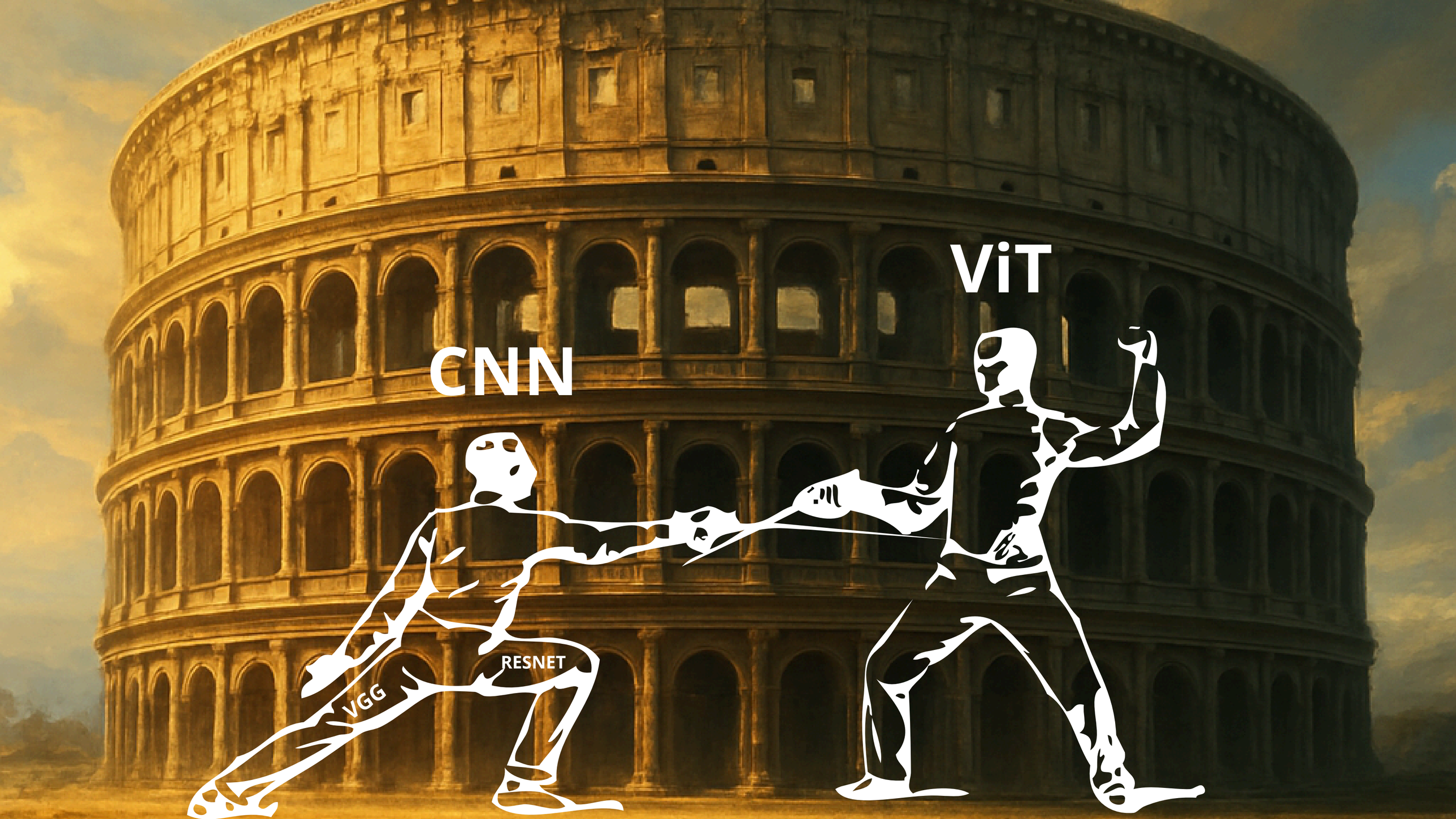

ViT
CNN
RESNET
VGG

Présentation des datasets

1) CIFAR-10

- 60000 images couleur de 32*32 pixels
- 10 classes

Motivation du choix

- Faible résolution
- Volume limité

Intuition : CNN >> ViT

Présentation des datasets

2) Tiny Imagenet

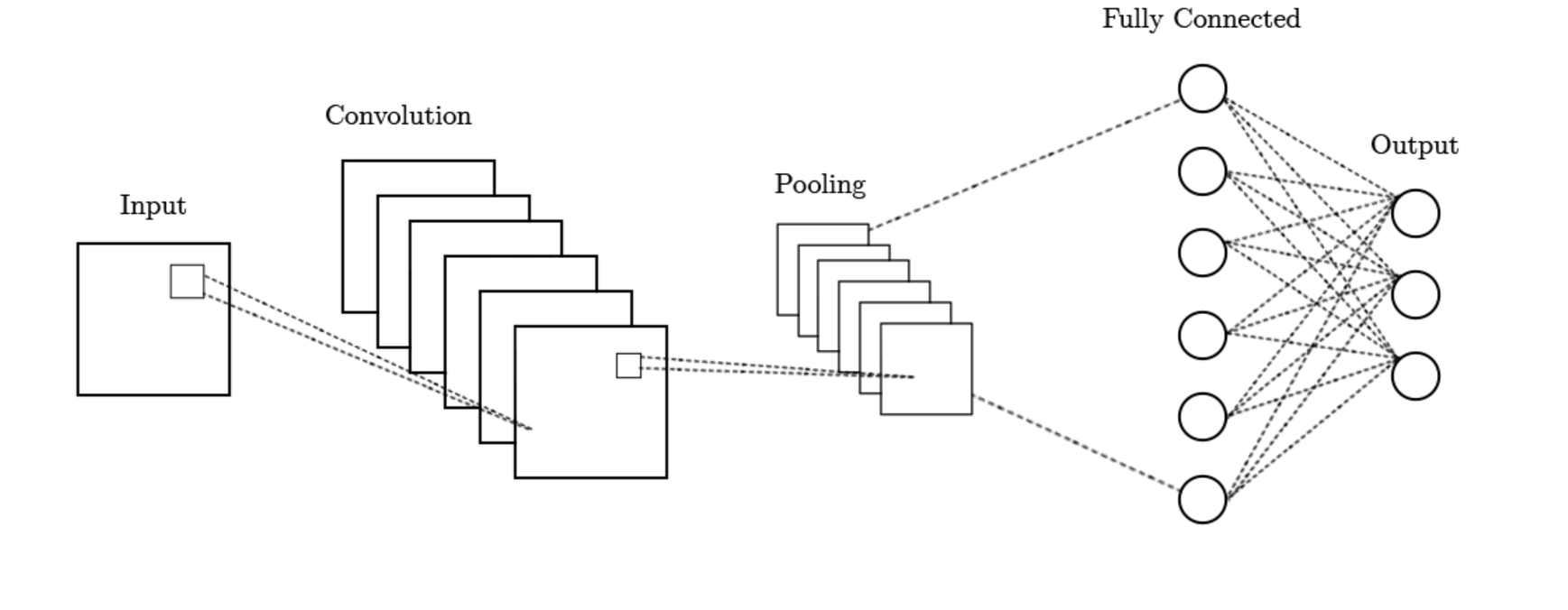
- 110000 images couleur de 64*64 pixels
- 200 classes

Motivation du choix

- Résolution doublée
- Volume plus important

Intuition : meilleurs résultats pour le ViT

Architecture CNN : From scratch



- Apprend les formes, textures et bords.
- Invariance spatiale & Hiérarchie visuelle.

