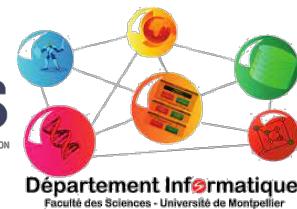




CENTRE D'ECOLOGIE  
FONCTIONNELLE  
& EVOLUTIVE



**tetis**  
TERRITOIRE ENVIRONNEMENT TÉLÉDÉTECTION  
INFORMATION SPATIALE



Département Informatique  
Faculté des Sciences - Université de Montpellier

Université de Montpellier  
Faculté des Sciences

## MASTER 2 INFORMATIQUE Parcours IMAGINE

IA et modélisation de la beauté: apport des autoencodeurs variationnels pour caractériser l'efficacité du traitement cérébral de l'information

Roland BERTIN-JOHANNET

Tuteurs dans l'établissement d'accueil : Jérôme Pasquet et Julien Renoult  
Tuteur à l'université: William Puech

# Contents

<b>1 Bref résumé du stage</b>	<b>4</b>
<b>2 Présentation de l'organisme d'accueil</b>	<b>5</b>
<b>3 Présentation de la mission</b>	<b>5</b>
3.1 La beauté . . . . .	5
3.2 Point de vue écologique . . . . .	5
3.2.1 La théorie du handicap : . . . . .	5
3.2.2 La dérive de Fisher : . . . . .	5
3.2.3 L'exploitation de préférences latentes : . . . . .	6
3.3 Motivation et idée du stage . . . . .	6
3.3.1 Théorie de la Fluence . . . . .	6
3.3.2 Modélisation du cerveau avec les Réseaux de Neurones Convolutifs (CNN) . . . . .	7
3.3.3 Autoencodeurs Variationnels . . . . .	7
3.3.4 Assemblage des concepts introduits . . . . .	12
<b>4 Environnement technique</b>	<b>12</b>
4.1 Cluster de calcul Meso-LR . . . . .	12
4.2 Supercalculateur Jean Zay . . . . .	12
4.3 Langage, librairies, Pages GitHub . . . . .	13
<b>5 Méthode générale</b>	<b>14</b>
5.1 Bases de données . . . . .	14
5.2 Modélisation de la fluence de traitement neuronal (métriques) . . . . .	14
5.2.1 Mesures de sparsité . . . . .	14
5.2.2 Qualité de reconstruction . . . . .	14
5.2.3 Sharpness Aware Minimization . . . . .	14
5.2.4 Attention . . . . .	16
5.3 Développement d'un Autoencodeur Variationnel . . . . .	17
5.3.1 Méthodes d'upsampling : . . . . .	19
5.3.2 Différences dans l'espace latent : . . . . .	19
5.3.3 Fonctions de coût : . . . . .	20
5.3.4 Méthodes de régularisation : . . . . .	20
5.3.5 Connexions résiduelles : . . . . .	20
5.3.6 Différents modèles . . . . .	20
5.3.7 Étude des représentations . . . . .	20
<b>6 Tâches</b>	<b>27</b>
6.1 Reprise du stage de l'année précédente . . . . .	27
6.2 Mécanismes d'attention . . . . .	27
6.2.1 Attention en profondeur avec sigmoïde et contrainte de sparsité (pas de résiduel) . . . . .	27
6.2.2 Attention en profondeur avec softmax à température variée . . . . .	27
6.2.3 Limites de notre approche sur l'attention : . . . . .	28
6.3 Spécialisation des modèles visuels aux ethnies et sexes . . . . .	28
6.3.1 Origine de l'idée . . . . .	28
6.3.2 Interprétation biologique . . . . .	30
6.3.3 Mise en place . . . . .	30
<b>7 Ajouts à la méthode</b>	<b>32</b>
7.1 Tentatives de reconstruction sans décodeur . . . . .	32
7.1.1 La motivation : . . . . .	32
7.1.2 La méthode : . . . . .	32
7.1.3 Le problème rencontré : . . . . .	32
7.1.4 Exemples adversariels : . . . . .	32
7.2 Utilisation d'une contrainte de sparsité . . . . .	33
7.2.1 Motivation biologique : . . . . .	33

7.2.2	La contrainte mise en place . . . . .	33
<b>8</b>	<b>Pistes abandonnées</b>	<b>41</b>
8.1	Pistes abandonnées . . . . .	41
8.1.1	Normalisation Spectrale . . . . .	41
8.1.2	Autres contraintes de sparsité . . . . .	41
8.1.3	Generative Invertible Flows . . . . .	42
8.1.4	Information mutuelle . . . . .	42
8.1.5	Convolutions Séparables en Profondeur . . . . .	43
8.1.6	Pistes pour la reconstruction sans décodeur . . . . .	45
<b>9</b>	<b>Résultats</b>	<b>47</b>
9.1	Analyse statistique en lien avec la beauté . . . . .	47
9.1.1	Corrélations . . . . .	47
9.1.2	Modèles linéaires . . . . .	49

## **Remerciements**

Je remercie Théo Oriol pour m'avoir prêté son accès au supercalculateur Jean Zay, Sonia Mai pour m'avoir aidé à régler les problèmes de compte sur le mésocentre Meso-LR, Nicolas Dibot pour m'avoir accueilli dans son bureau et aidé à m'installer au CEFE.

