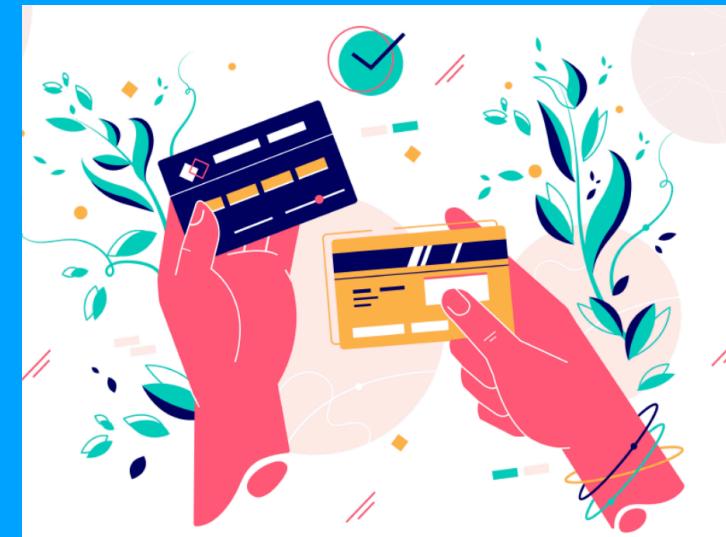


Dashboard et veille technique

Prêt à dépenser

EC - novembre 2023



Contexte et problématique

Entreprise

« Prêt à dépenser » est une société financière qui propose des **crédits à la consommation** à des personnes ayant peu ou pas d'historique de prêt.

Besoin 1

- La société souhaite développer un dashboard interactif pour que les chargés de relation client puissent **expliquer de façon transparente les décisions d'octroi de crédit**.

Mission 1

- Développer un **dashboard interactif** sur la base de l'API de prédiction réalisée lors du projet précédent.

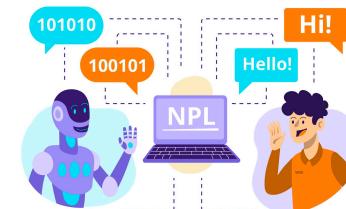


Besoin 2

- La société est soucieuse de mettre en oeuvre les **dernières techniques** en data science sur les thématiques de données texte et image.

Mission 2

- Réaliser un **état de l'art** sur une technique récente de modélisation de données texte ou de données image.
- **Analyser, tester et comparer** cette technique à une approche plus classique.



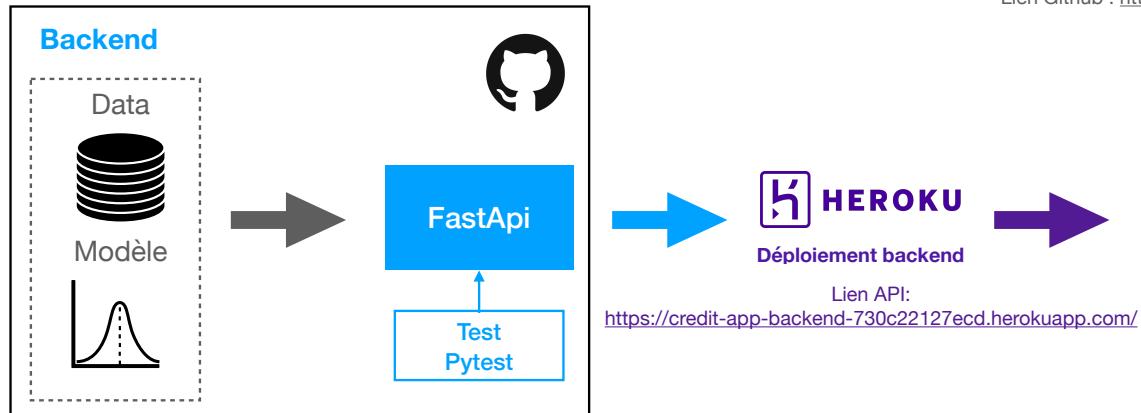
Dashboard

Déploiement du dashboard

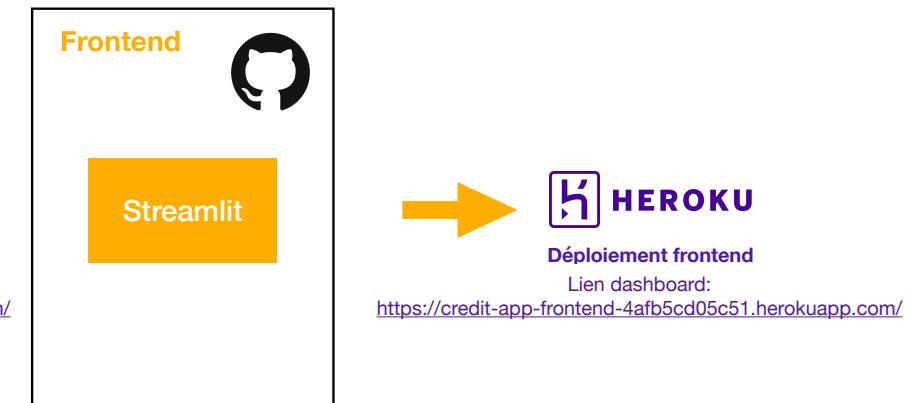
La robustesse et l'efficacité de **FastAPI** pour le traitement backend, la gestion des données et de la logique business.