- Prise de décision
- sur-apprentissage (overfitting)

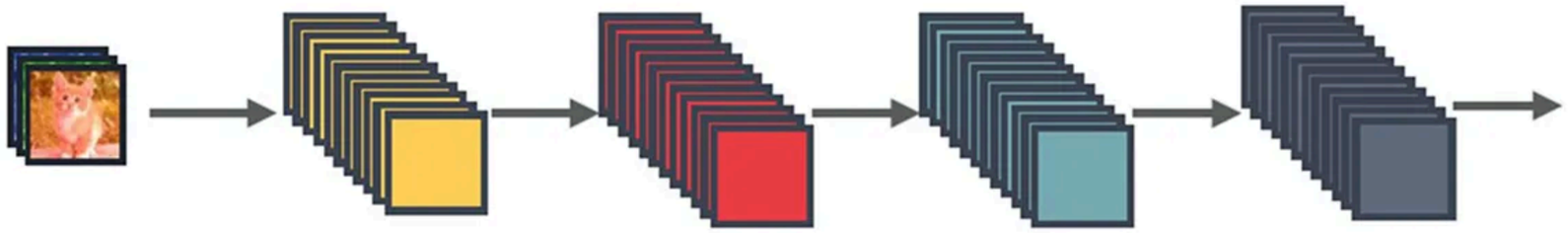
Localité (Pixel voisine)

Invariance par Translation

Efficacité (Small Data)

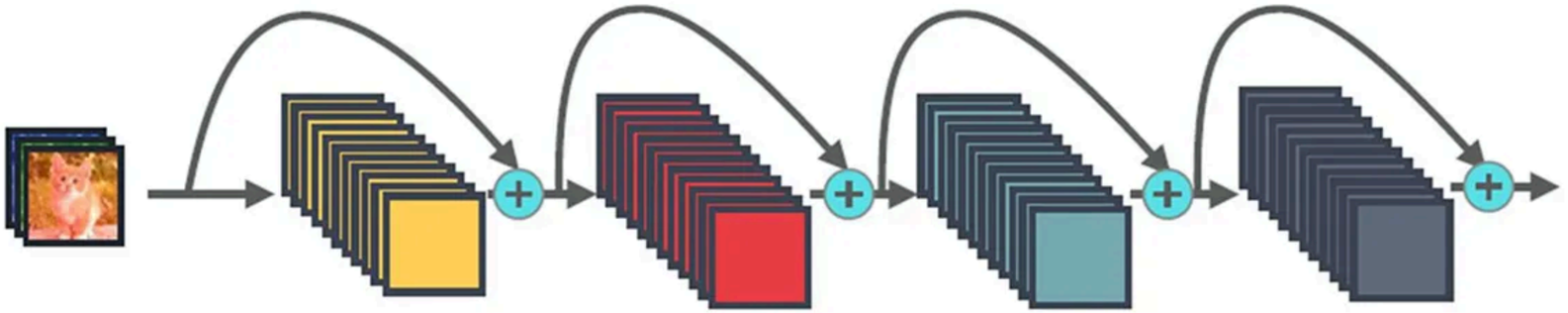
Architecture CNN : VGG / RESNET

1) VGG



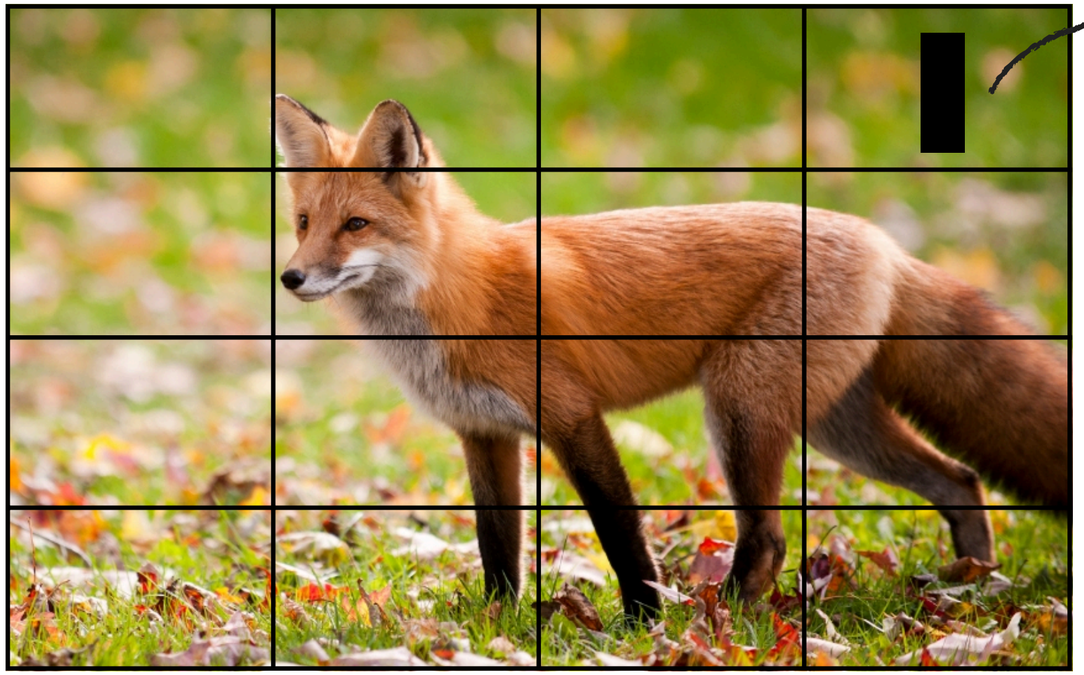
- Séquentiel (Conv + BN + ReLU)
- 6 couches (5 Convs + 1 FC)
- Max Pooling
- Empilement simple & profond