Je remercie Julien Renoult et Jérôme Pasquet pour leur encadrement de qualité, la liberté de manœuvre qu'il m'ont laissée tout en me fournissant un foisonnement d'idées.

Je remercie aussi William Puech de s'être assuré à plusieurs reprises que tout se passait bien dans le stage.

# Introduction

## 1 Bref résumé du stage

Nous fournissons ici un bref résumé du stage pour pouvoir se situer par la suite. Des définitions plus détaillées des différents termes sont données tout au long du rapport.

L'objectif était de tester différentes métriques de Fluence (la facilité de traitement de l'information dans le cerveau) comme prédicteurs de la beauté. Pour modéliser le traitement de l'information dans le cerveau, l'idée était d'utiliser des Autoencodeurs Variationnels (VAE), qui sont des modèles de Deep Learning.

Dans un premier temps, nous avons cherché à développer un VAE qui réponde à nos diverses besoins. Cela a pris un certain temps car dans l'état de l'art, les modèles les plus performants avaient une architecture s'éloignant des modèles du cerveau, et les modèles simples et proches des modèles de neurosciences avaient des performances insuffisantes.

Ensuite, nous avons cherché à améliorer nos modèles de deux manières : d'abord, nous avons exploré différentes manières de se passer du décodeur de notre autoencodeur (explications plus loin), afin de se débarrasser de son biais. Ensuite, nous avons cherché à ajouter une contrainte de sparsité (définie plus bas) à l'entraînement de notre VAE pour le rendre plus réaliste biologiquement.

Nous avons alors cherché à mettre en place différentes métriques de fluence sur nos réseaux entraînés, et réalisé différentes analyses statistiques sur ces métriques.

Après avoir exploré ces métriques de fluence, nous avons commencé à nous intéresser à des mécanismes d'attention, et comment l'attention que l'on porte sur une image peut indiquer qu'on la trouve belle ou pas belle. Nous considérions cette approche au moins partiellement indépendamment de la théorie de la Fluence ; néanmoins nous nous sommes rendus compte qu'avec les mécanismes d'attention de l'état de l'art en Deep Learning, il était difficile de représenter le type d'attention que nous avions en tête. Nous avons alors interprété notre mesure d'attention comme une mesure de complexité des images, et réalisé des analyses statistiques dessus.

Cela n'a pas encore été réalisé au moment du rendu de ce rapport ; néanmoins, l'objectif dans le dernier mois est de rassembler le travail de développement réalisé pendant le stage dans une interface de code facile d'utilisation, avec une documentation et potentiellement des vidéos explicatives.

## 2 Présentation de l'organisme d'accueil

**TETIS** (Territoires, environnement, télédétection et information spatiale) : L'UMR Tetis s'est structurée autour de deux dimensions scientifiques qui se retrouvent dans son intitulé : "Territoires, environnement (dimension thématique), télédétection et information spatiale (dimension méthodologique).  
**CEFE** : Le CEFE est un des plus importants laboratoires de recherche en Ecologie en France.

Le projet du CEFE vise à comprendre la dynamique, le fonctionnement et l'évolution du vivant, de «la bactérie à l'éléphant», et «du génome à la planète».

## 3 Présentation de la mission

### 3.1 La beauté

La Beauté est un phénomène étudié depuis des milliers d'années (on en retrouve des discussions explicites chez Platon et Aristote par exemple). Historiquement, les explications de ce phénomène étaient d'abord **universalistes** (chez Platon, le beau correspond à ce qui est bon et avantageux), avant que la dimension **subjective** soit prise en compte (par Hume par exemple). Les approches universalistes expliquent les caractéristiques qui font de n'importe quel objet un objet beau (symétrie, propreté, santé, etc.), là où les approches subjectives soulignent que chaque observateur perçoit l'objet différemment et par conséquent le phénomène de beauté dépend de l'observateur. Ce stage s'inscrit dans un projet d'étude d'une approche **interactionniste** de la beauté (interactionniste : la beauté provient de l'interaction de l'observateur avec l'observé, et non pas de l'un seul ou de l'autre seul) : la **théorie de la Fluence**, selon laquelle *est beau un stimulus qui est traité avec efficience par le système sensoriel qui le perçoit*.

### 3.2 Point de vue écologique

Ce stage a eu lieu au CEFE, qui est un laboratoire d'écologie. Le projet écologique derrière notre étude de la beauté est de savoir si les animaux ont une sensibilité au Beau, et si oui, quels en sont les origines et les effets. Certainement, à nos yeux, la queue du paon, le chant des oiseaux et les mouvements agiles des félins sont beaux. Il existe plusieurs théories qui tentent d'expliquer l'évolution de ces signaux dans la nature [42]. En voici trois :

#### 3.2.1 La théorie du handicap :

[4] Selon cette théorie, les animaux qui possèdent un surplus de ressources investiraient dans un "handicap" (la queue gigantesque du paon qui l'empêche de voler par exemple) que d'autres animaux ne pourraient pas se permettre d'arborer. Aux yeux des potentiels partenaires reproductifs, la présence de ce handicap serait une assurance que l'individu possède plus de ressources que les autres. Ce mécanisme pourrait expliquer l'extravagance des signaux sexuels, mais n'en explique pas la spécificité (la queue du paon est longue, mais qu'en est-il des motifs raffinés et élaborés visibles sur les plumes de cette queue), les aspects universels ni la diversité.

