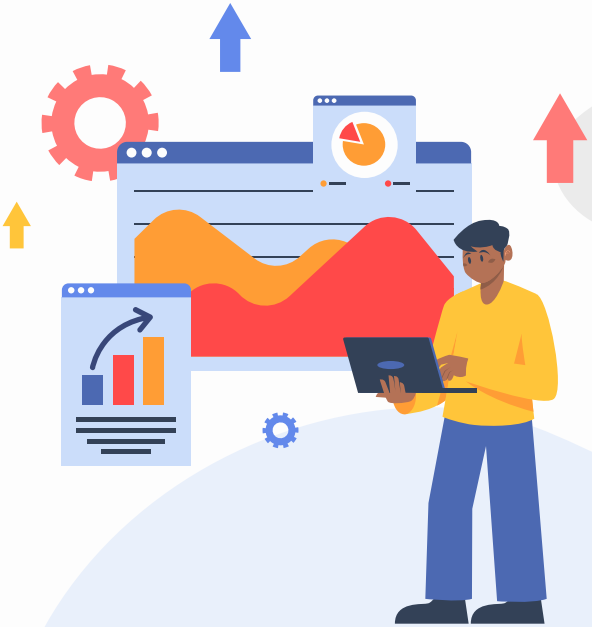
Soutenance de projet
30 janvier 2025

CentraleSupélec

HeadMind Partners

Challenge IA

Etude de cas HeadMind Partners - Dior



BOUAITA Rayane – DAVID Erwan – EL ANATI Pierre – FAYNOT Guillaume – TRIER Gabriel

01
Problématique

Contexte

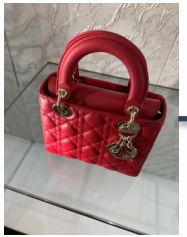
Images de test

BDD d’images

Catégories produits	%
Bags	30
RTW	23
Accessories	20
Shoes	12
SLG	11
Watches	0,04

02
Pipeline

Pipeline



DAM
Img_test
Préprocessing
Calcul d'embeddings
Métrique
Matching

03
Préprocessing

Préprocessing

Images de test

BDD d’images

Problèmes rencontrés

- Taille (HD)
- Dimensions (carré vs rectangle)
- Arrière-plan

Solutions apportées

- Suppression de l’arrière-plan (rembg)
- Crop autour de la forme isolée (carré)
- Diminution de la résolution (256x 256)

RemBg

- Module python **open source**
- Suppression d'arrière plan via **modèles profonds**

Fonctionnement de rembg

- Conversion en .png
- Normalisation par canal (RGB)
- Segmentation avec U²-Net (classification foreground vs background)
- Application du masque binaire au canal alpha

Image initiale

.remove()

Image modifiée

Annexe U²-Net

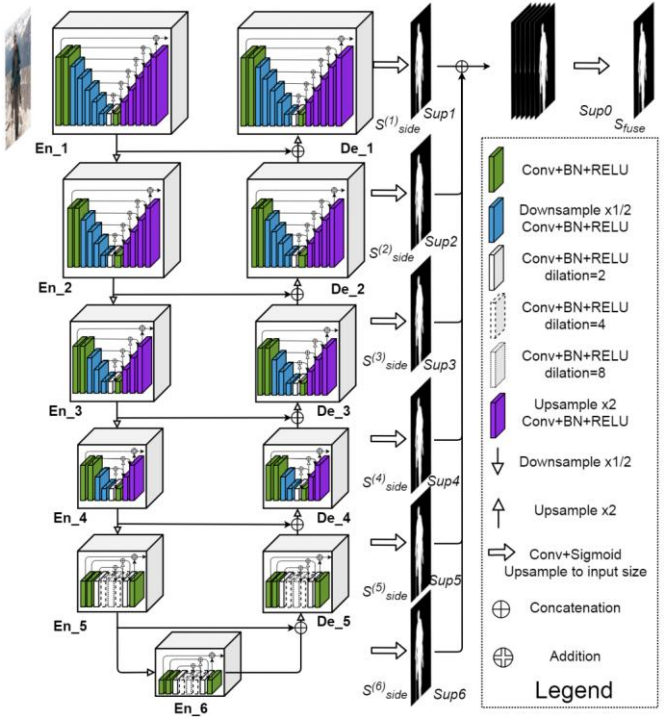


Figure 5. Illustration of our proposed U²-Net architecture. The main architecture is a U-Net like Encoder-Decoder, where each stage consists of our newly proposed residual U-block (RSU). For example, **En_1** is based on our RSU block shown in Fig. 2(e). Detailed configuration of RSU block of each stage is given in the last two rows of Table 1.