La simplicité et l'interactivité de **Streamlit** pour l'interface utilisateurs et la visualisation des données.

Lien Github : https://github.com/estellec18/app_credit_scoring



Lien Github : https://github.com/estellec18/dashboard_credit_scoring

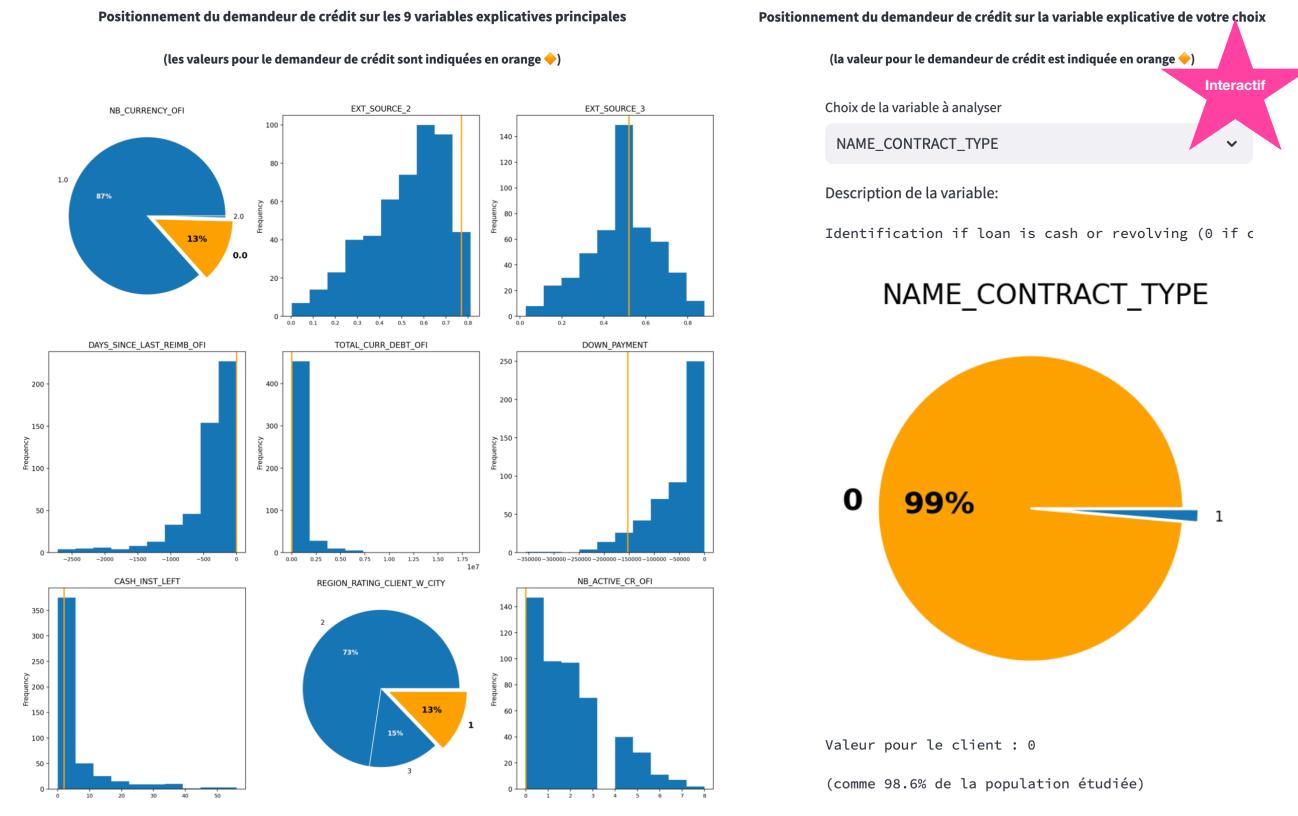


Les graphiques du dashboard

Graphique relatif au modèle



Graphiques relatifs au résultat obtenu pour un client spécifique



Etat de l'art : le Vision Transformer

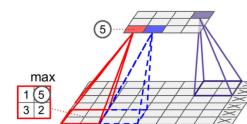
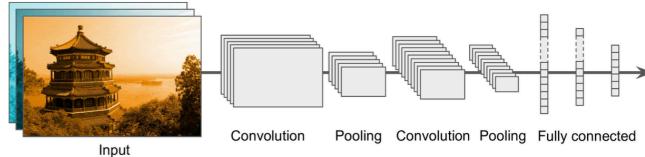
Computer vision

Vision Transformer (ViT)

Contexte

Computer vision

- ◆ Le **Convolutional Neural Networks** (CNN) est un modèle de deep learning dont l'architecture est issue de l'étude du cortex visuel humain. Il est utilisé pour la reconnaissance d'image depuis les années 80.



