

Projet_08: Réalisez un dashboard et assurez une veille technique

Jérôme LE GAL Etudiant OpenClassRooms – parcours Data Scientist Le 24/01/2025

Contexte et objectifs :

- Mission pour « Prêt à dépenser »
- Visualisation interactive pour les « chargé de relation client »
- Veille technique

- Dashboard sur le cloud
- Note méthodologique (technique de modélisation récente)

Elaborer un Dashboard

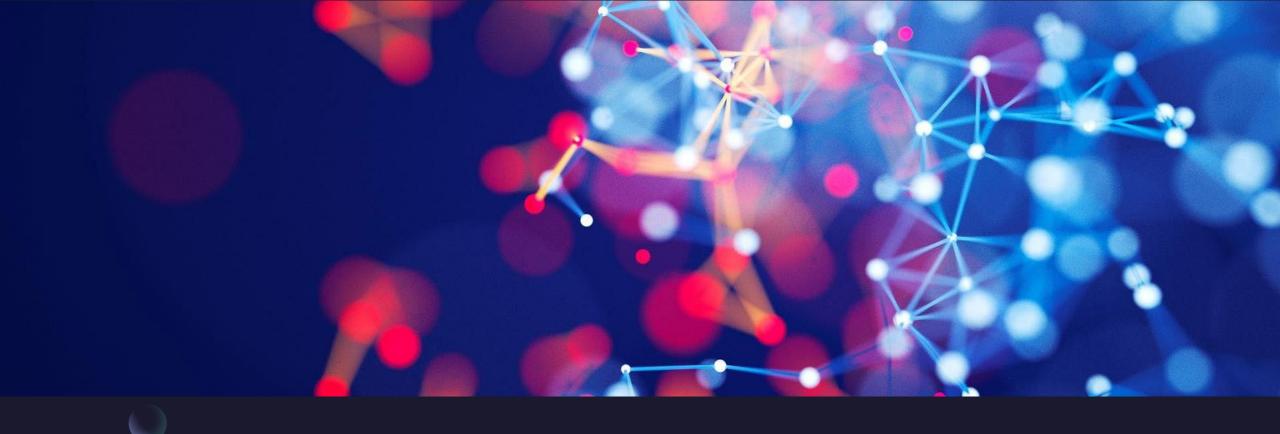
- I. Collecte d'informations
- 2. Comprendre le besoin des utilisateurs
 - 3. Comment répondre au besoin
 - 4. Blueprint
 - 5. Réalisation du Dashboard
 - 6. Déploiement / tests
 - 7. Présentation



Les besoins:

- ✓ Informations client
- ✓ Informations du prêt
- ✓ Score et probabilité
- ✓ Possibilité de comparaison
- ✓ Déploiement Cloud
- ✓ Critères d'accessibilité du WCAG





Démonstration:

http://dashboard-container.germanywestcentral.azurecontainer.io:8501/



Projet d'amélioration

Projet_06 : Classification d'images



- Dataset: 1050 images
- 7 catégories

Etat de l'art

- ProtoViT (10/2024):
 - Prototype-based
 - Transformers ViT



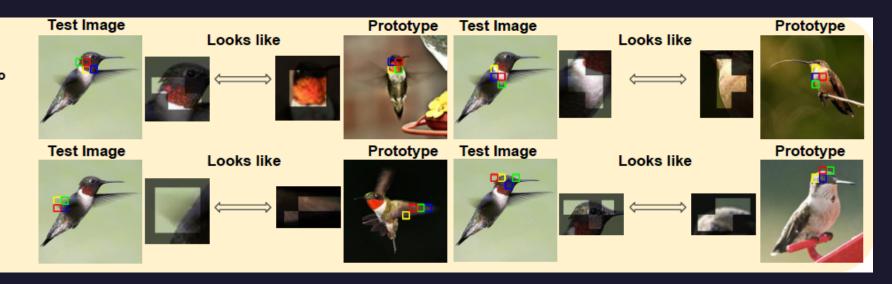
https://arxiv.org/abs/2410.20722





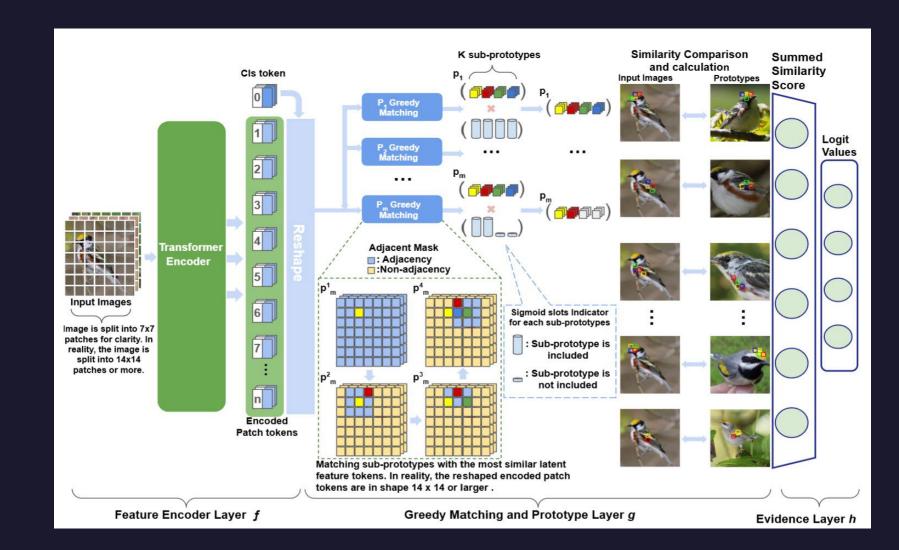
By ProtoViT (Ours):

Deformed prototypes adapt to shape and have semantic coherence. Each feature is represented by 4 boxes.



Architecture

- Feature Encoder Layer
- Greedy Matching
- Evidence Layer



Entrainement du modèle

- Initialisation des prototypes
- Former les couches d'encodage des features
- Elagage des slots
- Projection des prototypes
- Finetuning

Initialized ProtoViT

Stage 1: Optimizing layers before the last layer

Phase 1 Warm-up training: train feature encoder with an extremely small learning rate Phase 2 Joint Optimization: train feature encoder with a larger learning rate

Objective function:

$$\mathcal{L}_{total} = \mathcal{L}_{CE} + \lambda_1 \mathcal{L}_{Clst} + \lambda_2 \mathcal{L}_{Sep} + \lambda_3 \mathcal{L}_{Coh} + \lambda_4 \mathcal{L}_{Orth}$$

Stage 2: Slots Pruning

Objective function:

$$\mathcal{L}_{prune} = \mathcal{L}_{CE} + \lambda_5 \mathcal{L}_{Coh}$$

Stage 3: Prototype Projection

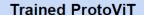
Objective: 1. Round the fractional approximation of slots indicator values to the closest integer (1/0)

2. Project prototypes to the K nearest latent feature tokens

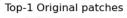
Stage 4: Last Layer Fine-Tuning

Objective function:

$$\mathcal{L}_h = \mathcal{L}_{CE} + \lambda_6 \mathcal{L}_{Sparsity}$$



Interprétabilité





Top-1 Activated prototype



Top-2 Original patches



Top-2 Activated prototype



Top-3 Original patches



Top-3 Activated prototype



Top-4 Original patches



Top-4 Activated prototype



Top-5 Original patches



Top-5 Activated prototype



Similarity X Last Layer = Total weight



$$3,94 \times 0,868 = 3,419$$

$$3,94 \times 0,867 = 3,415$$

$$3,92 \times 0,872 = 3,418$$



Top-1 Original patches Top-1 Activated prototype

Top-2 Original patches





Top-3 Original patches





Top-4 Original patches



Top-4 Activated prototype



Top-5 Original patches



Top-5 Activated prototype



Similarity X Last Layer = Total weight

 $0.65 \times 0.947 = 0.615$



 $0,44 \times 0,945 = 0,415$

 $0,44 \times 0,943 = 0,414$

 $0,43 \times 0,94 = 0,404$

Logit total: 46,378

Classe: 3

Logit total: 2,608 Classe: 6

Comparatif des résultats

Modèle	Accuracy	Macro F1-score	Classes améliorées
VGG16	83.33%	0.83	Watches, Kitchen&Dining
ProtoViT	88.10%	0.88	Baby Care, Beauty&Personal

- Accuracy: +4,77% avec ProtoViT
- F1-score: +5% avec ProtoViT
- Meilleure gestion des confusions



Conclusion:

- Performances globales surpassées
- ➤ Robustesse pour les classes plus complexes
- > Généralisation plus robuste