04
Data augmentation

Data augmentation

Chaussure gauche ou droite ?

- Flip horizontal

Photo mal orientée

- Rotation +/- 45°

Objet mal éclairé

- Ajustement des couleurs
- Ajustement de la luminosité

Objet mal cadré

- Redimensionnement

Bilan - Pipeline

DAM → Rembg → Crop & fit → Data

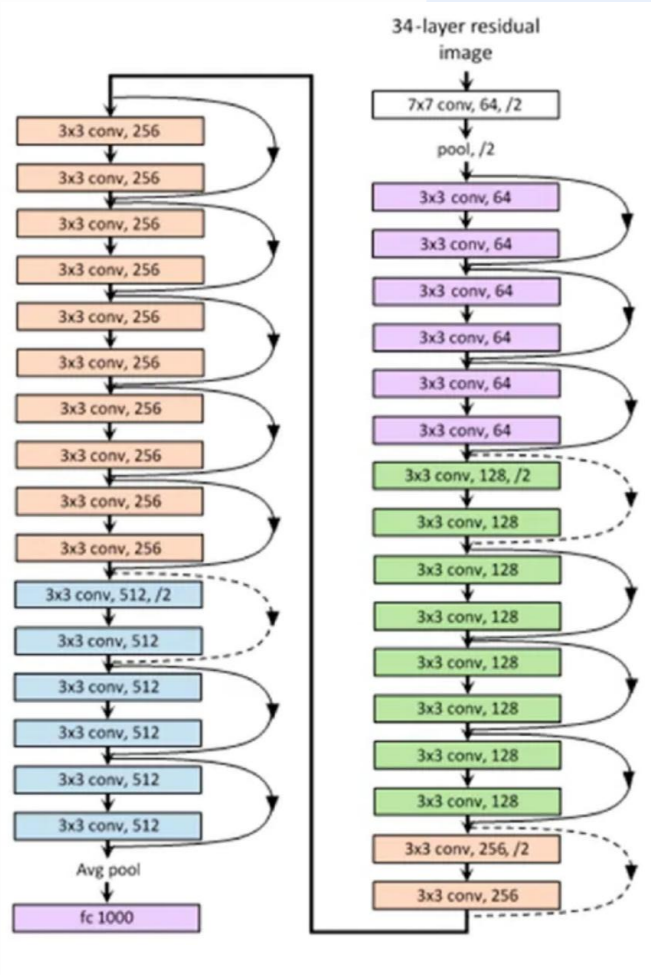
Img_test → Rembg → Crop & fit → Data

Calcul d'embeddings → Métrique → Matching

05
Modèles

ResNet

- Architecture introduite en 2015 (*Microsoft Research*)
- 1ère place ILSVRC → **3,57%**
- Skip connections
- Entrainé sur ImageNet
- Utilisé comme extracteur de caractéristiques

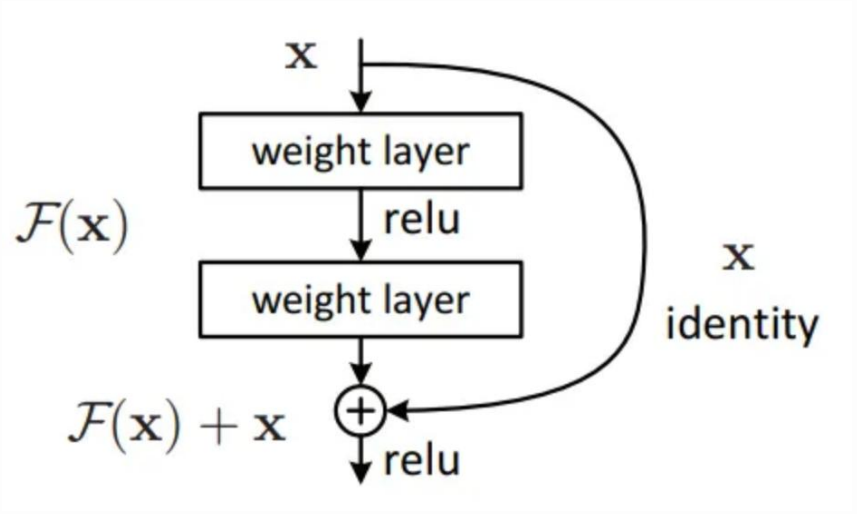


Skip Connections

Problème : Vanishing Gradient

Solution : Skip Connections

- Apprentissage de **fonction résiduelle**
- **Résolution** des **limites de profondeur**