Le CNN est un standard pour les tâches de reconnaissance d'image ; de nombreux modèles ont vu le jour (VGG, Resnet, Efficientnet...).



NLP

- ◆ Le **Transformer** est une architecture de Neural Network très populaire en NLP, qui tire son succès de son **mécanisme d'attention***.

Avec le Transformer (i) l'ensemble des données est traité en une fois et (ii) le positionnement de chaque input retient son contexte dans la séquence grâce au mécanisme d'attention, ce qui permet au modèle de s'entraîner sur des datasets plus importants et de réduire le temps d'entraînement.

Le Transformer est devenu un standard pour les tâches de NLP : les algorithmes les plus populaires l'utilisent (BERT, GPT, BART...).



ViT

- ◆ En 2020 est introduit le **Vision Transformer** dans le rapport de recherche « An image is worth 16x16 words » (Dosovitskiy et al. 2020)

Principe : appliquer un Transformer directement sur des images.

Le modèle ViT est pré-entraîné sur les datasets ImageNet et ImageNet-21k et atteint des performances comparables, voire supérieures, à celles des CNN dans de nombreuses tâches de reconnaissance d'image.

Le ViT trouve aujourd'hui de nombreuses applications en computer vision : tâches « classiques » (détection d'objet, segmentation d'image, classification..), modèles génératifs, tâches multi-modales, traitement de vidéo...

* source image : « Hands-On Machine Learning with Scikit-Learn, Keras & Tensorflow », A. Géron, O'Reilly (2019)

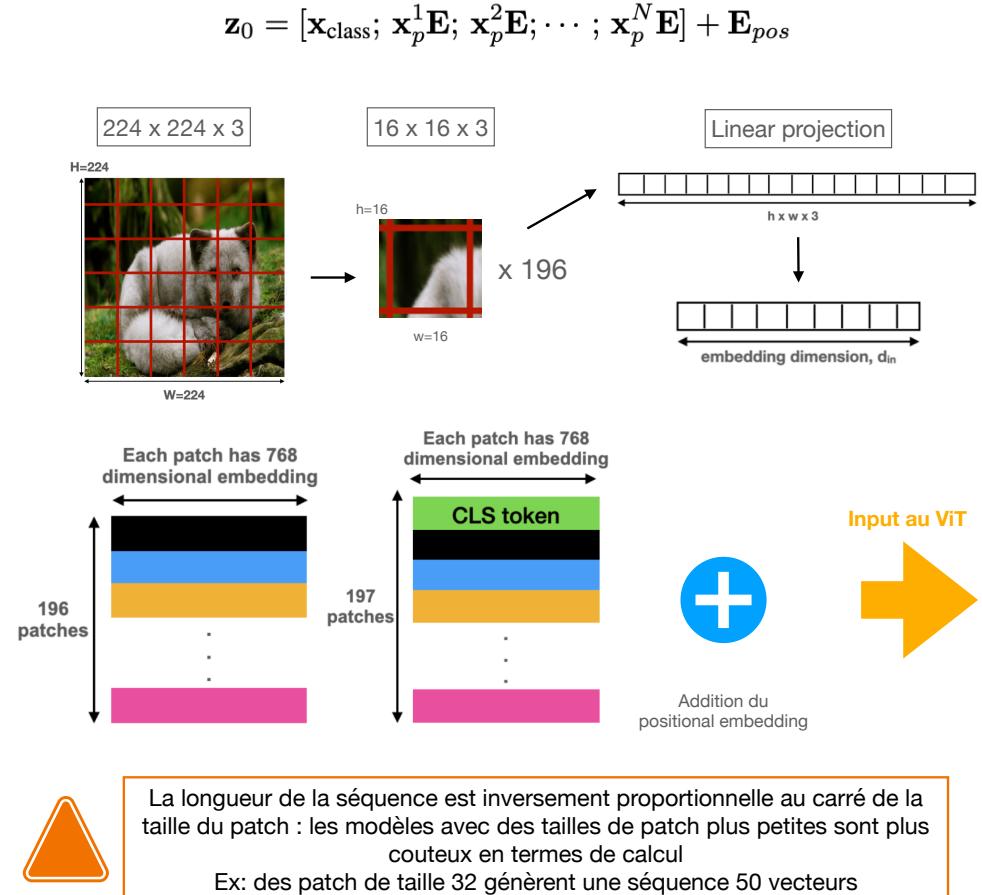
* introduction du mécanisme d'attention dans le rapport de recherche : « Attention is all you need » (Vaswani et al., 2017)

Vision Transformer (ViT)