#### 3.2.2 La dérive de Fisher :

[39] [5] Cette théorie introduit un mécanisme selon lequel si une préférence existe pour une certaine caractéristique chez les partenaires reproductifs, alors 1) avoir cette caractéristique est avantageux, et 2) avoir cette préférence l'est aussi, puisqu'elle donne une descendance plus attractive. Ainsi, plus la caractéristique est préférée, plus il est avantageux de la préférer. En ajoutant à cela le fait qu'avec la préférence pour la caractéristique, la présence de la caractéristique dans la population augmentera, on met à jour un mécanisme de *course en avant* qui pousse à l'extrême l'expression d'une caractéristique arbitraire. Cette *course en avant* explique à la fois l'extravagance des signaux, mais aussi leur diversité, puisque le mécanisme s'applique à une caractéristique arbitraire. Ce mécanisme peut aussi expliquer la spécificité des signaux, mais pas les aspects universels.

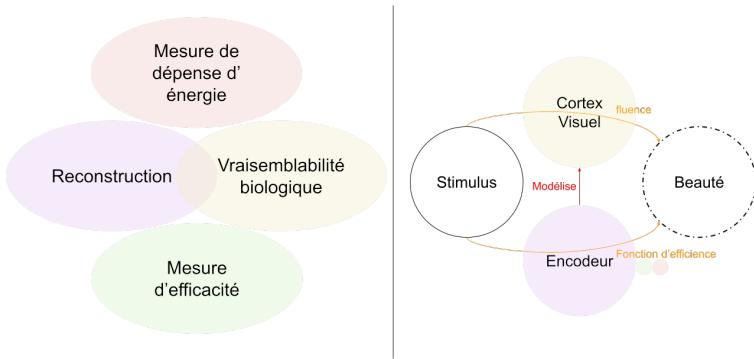


Figure 1: L’objectif du stage est d’établir des métriques d’efficience et d’efficacité pour extraire de notre modèle du cortex visuel, un VAE, une mesure de fluence que nous tentons de comparer à des scores de beauté sur des visages.

### 3.2.3 L’exploitation de préférences latentes :

[45] Selon cette théorie, il y aurait des biais chez les animaux qui les pousseraient à préférer arbitrairement certains signaux plutôt que d’autres, sans tenir compte de l’utilité de leur signification. Une telle préférence pourrait, par exemple, être le résultat d’un biais dans le système visuel qui cause du plaisir en réaction à certains motifs de fourrure, ou certaines couleurs d’écailles. Une telle préférence est latente car elle est un effet de bord de l’évolution d’un système perceptuel fait pour résoudre des problèmes pratiques. Des signaux sexuels évolueraient alors qui exploitent ces préférences latentes. Cette théorie peut expliquer à la fois l’extravagance (exploitation au maximum de la préférence), la diversité (les systèmes perceptuels sont diverses), les aspects universels (mais partagent des caractéristiques communes), et la spécificité (peut-être que les motifs sur la queue du paon sont adaptés pour créer une réaction particulière dans le cerveau de l’observateur).

**Finalement :** Les animaux ont-t-ils, comme le pensait Darwin, un goût esthétique ? Contrairement aux humains, nous ne pouvons pas le leur demander. En conséquence, nous cherchons à établir chez les humains un modèle mathématique de la beauté, qui sera ensuite testé chez les animaux.

## 3.3 Motivation et idée du stage

Dans le cadre de ce stage, nous testons la théorie de la Fluence comme explicatrice du phénomène de beauté chez les humains (voir figure 1 pour plus de détails). Pour comprendre le modèle mathématique que nous mettons en place, il est nécessaire de se familiariser avec la **théorie de la Fluence**, la modélisation du cerveau par les **Réseaux de Neurones Convolutifs**, et avec les **Autoencodeurs Variationnels**. [15] [44] [28]

### 3.3.1 Théorie de la Fluence

- La théorie de la fluence est une des théories dominantes pour expliquer le phénomène de la beauté [57]. Elle énonce que la beauté émerge en conséquence d’une facilité du traitement de l’information dans le système perceptuel[41]. Cette théorie explique ainsi la préférence pour les prototypes (des stimulus qui présentent les caractéristiques générales de leur catégorie)[56], la symétrie, etc.

Il est intéressant de spécifier qu’un défaut de cette théorie est qu’elle n’explique pas le phénomène d’ennui face à un stimulus trop simple. Une approche dans [11], basée sur la *théorie à processus*

*duaux*, ajoute à la théorie de la fluence un "besoin d'enrichissement cognitif" qui pousse le sujet à rechercher la difficulté de traitement de l'information et à l'apprécier.