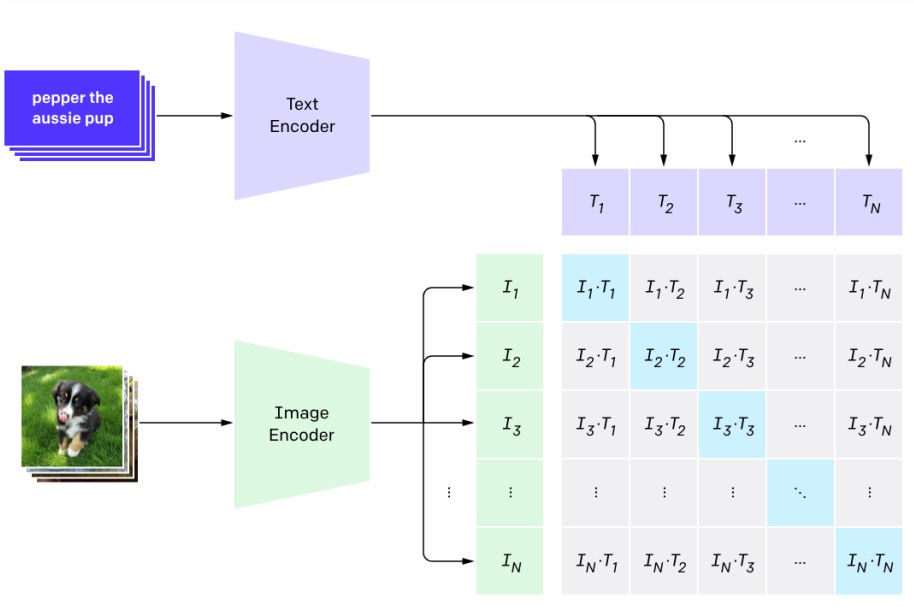
ResNet50

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	7×7, 64, stride 2				
conv2_x	56×56	3×3 max pool, stride 2				
		$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 64 \\ 3\times3, 64 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 64 \\ 3\times3, 64 \\ 1\times1, 256 \end{bmatrix}\times3$
conv3_x	28×28	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 128 \\ 3\times3, 128 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times4$	$\begin{bmatrix} 1\times1, 128 \\ 3\times3, 128 \\ 1\times1, 512 \end{bmatrix}\times8$
conv4_x	14×14	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 256 \\ 3\times3, 256 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times6$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times23$	$\begin{bmatrix} 1\times1, 256 \\ 3\times3, 256 \\ 1\times1, 1024 \end{bmatrix}\times36$
conv5_x	7×7	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times2$	$\begin{bmatrix} 3\times3, 512 \\ 3\times3, 512 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$	$\begin{bmatrix} 1\times1, 512 \\ 3\times3, 512 \\ 1\times1, 2048 \end{bmatrix}\times3$
	1×1	average pool, 1000-d fc, softmax				
FLOPs		1.8×10^9	3.6×10^9	3.8×10^9	7.6×10^9	11.3×10^9

Clip

Formation auto-supervisée :

- Deux réseaux de neurones distincts : ViT et transfomer
- Générer deux vecteurs de même dimension dans un espace commun
- Rapprocher les paires correspondantes et à éloigner les paires incorrectes
- La perte contrastive InfoNCE



Clip

Motivations

Meilleure correspondance sémantique des images

- CLIP encode des concepts visuels de manière plus riche qu'un ResNet, qui lui se limite aux caractéristiques discriminatives utiles pour la classification.

Robustesse aux Variations Visuelles

- Entraîné sur des données variées avec un alignement image-texte, ce qui lui permet de mieux capturer les similitudes visuelles sous différents angles.