Transformation de l'image en token

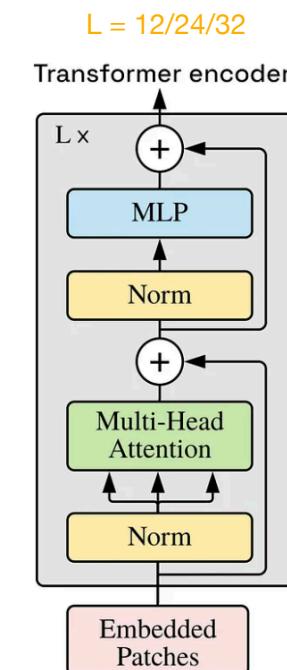
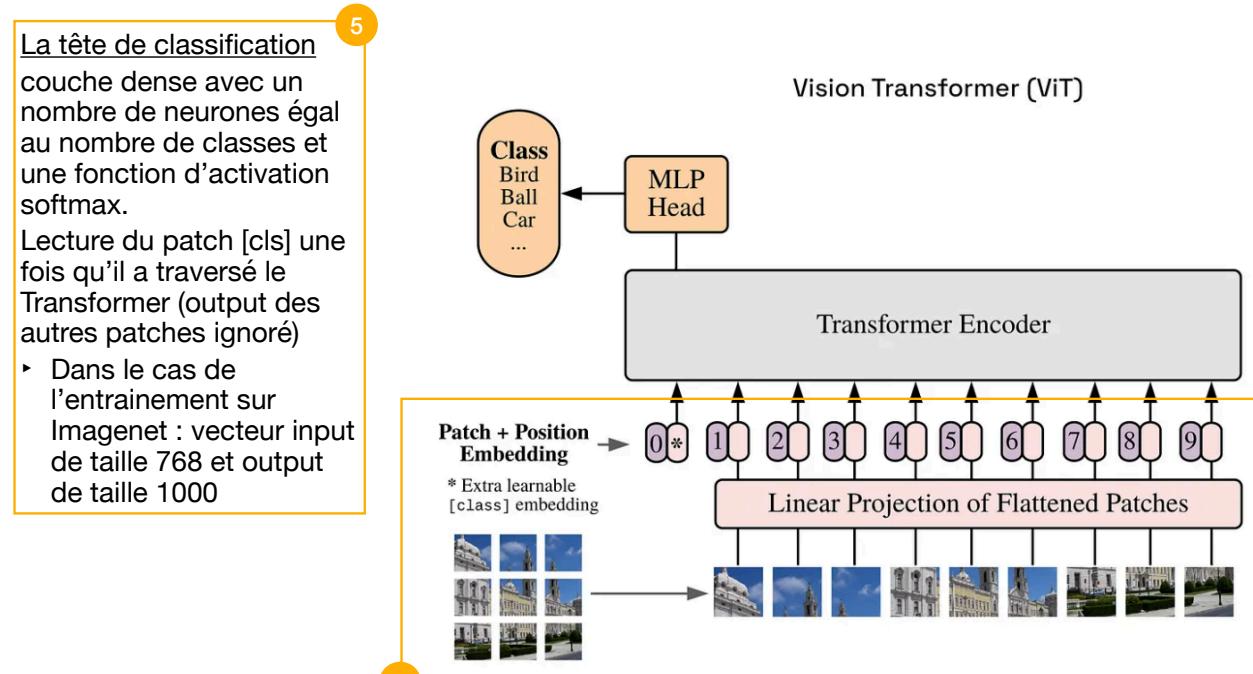
Etapes de l'embedding d'une image

- L'image est pré-traitée : soustraction de la moyenne et division par l'écart type (classique).
- L'image est divisé en « patches ».
 - Une image de taille 224 x 224 est divisée en 196 « patches » de taille 16 x 16
- Chaque patch est « **flattened** » via une projection linéaire et « **embedded** » via une matrice d'embedding initialisée au hasard.
 - Nous obtenons ainsi 196 vecteurs de taille 768 (dimension D propre au modèle)
- Ajout d'un **patch [cls]** à la séquence (« learnable parameter ») ; il sera utilisé pour lire les résultat de la classification à la fin du modèle.
 - Nous obtenons alors 197 vecteurs de taille 768
- Pour permettre au Transformer d'apprendre à différencier les patches selon leur localisation, ajout d'un « **positional embedding** » (« learnable parameter »).
 - Embedding positionnel à une dimension : l'input est une séquence de patches dans l'ordre matriciel.
 - Somme « element-wise » de la matrice de vecteurs et du paramètre de positional embedding.
 - Les 197 vecteurs de taille 768 sont alors passés au Transformer



Vision Transformer (ViT)

Architecture globale du modèle



- Multi-Layer Perceptron (MLP)**
- couche linéaire : projection de l'output du MHA dans une dimension plus élevée
 - couche d'activation GELU
 - couche dropout (contre l'overfitting)
 - couche linéaire : projection de l'output à la même taille que le MHA
 - couche dropout

