored, but weren't influential in results
- For CNNs, finetuning the whole pre-trained ResNet yielded the best result

Retrieval task final results :

Results on Versace ($N = 50$) and Givenchy ($N = 309$) support datasets. Only first 50 epochs are plotted, but contrastive loss was trained for another 40 epochs.





Experiments and results - Retrieval task final results

Prototypical network's results :

mAP (%)	**20-way 1-shot**	**60-way 1-shot**	**60-way 5-shot**
VERSACE	26.46	24.97	28.29
Burberry	10.81	10.59	10.60
D&G	15.83	16.30	*None*
YSL	29.83	29.14	*None*

Table – mAP obtained using Prototypical Networks for VERSACE, Burberry, D&G and YSL on different FSL parameters

Prototypical Network's final results :

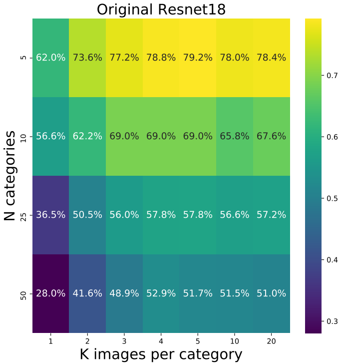
- Evaluated independently and respectively on four different brands, using a different methodology from other FSL models
- No apparent differences when changing the number of ways or shots
- Fail to obtain good enough results

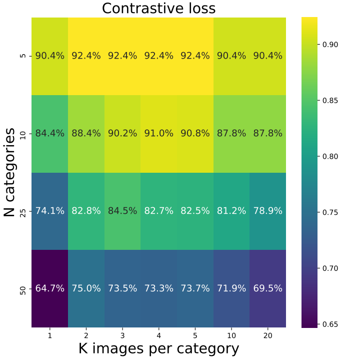
Factors that may influence the training :

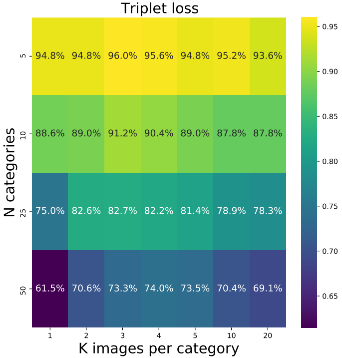
- Accessories & Ornaments
- Colour
- Encoder function

Experiments and results - Few-shot classification task 1/2

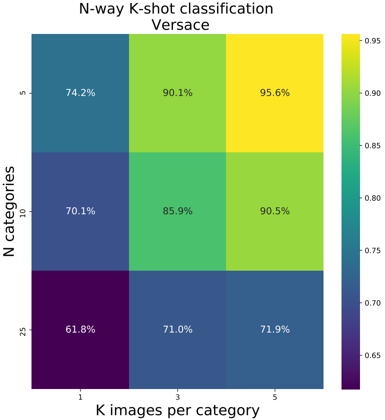
Few-shot classification task : in average, our networks have boosted by 23% the classification performance of our network on Versace's unseen data







Experiments and results - Few-shot classification task 2/2



MAML's final results :

- Evaluated on Versace, using the same methodology as other FSL model's
- Overall better performance than classical non-FSL methods, like pre-trained ResNet-18
- But fails to surpass other models

Conclusion

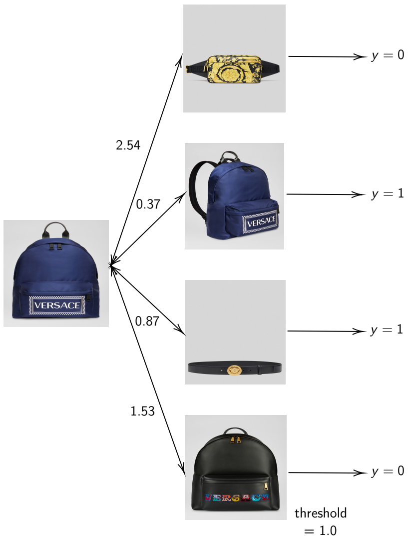
Conclusion :

- Explored computer vision techniques
- Proposed the use of siamese networks with margin losses to perform few-shot classification
- Obtained superior retrieval performance to existing deep learning off-the-shelf methods.
- Explored popular few-shot learning techniques

Further ideas :