Zero-shot

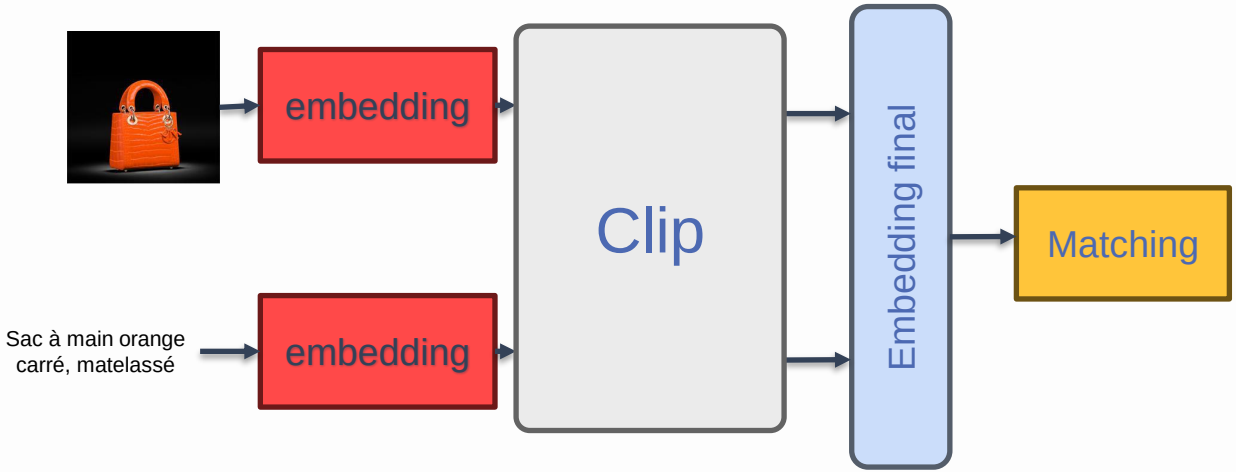
- CLIP peut fonctionner directement sans réentraînement.

Clip

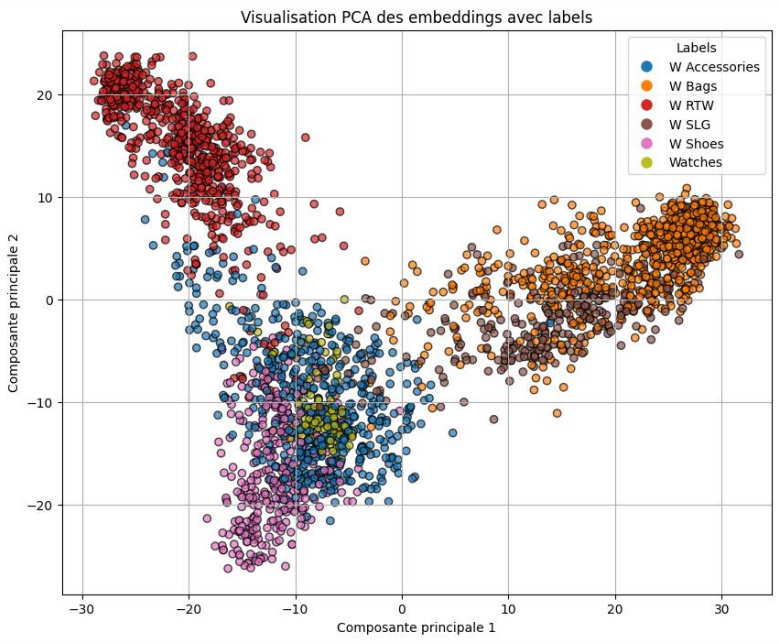
Embedding des images

ResNet50	Clip
Non	Non
Couleur	Couleur
Cosine	Cosine
41,25%	31,25%
55%	51.25%
61%	61.25%