- Mécanisme d'attention**

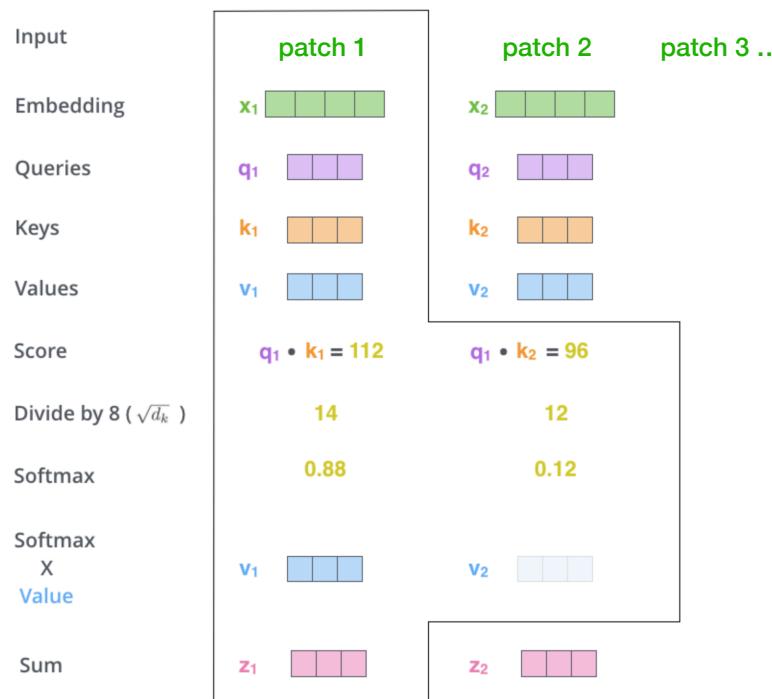
- Normalisation**
- Participe à la convergence rapide du gradient et à éviter le phénomène d'« exploding gradient »
 - Layer normalisation : calcul de la moyenne et de la variance pour chaque instance séparément, sur toutes les features
- * Normalisation indépendante de la taille de batch

* source image : « An image is worth 16x16 words » (Dosovitskiy et al. 2021)

Vision Transformer (ViT)

Focus sur le mécanisme d'attention

Détail du processus de self attention appliquée sur un patch



Ces calculs sont réalisés sous forme matricielle pour un traitement plus efficace. Les matrices **Query**, **Key** et **Value** sont obtenues en multipliant la matrice **d'embeddings** avec les matrices de poids entraînées (WQ, WK, WV).

$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V \quad \longleftrightarrow \quad \text{softmax}\left(\frac{Q \times K^T}{\sqrt{d_k}}\right)V = Z$$

Dans le mécanisme de « Multi-head attention » il y a N set de matrices de poids WQ/WK/WV (le Transformer de base en utilise 8, le ViT 12 ou 16 selon la variante).

Chacun de ces sets est initialisé au hasard. Après entraînement, ces sets permettent de projeter l'embedding input dans un subspace différent.

On se retrouve ainsi avec N matrices d'attention Z

Ces N matrices sont concaténées et multipliées par une matrice de poids W0 (entraînée conjointement avec le modèle).

On aboutit ainsi à **une unique matrice finale qui capture les informations des N matrices**.

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \dots, \text{head}_h)W^O$$

$$\text{where } \text{head}_i = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

Preuve de concept

Classification d'images

Data source

- 1050 articles de consommation courante classés dans 7 catégories distinctes.
- Les articles sont uniformément répartis entre les 7 catégories (150 articles par catégories).
- Format : .jpg



Objectif : classifier automatiquement les images dans leur catégorie respective à l'aide d'un algorithme de machine learning

- ➡ **Apprentissage supervisé** : les observations du training set sont labellisées.
- ➡ **Classification en classe multiples** : il s'agit de répartir les articles entre 7 catégories différentes.

Metrics

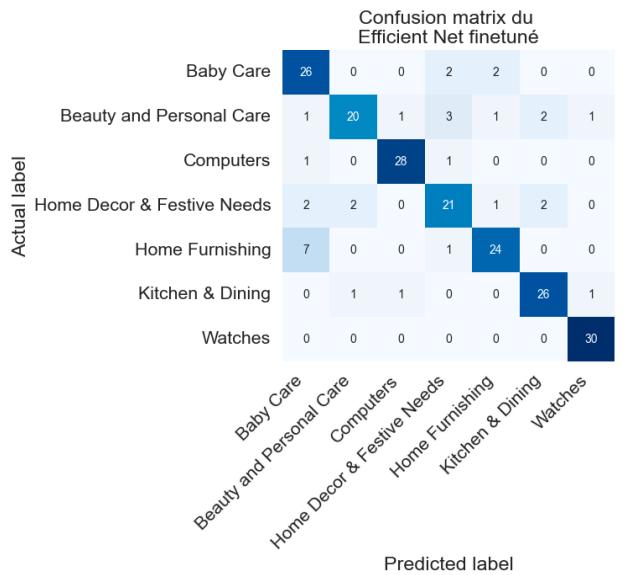
- ◆ **Accuracy** : le nombre de fois où le classifier prédit correctement,
- ◆ **AUC** : mesure de la capacité d'un classifier à faire la distinction entre les classes,
- ◆ **Matrice de confusion** : récapitulatif des erreurs,
- ◆ **Temps d'entraînement**,
- ◆ Différence entre les performances du training set et du validation set pour évaluer la **capacité de généralisation du modèle**,
- ◆ **Precision et recall** pour comparer les meilleurs modèles.