2) RESNET

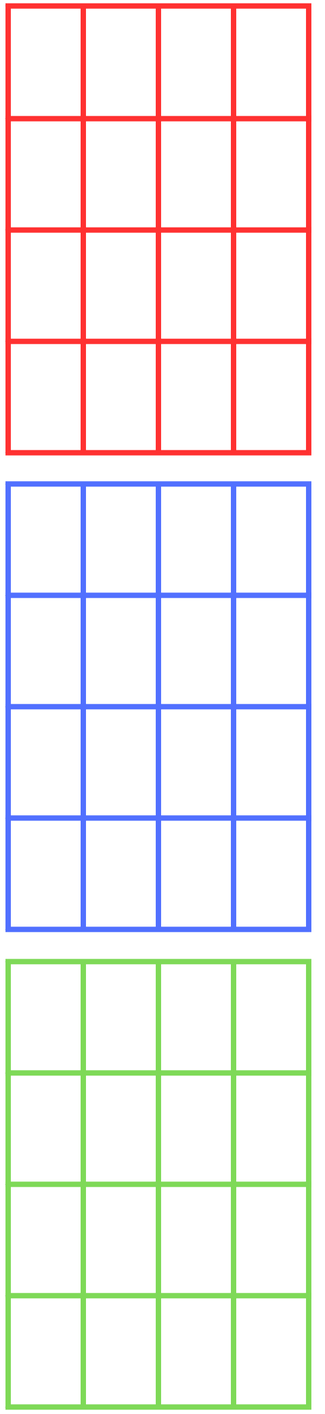


- Residual Block (avec Skip Connection)
- Convolution avec Stride=2 (moderne)
- 18 couches (17 Convs + 1 FC)
- Profondeur via résidus (x+F(x))