- La fluence du traitement de l'information peut être décomposée en deux aspects [44] : l'**efficacité** du traitement de l'information, c'est-à-dire la quantité d'information extraite, et l'**économie** du traitement de l'information, qui est plus grande si moins d'énergie a été dépensée pour extraire l'information.

- Des approches de modélisation mathématique de la fluence ont déjà été réalisées.

**Approches sans Réseaux de neurones :** Dans [43], des images de visages sont exprimées comme combinaisons de filtres prédéfinis semblables aux filtres trouvés dans le cortex visuel humain. Une mesure de la sparsité d'utilisation de ces filtres dans la combinaison est montrée comme bon prédicteur de la beauté de ces visages. Dans [2], la fluence du traitement de l'information est modélisée comme sa correspondance à une attente de l'observateur. Des concepts d'apprentissage par renforcement sont appliqués pour ajouter une dimension d'apprentissage à cette attente.

- **Approches avec Réseaux de Neurones :** Dans l'équipe E3CO où le stage s'est déroulé, deux approches ont déjà été réalisées : Melvin Bardin a modélisé la fluence comme la typicalité de la réaction du système visuel à une image, Nicolas Dibot l'a modélisée comme la sparsité des activations dans le système visuel.

### 3.3.2 Modélisation du cerveau avec les Réseaux de Neurones Convolutifs (CNN)

- **Contexte en neurosciences :** Dans une expérience très célèbre (voir figure 2), Hubel et Wiesel [18] ont démontré (en simplifiant) que certains neurones (appelés "cellules simples") dans le cortex visuel des chats répondent sélectivement à des barres visuelles d'une certaine orientation, à un certain endroit sur la rétine. D'autres cellules (appelées "cellules complexes") répondent à des barres d'une certaine orientation, dans une certaine zone dans la rétine. Le mécanisme proposé est que les cellules complexes intègrent les réponses des cellules simples par exemple en s'activant dès qu'une d'entre elles s'active. Le motif détecté par la cellule complexe est plus abstrait que celui détecté par la cellule simple (voir figure 4). Ainsi, une succession de couches de neurones intégrant chacun l'information de plusieurs neurones de la couche précédente permettrait aux cellules situées en haut de la hiérarchie de s'activer en réponse à des stimuli très complexes (par exemple, au visage de notre grand-mère, avec les "grandmother cells").
- **Réseaux de neurones convolutifs :** dans [8], le modèle du neocognitron est proposé, un modèle du cortex visuel basé sur les trouvailles de Hubel et Wiesel. Yann LeCun, inspiré partiellement par le noécognitron, crée ensuite les Réseaux de Neurones Convolutifs (CNN) [26] qui reposent sur une opération de convolution où les neurones s'activent en réaction à un certain motif (cellules simples) et une opération de *pooling*, où les activations de plusieurs neurones sont intégrées en une seule (cellules complexes) avec un opérateur max, ou moyenne par exemple (voir figure 3).
- **Validité comme modèle du cerveau** Au delà du fait que les CNNs sont directement inspirés du fonctionnement du système visuel, il existe de nombreux résultats permettant de penser qu'ils en sont de très bons modèles [28] : Les CNN ont tendance à faire le même type d'erreurs que les humains sur des tâches de classification d'images. De plus, les représentations neuronales apprises pendant l'entraînement des CNN sont structurellement corrélées aux représentations dans le cerveau, et les activations dans un CNN sont de bons prédicteurs des activations dans le cerveau.

### 3.3.3 Autoencodeurs Variationnels

- **Différents types d'apprentissage en Apprentissage Automatique :** L'apprentissage supervisé fait référence à une méthode d'apprentissage basée sur des paires (entrée, sortie) où l'on connaît une sortie pour chaque entrée, et le modèle doit apprendre à prédire la sortie. L'apprentissage non-supervisé est une méthode d'entraînement où le modèle apprend une sortie sans qu'on puisse lui donner d'exemple de ces sorties. L'exemple le plus connu d'apprentissage non-supervisé est l'algorithme du K-Means. Certains articles [51] pointent vers l'avantage de

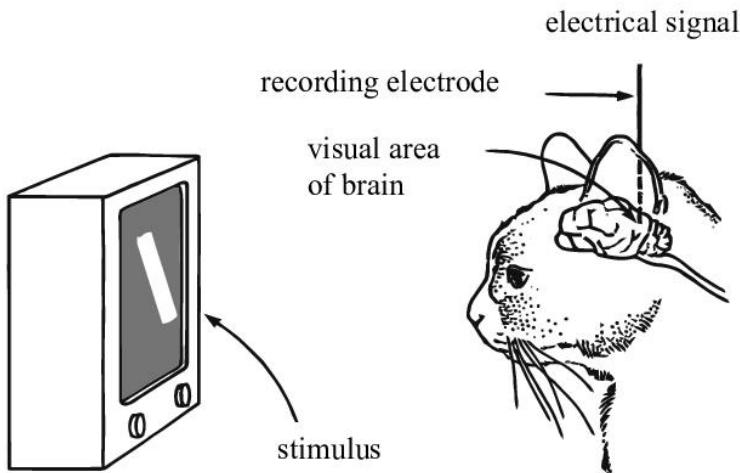


Figure 2: Dispositif mis en place par Hubel et Wiesel : un électrode mesure les fréquences d'activation d'un seul neurone, alors que le chat regarde un écran où est affichée une barre à différentes orientations.