Comparaison des résultats

Convolutional Neural Network vs. Vision Transformer

	Efficient Net	ViT
Optimizer	RMSPROP	Rectified Adam
Learning	0.00316	0.0001
Loss	Kullback Leibler Divergence	Categorical Crossentropy avec label smoothing de 0.2
Batch size	32	32
Epochs	30	30

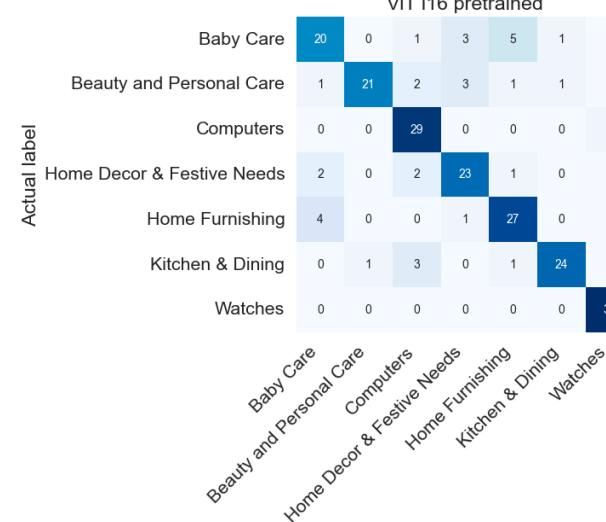
Hyperparamètres optimisés



Precision Recall

70%	87%
87%	69%
93%	93%
75%	75%
86%	75%
87%	89%
94%	100%

Confusion matrix du ViT I16 pretrained



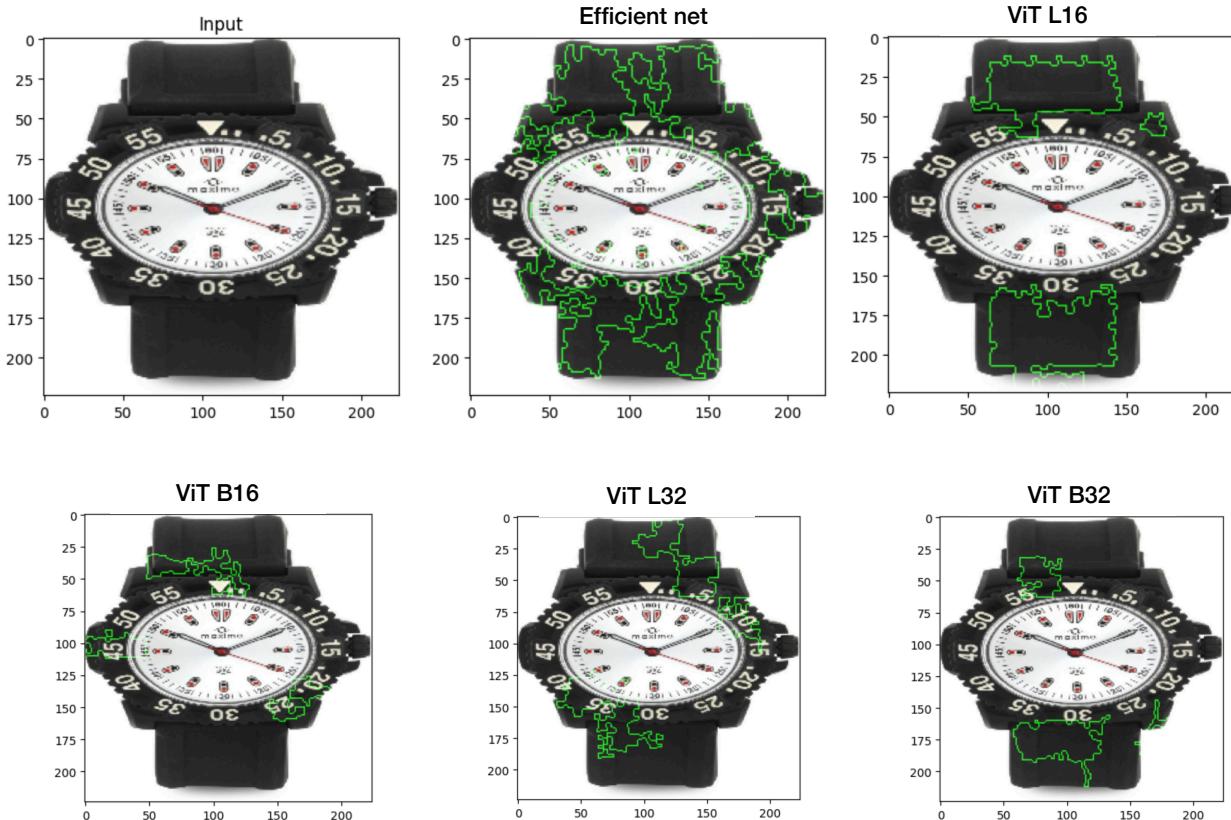
Precision Recall

74%	67%
95%	72%
78%	97%
77%	82%
77%	84%
92%	83%
97%	100%

Capacité à généraliser:
On note un écart plus important entre les performances du validation set et du training set pour le ViT que pour l'Efficient Net, caractéristique d'une capacité à généraliser plus limitée (phénomène d'overfitting)
(Cf. Détail en Annexes)