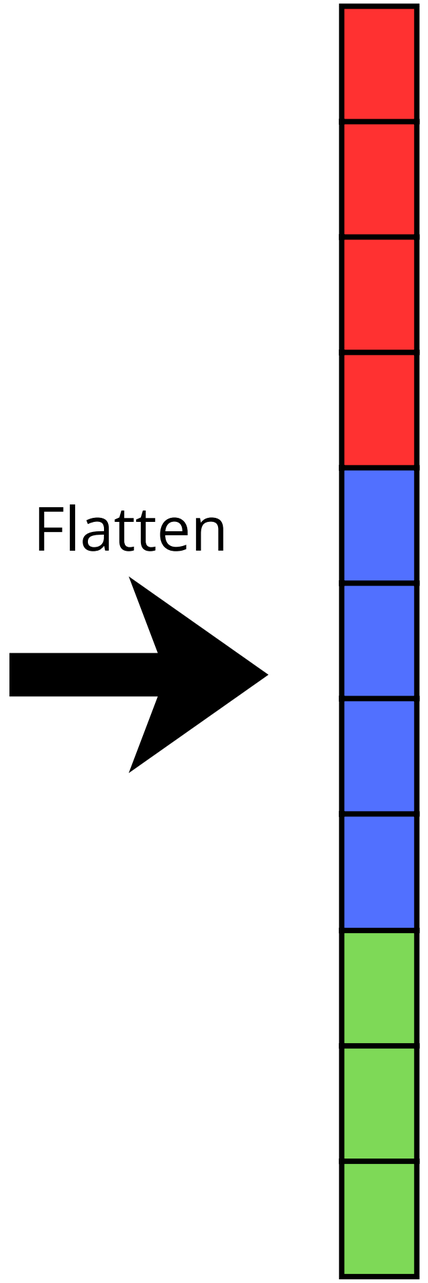
Vision Transformer

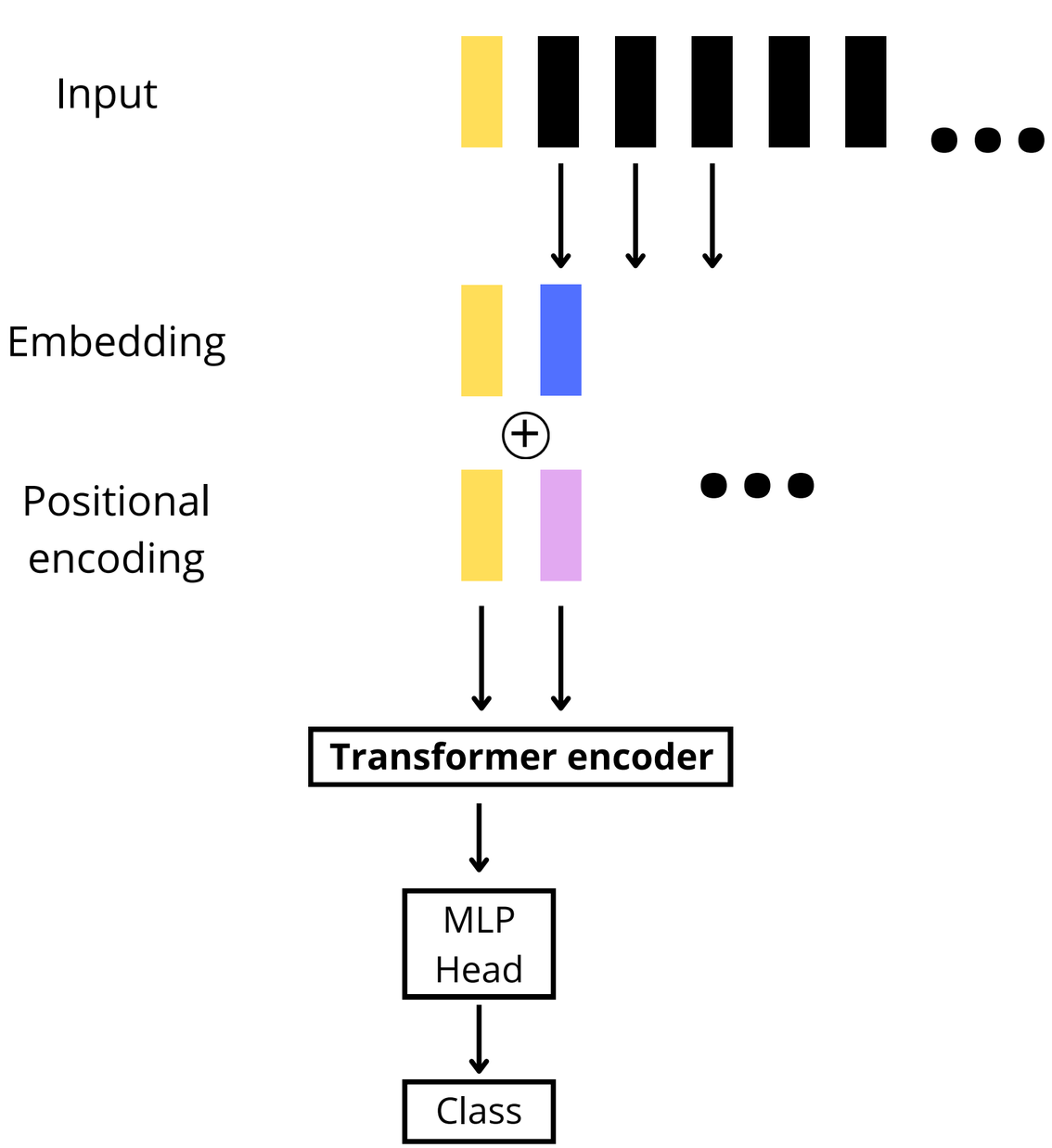


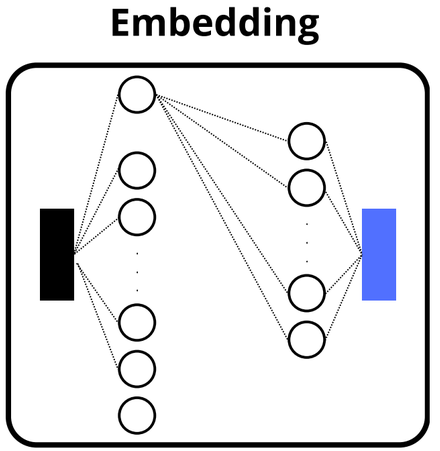
16 patches de 4*4 pixels

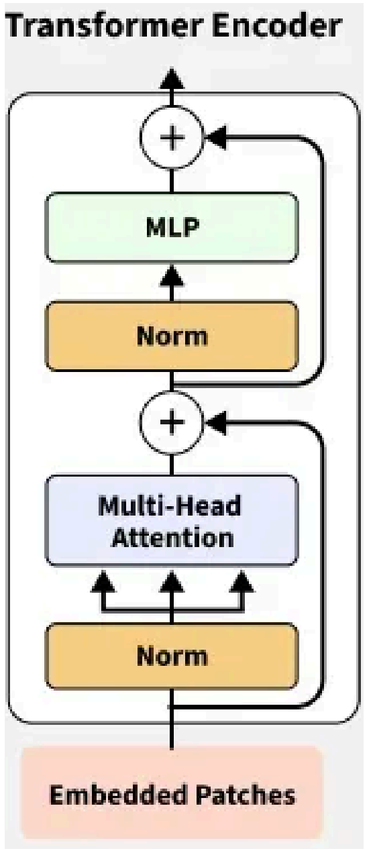


Flatten




Input
Embedding
Positional encoding
Transformer encoder
MLP Head
Class


Embedding


Transformer Encoder
MLP
Norm
Multi-Head Attention
Norm
Embedded Patches

Preprocessing

RandomCrop

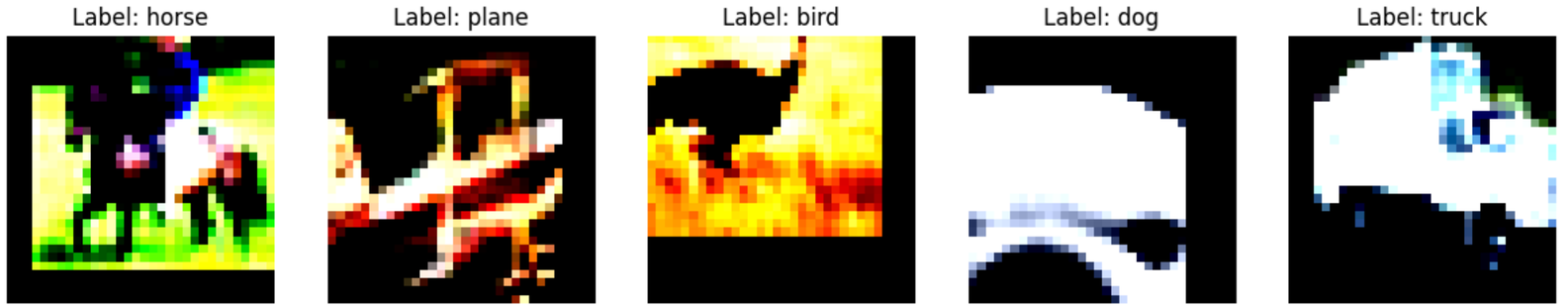
RandomHorizontalFlip

ColorJitter

Normalize

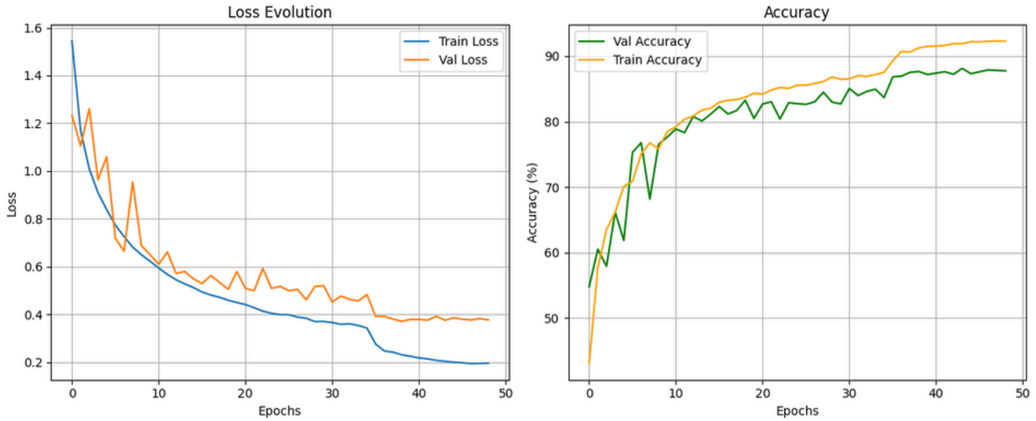
Avant

Après

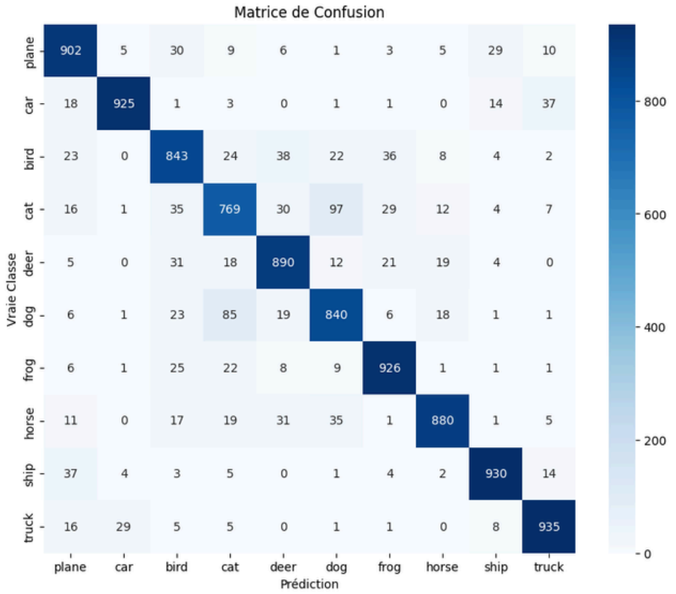


Architecture CNN : CIFAR-10 (32 x 32)