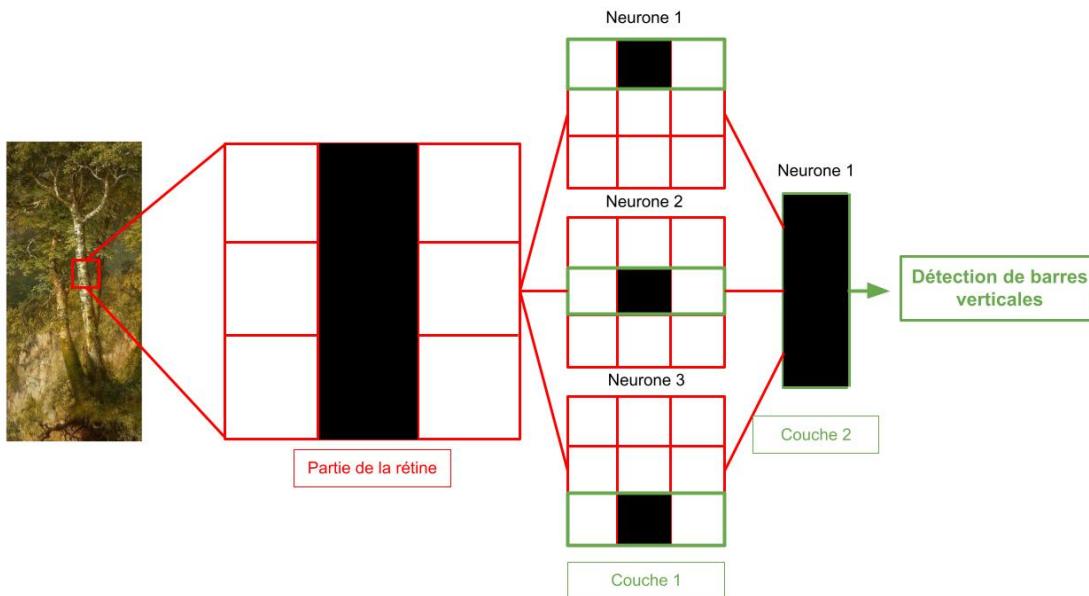


Figure 3: Une des diverses manières de comprendre intuitivement le fonctionnement des CNN, en rapport avec le cortex visuel : si l'on simplifie le tronc comme étant une barre verticale, on peut voir que les trois neurones de la couche 1, s'ils sont activés en même temps indiquent une barre verticale. Comme le neurone de la couche 2 s'active quand les trois neurones de la couche 1 sont activés, il détecte les barres verticales. Avec plus de neurones et plus de couches, on peut avoir des neurones qui s'activeront en réaction à des motifs plus complexes (par exemple, un visage).

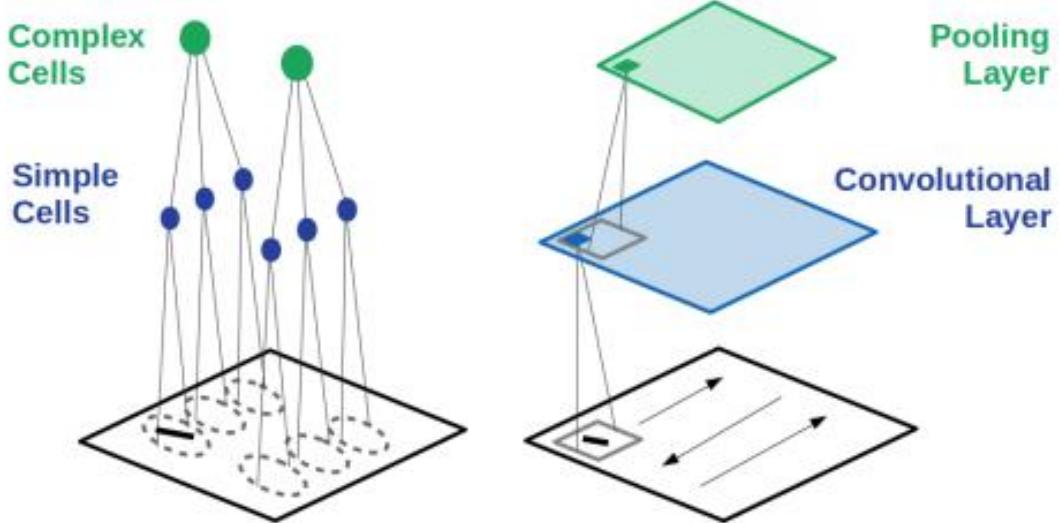


Figure 4: Lien entre le pooling et les couches de convolution chez les CNN, et les cellules simples et complexes dans le cortex visuel : les cellules complexes intègrent les activations des cellules simples, et les couches de pooling intègrent les activations des couches de convolution.

méthodes auto-supervisées (très semblables aux méthodes non-supervisées) pour obtenir de bons modèles du cerveau.