Comparaison des résultats

Saliency map



Limites du modèle ViT

1. Le Transformer est dépourvu de biais inductif

- Deux problématiques inhérentes à son architecture
 - Un process de tokenisation limitant :
 - une image est divisée en patches de taille égale qui ne se superposent pas. Les tokens ont ainsi un champ de vision limité et la relation avec les pixels adjacents n'est pas suffisamment prise en compte.
 - Un mécanisme d'attention limité :
 - du fait du grand nombre de tokens, la distribution des scores d'attention est lisse. Le modèle ViT ne parvient pas à se concentrer sur les tokens « importants ».
- Le modèle a ainsi tendance à avoir une attention redondante qui n'arrive pas à se concentrer sur une target.
 - Le ViT se concentre sur le background et non pas sur la forme caractéristique de l'image en lien avec la target.
 - Le ViT n'est pas sensible aux « high frequency ».

2. Le Transformer est gourmand en ressource

- Du fait de son mécanisme de self attention bi-directionnelle, dans lequel tous les tokens sont comparés par paire, un Transformer standard a une complexité quadratique par rapport à la longueur du token d'entrée (temps et mémoire).
- L'apprentissage des propriétés inductives par un Transformer nécessite **une quantité substantielle de données d'entraînement** alors que ces biais sont mis en oeuvre « by design » dans les CNN.
- Le modèle ViT a obtenu de bons résultats sur les tâches de classification et a même surpassé des architectures SOTA (ResNet) en termes d'accuracy et de cout pre-training. Mais ces résultats sont obtenus après entraînement du ViT sur des datasets énormes.
 - ➔ Sur des plus petits datasets, les CNN ont de meilleurs résultats.

A noter

- ❖ Le manque de biais inductif aboutit à **une architecture plus générale**. Les Transformers ont une vision globale de l'image dès les premières couches du modèle, alors que le CNN obtient ces informations seulement dans les dernières couches (car partant de champs de vision plus restreints).
- ❖ Si le modèle est entraîné sur de plus grands jeux de données (14M-300M images), **l'entraînement à grande échelle l'emporte sur les biais inductifs**. Le ViT atteint ainsi de très bons résultats si pré-entraîné sur de gros dataset (ImageNet-21k et JFT-300M) puis transféré à des tâches avec moins de datapoints.

Améliorations envisagées

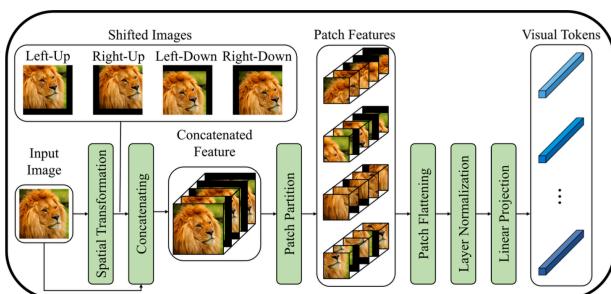
Solutions d'optimisation « classiques »

- Affiner les opérations de « data augmentation »,
- Entrainer les modèles sur plus de données,
- Optimiser les hyperparamètres clé (selon la littérature : learning rate, optimizer, weight decay et label smoothing)

Solutions envisagées dans la littérature et spécifiques à l'architecture du modèle

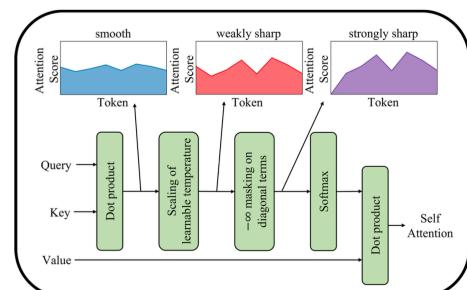
• Shifted Patch Tokenization (SPT)*

- Concaténation de différentes versions de la même image pour embedder plus d'informations spatiales dans les tokens et ainsi améliorer le biais inductif local du ViT.



• Locality Self-attention (LSA)*

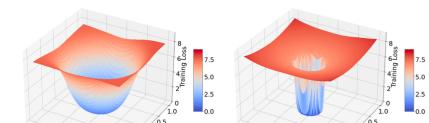
- Deux mécanismes pour jouer sur la smoothness de la distribution de la fonction softmax et ainsi améliorer le biais inductif local en forçant l'attention du ViT : (1) suppression de la composante diagonale de la matrice de similarité calculée par Query et Key pour exclure les relations intra-token de la fonction softmax et (2) un temperature scaling learnable (le ViT détermine la température de la fonction softmax durant l'entraînement).