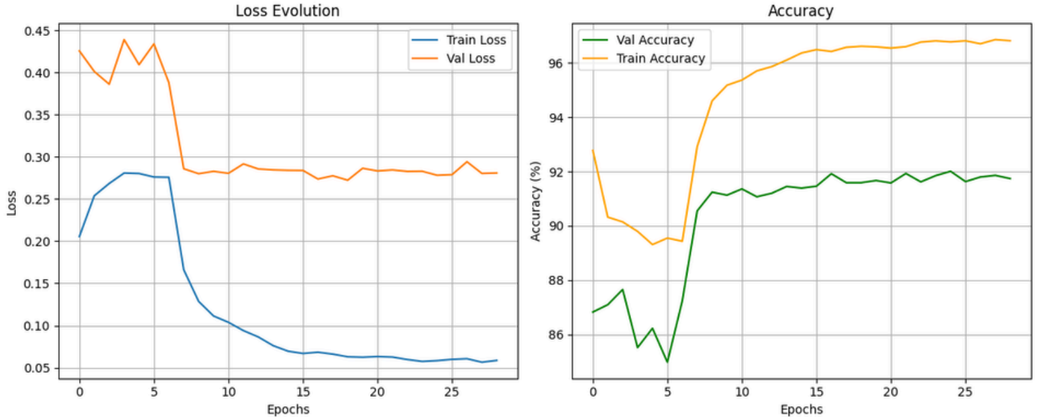
1) VGG



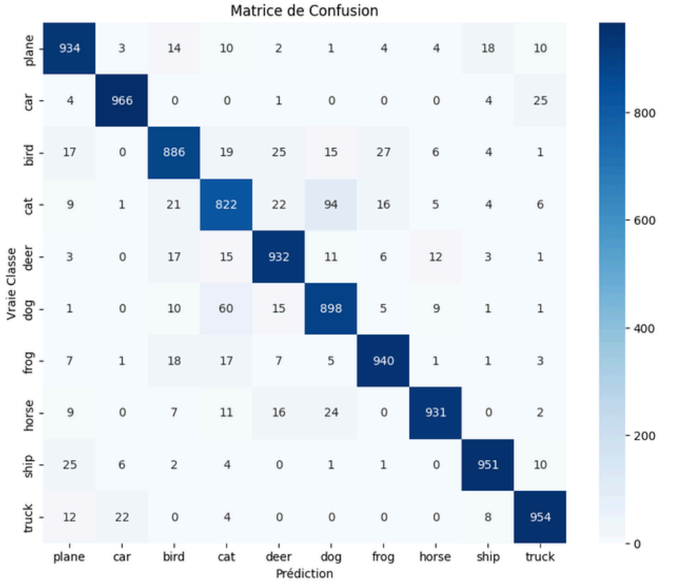
Accuracy : 88.40%



2) RESNET

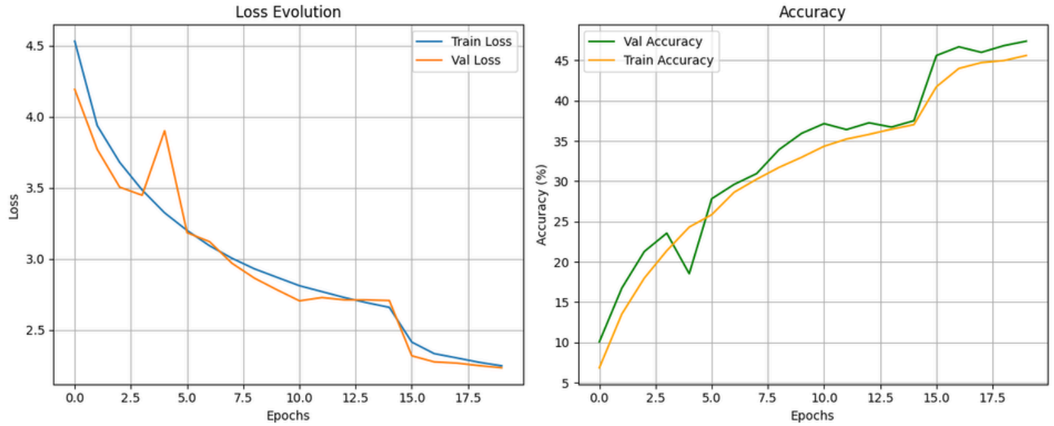


Accuracy : 92.14%



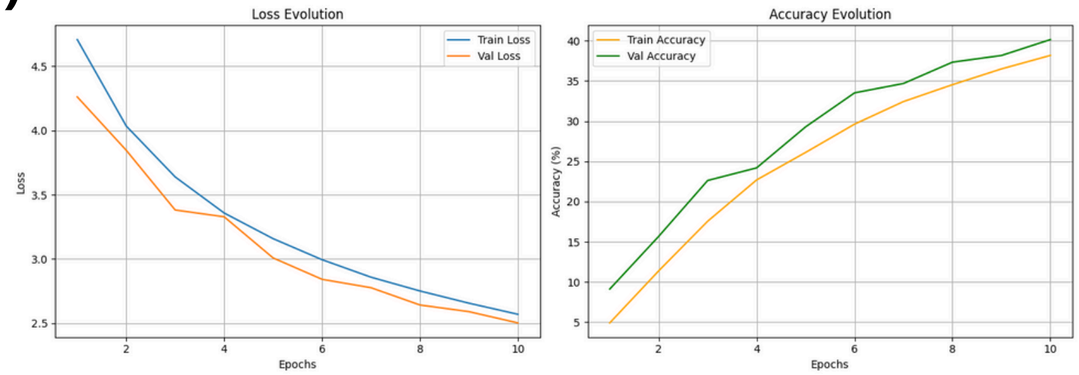
Architecture CNN : Tiny Imagenet (64 x 64)