- **AutoEncodeurs Variationnels (VAE)** Les VAE [23] sont des modèles conçus pour répondre à un certain nombre de problèmes dans le cadre de l’inférence bayésienne (voir figure 5 pour plus de détails). Brièvement, l’inférence bayésienne consiste à estimer des distributions de probabilité en intégrant une connaissance *à priori* dans notre estimation. Cela est exprimé mathématiquement par le **Théorème de Bayes** :

$$P(X) = \frac{P(X|Z)*P(Z)}{P(Z|X)}.$$

Où  $P(Z)$  est appelée l’*à-priori* et  $P(Z|X)$  est appelé l’*à-posteriori*.

Pour bien nous figurer les mécanismes des VAE, prenons un exemple concret directement lié à notre sujet. Imaginons que nous ayons à disposition une base d’images  $X$  dont nous tirons des images  $X_i$ . Une image carrée de 255 pixels de côté est lourde et il y a beaucoup de redondances (auto-corrélation spatiale, etc..) ainsi que de compression qui pourrait être réalisée avec une compréhension sémantique de l’image. Nous cherchons donc à extraire une représentation à basse dimension,  $Z$ , de notre base  $X$  à haute dimension. En particulier, pour un  $X$  donné, nous voulons avoir une distribution des  $Z$  possibles, c’est-à-dire  $P(Z|X)$  (l’*à-posteriori*). Analytiquement, en partant du théorème de Bayes,

$$P(X) = \frac{P(X|Z)*P(Z)}{P(Z|X)} \implies P(Z|X) = \frac{P(X|Z)*P(Z)}{P(X)}$$

Or  $P(X)$  n’a pas de solution analytique (certainement dans notre exemple c’est le cas). Grâce aux VAE, nous pouvons estimer l’*à-posteriori* sans avoir besoin de  $P(X)$ . Le VAE possède (voir figure 5 pour un schéma de l’architecture) un encodeur stochastique  $Q_\theta$  ( $\theta$  représente les paramètres du VAE) qui va prendre une image  $X_i$  en entrée et donner une distribution  $Q_\theta(Z_i|X_i)$  en sortie. Le décodeur,  $P_\theta$ , prend  $Z_i$  en entrée et sort  $P_\theta(\hat{X}_i|Z_i)$ , (à noter que c’est nécessaire pour la théorie, mais en pratique nous nous contenterons de dire que le décodeur sort  $\hat{X}_i$  directement),  $\hat{X}_i$  étant son approximation de  $X_i$ .

Un VAE étant un modèle génératif, l’objectif qu’il recherche est de maximiser la probabilité des

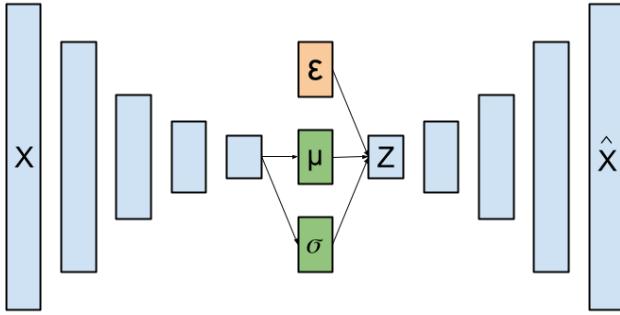


Figure 5: Les AutoEncodateurs Variationnels consistent en un encodeur (couches de convolution) et un décodeur (techniques détaillées plus tard). L'encodeur des VAE est stochastique, et le *re-parametrization trick* consiste à retirer la stochasticité de l'encodeur lui-même, en demandant à l'encodeur de générer une moyenne  $\mu$  et un écart-type  $\sigma$ , puis de tirer un échantillon  $\epsilon$  d'une loi normale centrée réduite, et de le multiplier par  $\sigma$  et lui ajouter  $\mu$ .

images naturelles dans la distribution des images qu'il génère. C'est-à-dire que l'on cherche à maximiser  $P(X)$ , ou  $\ln(P(X))$ , ce qui est équivalent et plus facile. Or,

$$\ln(P(X)) = \ln(\int P_\theta(X|Z)P(Z)dz)$$

(Où  $P(Z)$  est un a-priori que l'on peut choisir sur la forme que doit prendre  $P(Z|X)$  (par exemple, une gaussienne)).

$$\implies \ln(P(X)) = \ln(\int \frac{P_\theta(X|Z)P(Z)}{Q_\theta(Z|X)} * Q_\theta(Z|X)dz)$$

$$\implies \ln(P(X)) = \ln(\mathbf{E}_{z \sim Q_\theta(Z|X)} \frac{P_\theta(X|Z)P(Z)}{Q_\theta(Z|X)})$$

Par l'inégalité de Jensen,

$$\implies \ln(P(X)) \geq \mathbf{E}_{z \sim Q_\theta(Z|X)} \ln(\frac{P_\theta(X|Z)P(Z)}{Q_\theta(Z|X)})$$

$$\implies \ln(P(X)) \geq \mathbf{E}_{z \sim Q_\theta(Z|X)} [\ln(P_\theta(X|Z)) + \ln(P(Z)) - \ln(Q_\theta(Z|X))]$$

Finalement,

$$\ln(P(X)) \geq \mathbf{E}_{z \sim Q_\theta(Z|X)} [\ln(P_\theta(X|Z))] - \mathbf{E}_{z \sim Q_\theta(Z|X)} [\ln(Q_\theta(Z|X)) - \ln(P(Z))] \quad (1)$$

L'expression à droite de l'équation (1) est l'objectif de maximisation du VAE, appelé *Evidence Lower Bound*(ELBO). Puisque  $\ln(P(X))$  y est supérieur, maximiser l'ELBO implique maximiser  $\ln(P(X))$  et donc aussi  $P(X)$ .