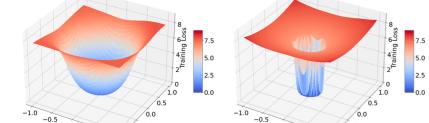
* SPT et LSA dans le papier « Vision Transformer for Small-Size Datasets » (Hoon Lee et al., 2021) et SAM dans le papier « When Vision Transformers outperform ResNets without pre-training or strong data augmentations » (Chen et al., 2022)

• Sharpness-aware minimizer (SAM)*

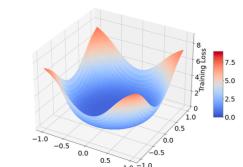
- Régularisation explicite de la géométrie des pertes avec l'optimiseur SAM : obtention d'un « paysage de perte » plus plats et d'une amélioration de la capacité à généraliser. Le ViT parvient alors à de meilleurs résultats sans pré-entraînement ou data augmentation.



(a) ResNet



(b) ViT



(d) ViT-SAM

Modèles alternatifs

- **DeiT** (Data-Efficient Image Transformer) de Facebook

“Training data-efficient image transformers & distillation through attention” (Touvron et all. 2021)

- Introduction d'un token de distillation, en plus du token [cls] : un paramètre « learnable », initialisé au hasard et positionné à la fin de la séquence. La catégorisation finale se basent sur ces deux tokens (qui ont leur propre fonction de cout).
- Le modèle s'appuie sur le principe de **Knowledge Distillation** (2015) suivant lequel 2 réseaux agissent de concert : un réseau fort, large et pré-entraîné (le professeur) et un réseau faible, petit et initialisé au hasard (l'étudiant). Le réseau étudiant apprend en imitant le modèle enseignant et exploite les connaissances de l'enseignant pour obtenir une meilleure accuracy.
- Les performances du modèle restent fortement dépendants de l'optimisation des hyperparamètres et de la data augmentation.

- **BEiT** (Bidirectional Encoder representation from Image Transformers) de Microsoft

“BEiT: BERT Pre-Training of Image Transformers” (Bao et all. 2021)

- Le BEiT tokenize les images en visual token, puis certains patches sont masqués au hasard et donnés au Transformer ; l'objectif de pré-entraînement est de retrouver les token originaux.
- C'est un ViT classique mais pré-entraîné de manière **self-supervised** : c'est la première fois que le ViT entraîné de manière self-supervised obtient de meilleurs résultats que l'entraînement supervisé.
 - ➡ Le modèle a battu le ViT original et le DeiT.

- **DINO** (Distillation with No Label) de Facebook

“Emerging Properties in Self-Supervised Vision Transformers” (Caron et all. 2021)

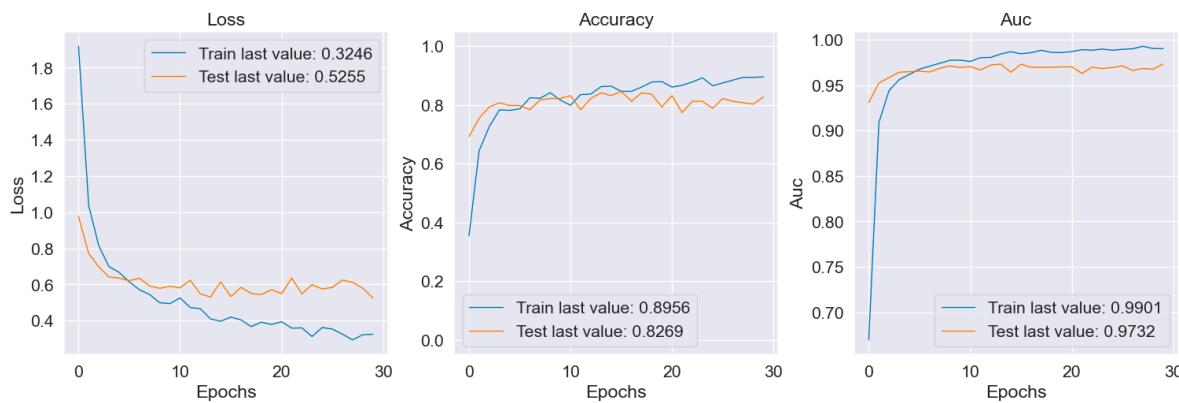
- DINO est une combinaison des deux modèles précédents : il fonctionne suivant le principe de Knowledge Distillation du DeiT et s'entraîne de manière non supervisée.
- A date, c'est le modèle avec les meilleurs résultats pour la tâche de classification*.

* <https://paperswithcode.com/sota/self-supervised-image-classification-on-1?p=beit-bert-pre-training-of-image-transformers>

Annexes

Training vs validation set

Efficient net



ViT L16