1) VGG



Accuracy : 47.45%

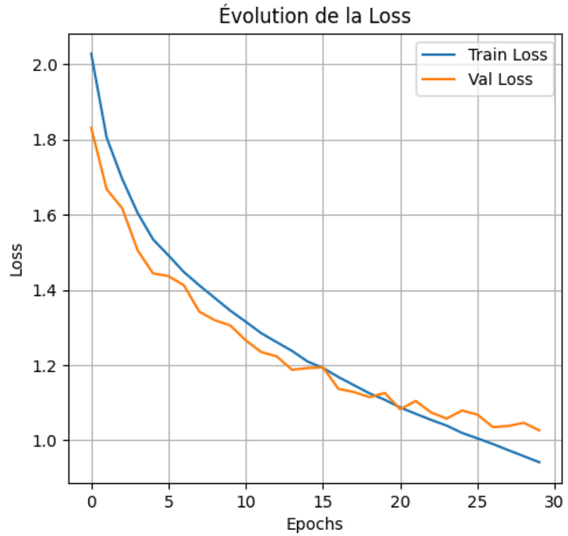
2) RESNET (stop à 10 epochs)

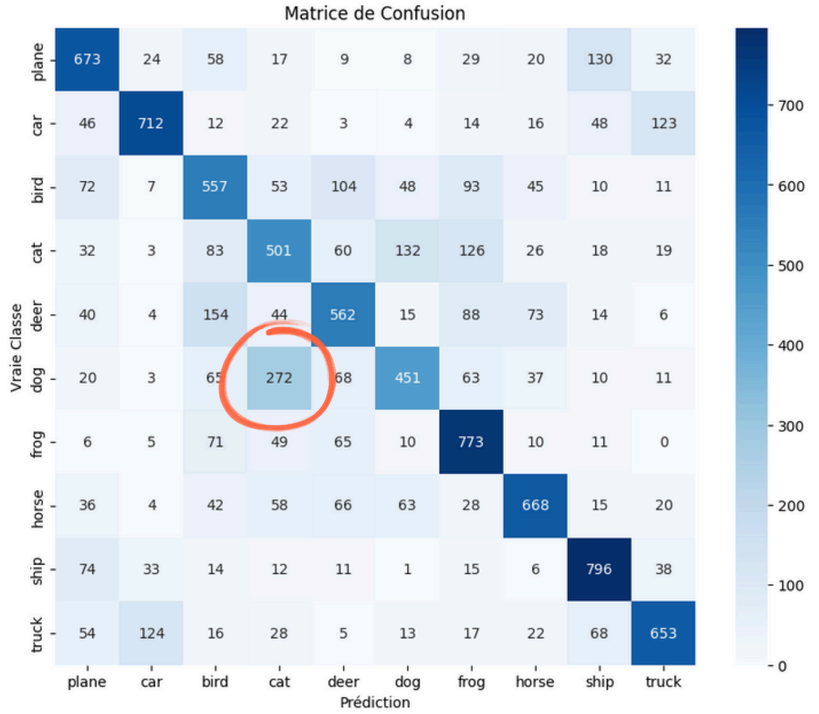


Accuracy : 43.60%

Résultats ViT

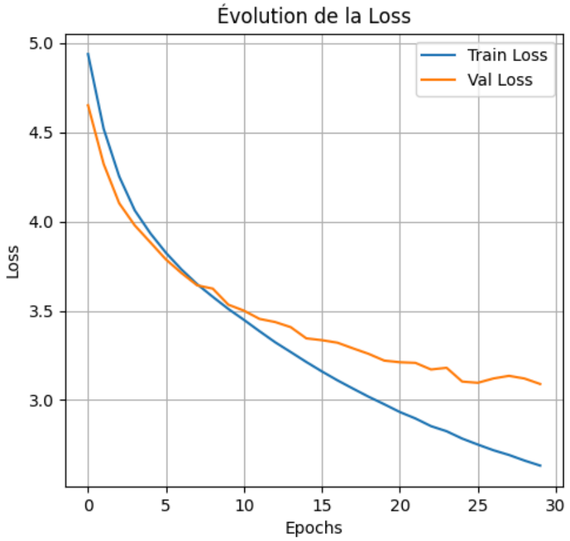
1) CIFAR-10





Accuracy : **64%**

2) Tiny Imagenet



Accuracy : **30%**

Tableau Récapitulatif

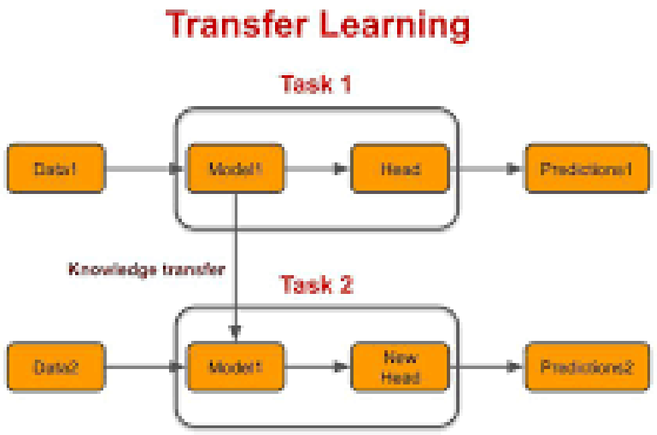
Dataset	Architecture	Modèle Spécifique	Accuracy	Temps d'entrainement	Pourquoi ce résultat ?
CIFAR-10 *(32x32 px, 50k images)*	**CNN** *(Fort Biais Inductif)*	**VGG** (539,786 param.) **ResNet** (2,797,610 param.)	**88%** **92%**	**364.39s** **757.31s**	Efficacité du Biais Inductif
	ViT *(Faible Biais Inductif)*	**ViT** "fait-maison" (811146)	**64%**		Déficit d'Apprentissage
Tiny ImageNet *(64x64 px, 100k images)*	**CNN** *(Fort Biais Inductif)*	**VGG** (2,246,088 param.) **ResNet** (11,271,432 param.)	**47.45%** ***43.60% (stop à 10 epochs)***	**2302.88s** **2922.43s**	Robustesse à la Complexité
	ViT 11,271,432 paramètres *(Faible Biais Inductif)*	**Compact ViT**	**...**	**...**	Double Peine : 1. Résolution 2. Data Hunger

Etude de robustesse

CNN vs ViT : Quand l’image se dégrade, quelle architecture survit ?

Méthodologie

1



Quelles architecture choisir pour chaque modèle ?