Si nous parvenons à bien entraîner notre VAE pour qu'il maximise l'ELBO, alors l'encodeur nous donnera nos valeurs d'*à-posteriori*  $Q_\theta(Z|X)$ .

L'ELBO est constitué de deux termes :

- $\mathbf{E}_{z \sim Q_\theta(Z|X)} [\ln(P_\theta(X|Z))]$  représente la probabilité des images  $X$  dans les distributions générées par le décodeur. Le modèle est donc bien entraîné pour maximiser la probabilité des images naturelles dans les distributions qu'il génère.

- $\mathbf{E}_{z \sim Q_\theta(Z|X)}[\ln(Q_\theta(Z|X)) - \ln(P(Z))]$  est la distance de Kullback-Leibler entre les distributions générées par l'encodeur, et l'*à-priori* que nous avons choisi. Ce terme n'implique que l'encodeur, et le pénalise s'il ignore l'*à-priori* que nous avons fixé sur la forme que devraient prendre ses distributions.

Il est courant d'utiliser une version modifiée de cette fonction de coût car l'ELBO tel quel a tendance à ne pas donner les meilleures performances. Dans notre cas, nous remplaçons le premier terme par la distance perceptuelle LPIPS [59] entre l'image en entrée et l'image en sortie. De plus, nous choisissons comme *à-priori* une gaussienne centrée réduite.

#### Ce qu'il faut en retenir :

Ainsi les VAE apprennent de manière non-supervisée une représentation latente de l'entrée, qui peut ensuite être inversée pour retrouver l'entrée. Dans notre cas, nous avons simplement affaire à un modèle qui, pour une image en entrée, va nous donner une distribution sur l'espace latent. Si nous échantillonons une valeur sur cette distribution pour ensuite la décoder avec le décodeur, nous obtenons une autre image qui doit être proche de l'image en entrée.

- **Utilité des VAE dans notre cas :** L'utilité des VAE dans notre cas est multiple : premièrement, il est à noter que la partie "encodeur" du VAE n'est rien d'autre qu'un CNN, et en conséquence un bon modèle du cerveau dont on peut extraire des métriques d'*économie* du traitement de l'information. De plus, la possibilité de reconstruire les images à partir de la représentation latente permet une métrique de l'*efficacité* du traitement de l'information. Finalement, l'apprentissage non-supervisé semble une méthode d'entraînement plus réaliste, car pour les humains dans le monde réel l'accès à des "vérités de terrain" est rare mais l'apprentissage a constamment lieu, indiquant de l'apprentissage non-supervisé.
- **Utilité du reparametrization trick** (Le reparametrization trick est expliqué à la figure 5) Nous justifions ici analytiquement l'utilisation du re-parametrization trick.  
Pour ancrer notre justification, expliquons comment est en général entraîné un modèle type CNN : par un algorithme de **desccente de gradient** qui se décrit ainsi :

1. Passer une image dans le modèle pour en obtenir les outputs.
2. Pour ces outputs, calculer la fonction de coût (ELBO dans notre cas).
3. Par rétropropagation (application récursive de la règle  $\frac{\delta f(g(x))}{\delta x} = \frac{\delta f(g(x))}{\delta g(x)} * \frac{\delta g(x)}{\delta x}$ ), trouver le gradient  $\nabla_w$  de la fonction de coût par rapport à chacun des paramètres  $w$  du modèle.
4. Changer chacun des poids  $w$  avec la règle  $w \leftarrow w - \nabla_w * \lambda$  où  $\lambda$  est un paramètre que nous choisissons, appelé le *learning rate*.
5. Recommencer les étapes 1 à 4 jusqu'à convergence de la fonction de coût.

Le reparametrization trick intervient lors de l'étape 3.

Supposons donc que nous essayons d'approximer le gradient de l'ELBO sans le reparametrization trick :

$$\nabla_\theta \mathbf{E}_{Q_\theta(Z|X)}[\log(p_\theta(x|z)) - \log(\frac{Q_\theta(Z|X)}{p(Z)})]$$

C'est égal à

$$\nabla_\theta \int[(\log(p_\theta(x|z)) - \log(\frac{Q_\theta(Z|X)}{p(Z)})) * Q_\theta(Z|X)] dz$$

En général, les conditions sont remplies pour passer le gradient dans l'intégrale. De plus, le gradient étant distributif sur la multiplication ("product rule") :

$$= \int (\nabla_\theta [\log(p_\theta(x|z)) - \log(\frac{Q_\theta(Z|X)}{p(Z)})] * Q_\theta(Z|X)) dz + \int ((\log(p_\theta(x|z)) - \log(\frac{Q_\theta(Z|X)}{p(Z)})) * \nabla_\theta [Q_\theta(Z|X)]) dz \quad (2)$$

Nous nous retrouvons avec deux intégrales dans l'équation (2).