- Self-supervised learning techniques could provide more data to build proper embeddings from

mAP Pipeline



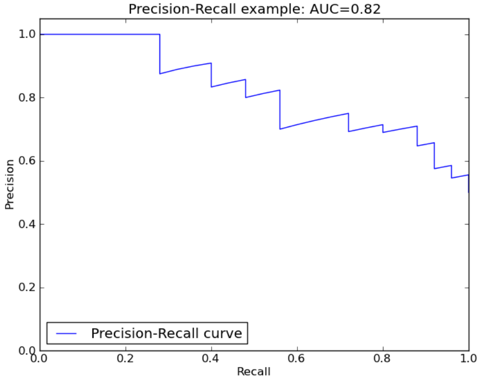
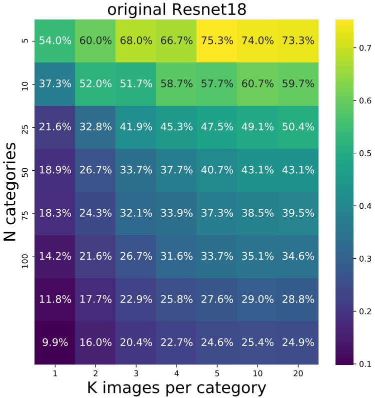
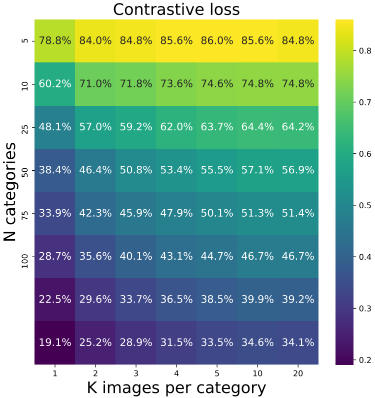
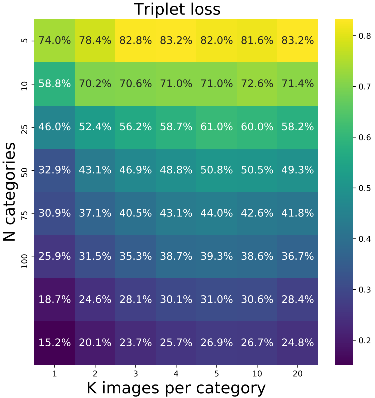


Figure – Precision-recall curve

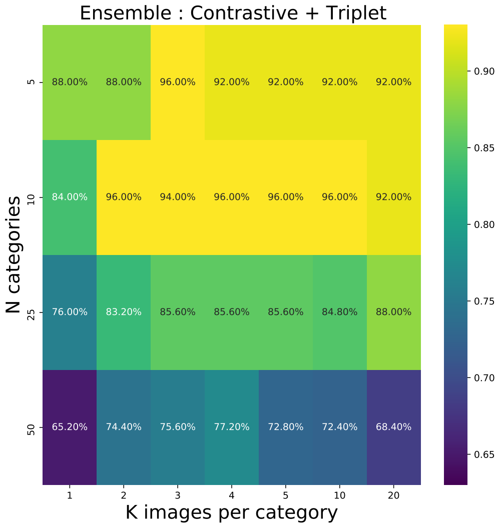
Givenchy few-shot classification

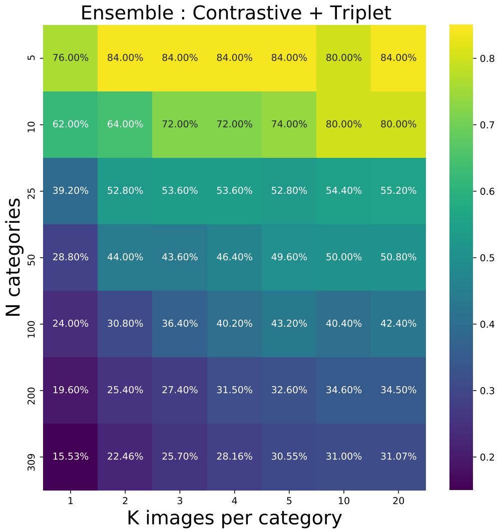






Ensemble


Ensemble : Contrastive + Triplet
N categories
K images per category
5
88.00% 88.00% 96.00% 92.00% 92.00% 92.00% 92.00%
10
84.00% 96.00% 94.00% 96.00% 96.00% 96.00% 92.00%
25
76.00% 83.20% 85.60% 85.60% 85.60% 84.80% 88.00%
50
65.20% 74.40% 75.60% 77.20% 72.80% 72.40% 68.40%
1 2 3 4 5 10 20
0.90
0.85
0.80
0.75
0.70
0.65


Ensemble : Contrastive + Triplet
N categories
K images per category
5
76.00% 84.00% 84.00% 84.00% 84.00% 80.00% 84.00%
10
62.00% 64.00% 72.00% 72.00% 74.00% 80.00% 80.00%
25
39.20% 52.80% 53.60% 53.60% 52.80% 54.40% 55.20%
50
28.80% 44.00% 43.60% 46.40% 49.60% 50.00% 50.80%
100
24.00% 30.80% 36.40% 40.20% 43.20% 40.40% 42.40%
200
19.60% 25.40% 27.40% 31.50% 32.60% 34.60% 34.50%
309
15.53% 22.46% 25.70% 28.16% 30.55% 31.00% 31.07%
1 2 3 4 5 10 20
0.8
0.7
0.6
0.5
0.4
0.3
0.2