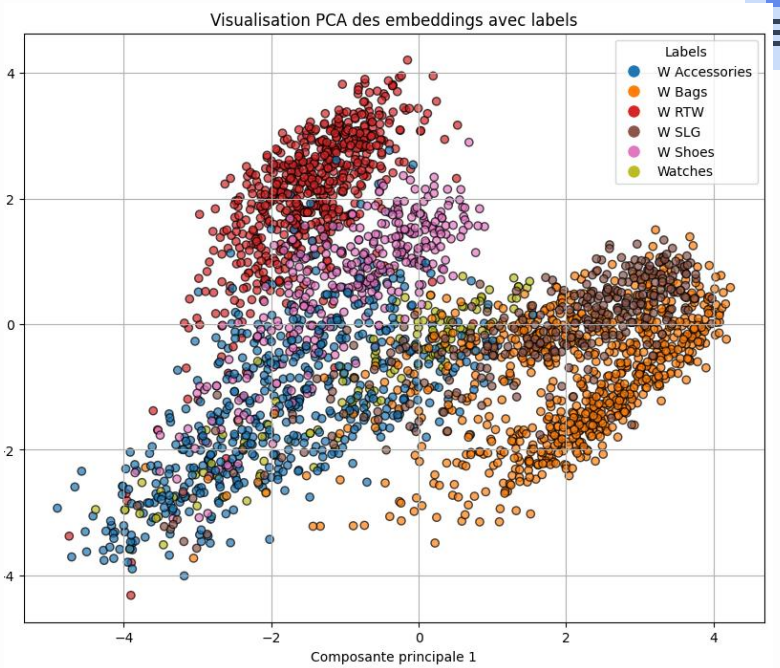
Embedding des images et du texte



Clip



ResNet50



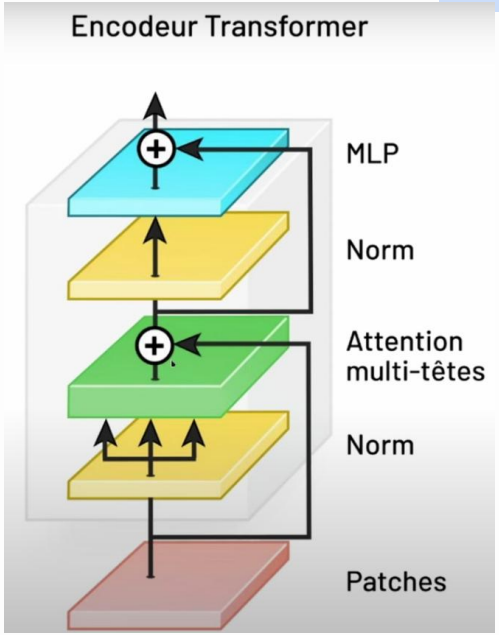
CLIP

DINOv2

Pré-traitement de l'image
Redimensionnement, conversion en tenseur, normalisation

Traitement
Découpage en patchs, encodage des patchs (partie locale de l'image),
Ajout d'un token CLS (résumé global de l'image)
Passage des patchs dans un transformer -> multi heads attention

Extraction du token
Représentation riche et compressée de l'image



DINOv2

Motivations

Par rapport à ResNet

- Plus robuste aux variations intra-classe (ex: variations d'éclairage, d'angles de prise de vue ou de bruit dans les images)
- Contrairement à ResNet (réseau convolutif classique), DINOv2 utilise un **Vision Transformer (ViT)**

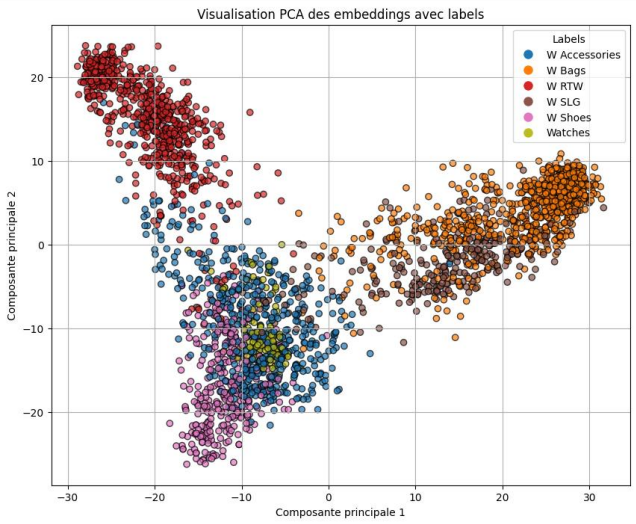
Par rapport à Clip

- Encoder d'image plus finement optimisé

DINOv2

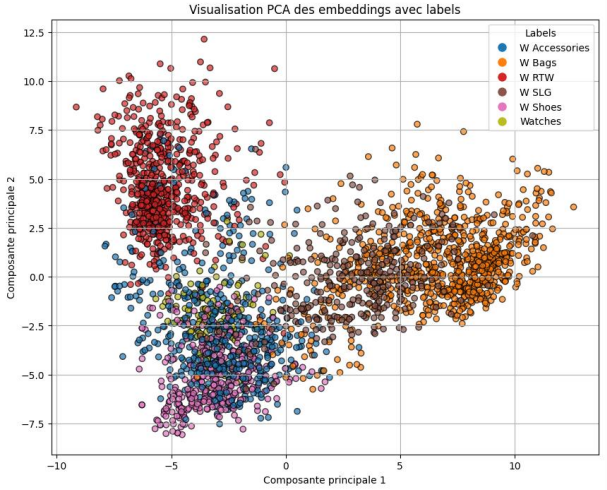
Résultats

DinoV2	ResNet50
Non	Non
FlipV+Couleur	FlipH +Couleur
Cosine	Cosine
42,50%	45%
55%	56,25%
60%	65%