Quelles informations sont nécessaires avant de faire le finetuning

2

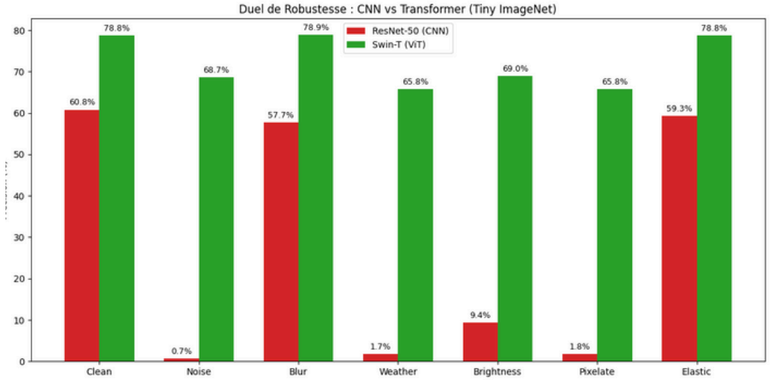
Perturbation du dataset de test

Dans quel ordre appliquer nos transformations ?

Quelles perturbations choisir pour réaliser la comparaison ?

3

Evaluation des résultats



La nécessité d’une précision relative

Motivation sur le choix des différentes métriques

Formulation d’hypothèses afin d’expliquer nos résultats

Transfer Learning

a) Modèles pré-entrainés

Resnet-50

- 23.9 M de paramètres
- CNN
- 2015 (Conférence CVPR 2016)
- Microsoft
- Connexions Résiduelles (Skip connections)

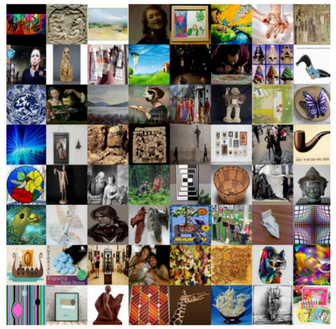
Swin

- 27 M de paramètres
- Hierarchical Vision Transformer
- 2021 (Conférence ICCV 2021)
- Microsoft Asia
- Attention à Fenêtres Décalées (Shifted Windows)

b) Dataset d'entrainement

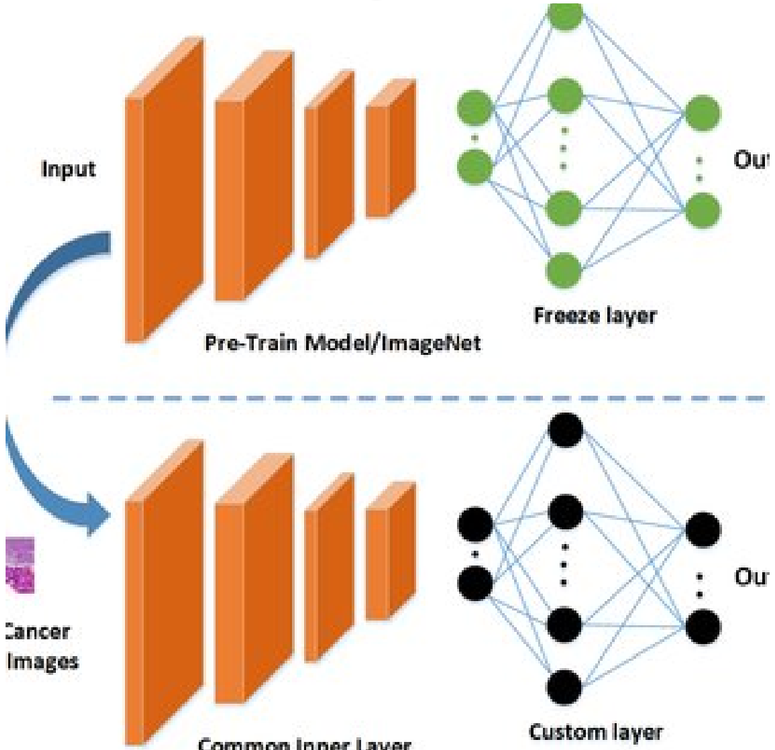
- **Nom** : ImageNet (ILSVRC 2012).
- **Volume** : 1.28 million d'images d'entraînement.
- **Diversité** : 1 000 classes différentes (animaux, objets du quotidien, véhicules, nourriture, etc.)
- **Résolution** : 224*224

c) La Tâche Cible : Tiny ImageNet



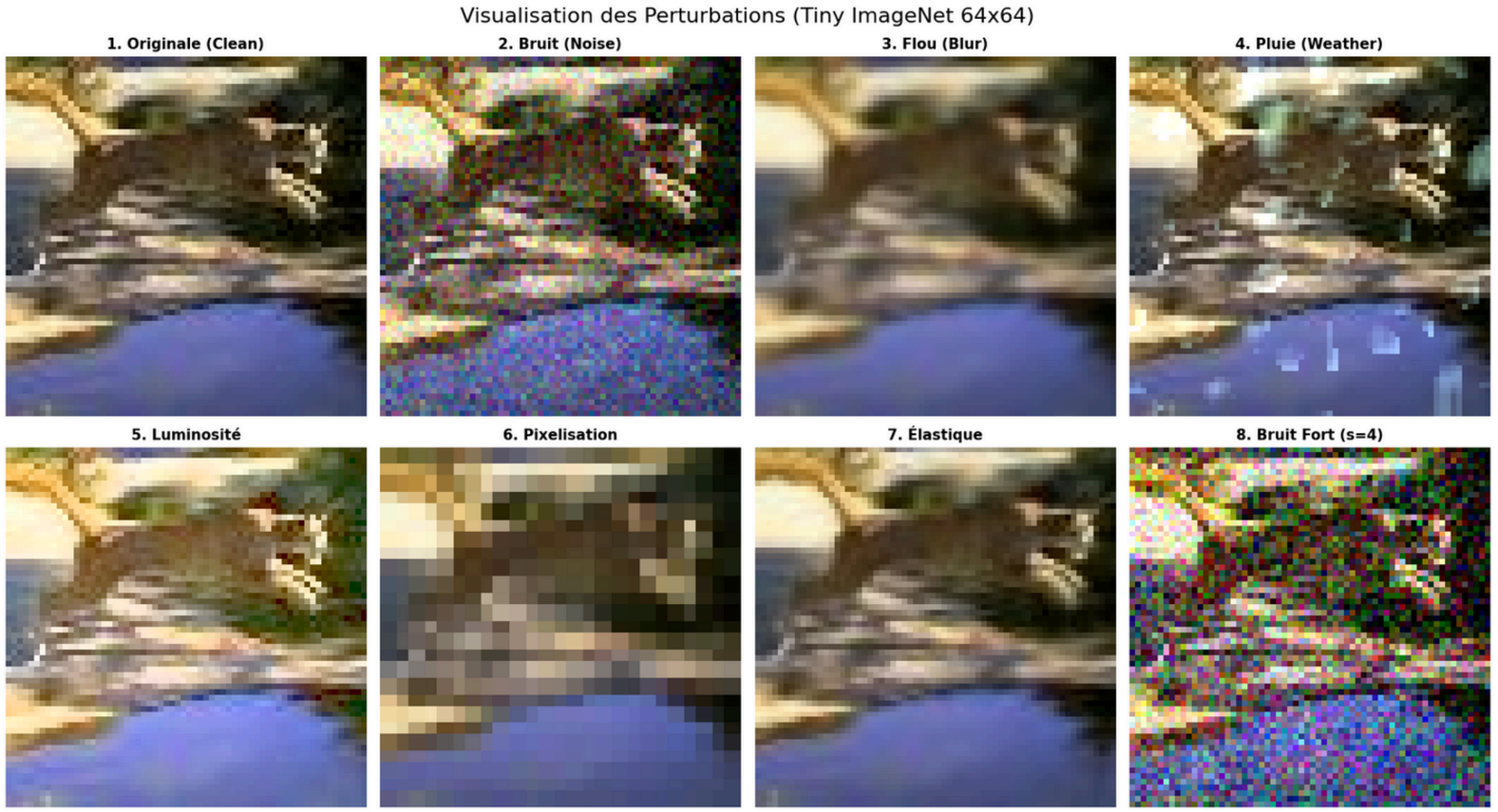
- 110000 images couleur de 64*64 pixels
- 200 classes