ResNet50

DinoV2

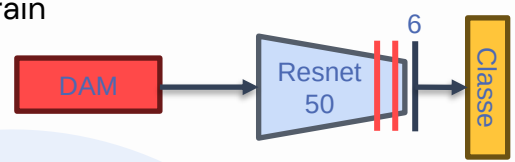


06
Fine-Tuning

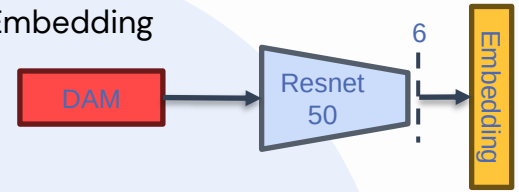
Fine-Tuning

Spécialisation sur le Dataset

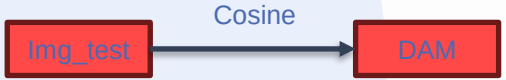
1. Train



2. Embedding

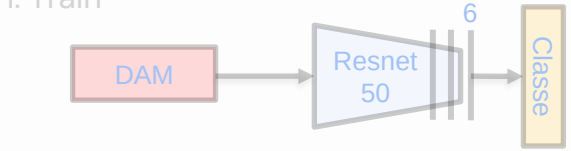


3. Retrieval

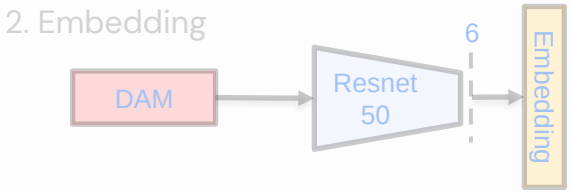


Récupération par classe

1. Train



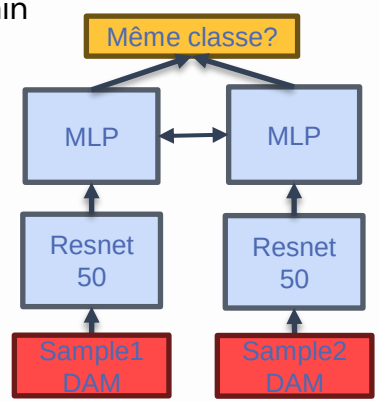
2. Embedding



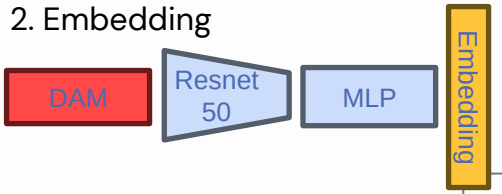
3. Retrieval

Rapprochement inter-classes

1. Train



2. Embedding



07
Métriques

Métriques

Comment retrouver une image de train à partir d'une image de test ?

=

Quelle métrique pour trouver un minimum de distance entre les embeddings ?

Similarité cosinus

$$D(A,B) = \frac{A \cdot B}{\|A\|\|B\|}$$

L1

$$D(A,B) = \sum_i |A_i - B_i|$$

L2

$$D(A,B) = \sqrt{\sum_i (A_i - B_i)^2}$$

Résultats

Embeddings	ResNet50	ResNet50	ResNet50	ResNet50	ResNet50	ResNet50	ResNet50	DinoV2	Clip	ResNet50	ResNet50
Fine tuning	Non	Non	Non	Non	Non	Non	Non	Non	Non	Oui	Oui+classe
Data augmentation	Non	Non	Non	Flip H	Couleur	FlipH +Couleur	FlipV+Couleur	FlipV+Couleur	Couleur	Non	Non
Méthode	Cosine	L1	L2	Cosine	Cosine	Cosine	Cosine	Cosine	Cosine	Cosine	Cosine
Accuracy Top 1	45%	45%	45%	45%	41,25%	45%	37,50	42,50%	31,25%	28%	46%
Accuracy Top 3	58,75%	57,5%	56,25%	60%	55%	56,25%	54%	55%	51.25%	46%	59%
Accuracy Top 5	67,5%	63,75%	66,25%	68,75%	61%	65%	59%	60%	61.25%	53%	67%

Constatations

- ResNet50 > DinoV2 > Clip
- FlipH > No Data Augmentation
- Cosine > L1 & Cosine > L2

07
Conclusion

Merci