d) Transfer Learning



Métrique	ResNet-50 (CNN)	Swin-Tiny (ViT)
Temps	**~45 min (3 epochs)**	**~55 min (3 epochs)**
Précision Train	**63%**	**78%**
Précision Validation	**60.81%**	**78.75%**

Perturbations des images



Mais nos modèles ont été entrainés sur des images 224*224 normalisées

Pipeline : 1) resize → 2) corruption → 3) normalization

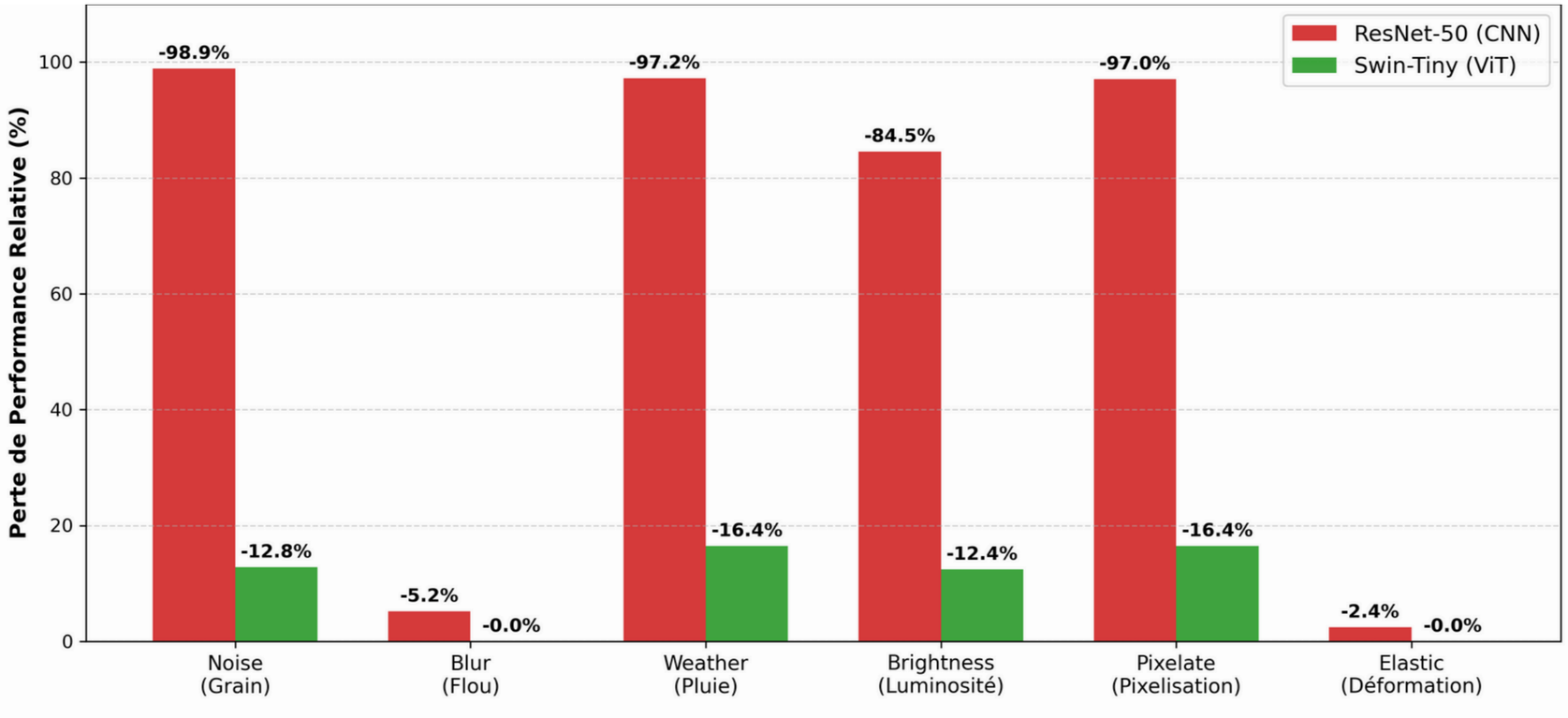
Evaluation des résultats

Swin-T a une accuracy plus élevé que ResNet

=> On définit une métrique relative pour évaluer la précision après corruption

$$\text{Drop Rate}(\%) = \left(\frac{\text{Acc}_{\text{clean}} - \text{Acc}_{\text{corrupted}}}{\text{Acc}_{\text{clean}}} \right) \times 100$$

Evaluation du Drop rate pour chaque corruption



- Le Transfer Learning ne suffit pas à garantir la robustesse
- Effondrement du ResNet sur les perturbations à haute fréquence (Bruit/Pixelisation, Drop > 95%) vs Résilience du Swin
- Les CNN sont robustes aux perturbations géométriques et basses fréquences qui préservent la cohérence locale

Shape bias : Geirhos et al., "ImageNet-trained CNNs are biased towards texture", ICLR 2019.

Visualisation des Espaces latents

	Drift Noise	Drift Blur
Swiny-Tiny	**0.67**	**0.11**
ResNet-50	**0.58**	**9.32**

$$\text{Normalized Drift}(\%) = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{\|x_{\text{clean}}^{(i)} - x_{\text{noise}}^{(i)}\|_2}{\|x_{\text{clean}}^{(i)}\|_2}\right)$$