L'idée dans la pratique, sera d'approcher la valeur des intégrales par échantillonnage de plusieurs valeurs (approche *Monte Carlo*). Le problème ici réside dans la seconde intégrale, qui n'est pas

sous la forme d'une espérance selon  $Q_\theta(Z|X)$  : cela ne nous permet pas de l'estimer stochastiquement en Monte-Carlo. Nous pouvons le faire néanmoins, car rien n'empêche de l'implémenter, mais les estimations du gradient auront une grande variance.

Au contraire, regardons ce qui se passe si l'on utilise le re-parametrization trick :

Avec le re-parametrization trick, au lieu de générer directement  $Q_\theta(Z|X)$  avec l'encodeur, nous générerons les vecteurs  $\mu_\theta$  et  $\sigma_\theta$ . Pour échantillonner  $z$ , nous échantillonnerons une valeur  $\epsilon \sim \mathcal{N}(0, Id)$  et calculons  $z = \mu_\theta + \sigma_\theta * \epsilon$ . Ce procédé est équivalent à l'échantillonnage d'une loi  $\mathcal{N}(\mu_\theta, diag(\sigma_\theta))$ . Néanmoins, si nous calculons à présent le gradient de l'ELBO (comme pour l'équation (2)) :

$$\begin{aligned}\nabla_\theta \mathbf{E}_{Q_\theta(Z|X)}[\log(p_\theta(x|z)) - \log(\frac{Q_\theta(Z|X)}{p(Z)})] \text{ devient} \\ \nabla_\theta \mathbf{E}_{p(\epsilon)}[\log(p_\theta(x|\mu_\theta + \sigma_\theta * \epsilon)) - \log(\frac{q_\theta(\mu_\theta + \sigma_\theta * \epsilon|x)}{p(\mu_\theta + \sigma_\theta * \epsilon)})]\end{aligned}$$

Gardons en mémoire que  $z = \mu_\theta + \sigma_\theta * \epsilon$ . Nous pouvons écrire :

$$\begin{aligned}\nabla_\theta \mathbf{E}_{p(\epsilon)}[\log(p_\theta(x|\mu_\theta + \sigma_\theta * \epsilon)) - \log(\frac{q_\theta(\mu_\theta + \sigma_\theta * \epsilon|x)}{p(\mu_\theta + \sigma_\theta * \epsilon)})] \\ = \nabla_\theta \int [(\log(p_\theta(x|z)) - \log(\frac{Q_\theta(Z|X)}{p(Z)})) * p(\epsilon)] d\epsilon \\ = \int \nabla_\theta [p(\epsilon) \log(p_\theta(x|z))] dz - \int \nabla_\theta [p(\epsilon) \log(\frac{q_\theta(Z)}{p(Z)})] d\epsilon\end{aligned}$$

Or, cette fois-ci, comme nous faisons l'espérance sur  $\epsilon$  qui ne dépend pas de  $\theta$ , nous pouvons ré-écrire cette expression comme suit :

$$\nabla_\theta \mathbf{E}_{p(\epsilon)}[\log(p_\theta(x|\mu_\theta + \sigma_\theta * \epsilon)) - \log(\frac{q_\theta(\mu_\theta + \sigma_\theta * \epsilon|x)}{p(\mu_\theta + \sigma_\theta * \epsilon)})] = \mathbf{E}_{p(\epsilon)} \nabla_\theta [\log(p_\theta(x|z)) - \mathbf{E}_{p(\epsilon)} \nabla_\theta [\log(\frac{q_\theta(Z)}{p(Z)})]] \quad (3)$$

Nous voyons à l'équation (3) que grâce au re-parametrization trick, nous n'avons que des espérances, et en pratique la variance des estimations par *Monte-Carlo* du gradient sera plus basse, et l'entraînement du modèle sera donc plus stable.

### 3.3.4 Assemblage des concepts introduits

Maintenant que nous avons introduit tous les concepts, nous pouvons expliquer l'objectif du stage : Il s'agit d'entraîner des VAE sur des images de visages, et d'en extraire des métriques d'*économie* et d'*efficacité* du traitement de l'information, afin de voir si elles corrèlent et permettent de prédire la beauté.

Un objectif supplémentaire est de fournir une interface permettant de ré-entraîner les modèles et d'obtenir les métriques sur n'importe quelle nouvelle base de données. Ainsi, le modèle mathématique pourrait être testé sur les animaux.

## 4 Environnement technique

### 4.1 Cluster de calcul Meso-LR

Le mésocentre Meso@LR est un centre spécialisé offrant des ressources partagées et des infrastructures de pointe pour le calcul intensif (High Performance Computing - HPC) et le traitement massif des données.

Grâce à leurs CPUs avec beaucoup de RAM (3To), nous avons pu finir lancer le code du stage de l'année dernière où des ACPs sur les activations de VGG16 prenaient beaucoup de place en mémoire.

### 4.2 Supercalculateur Jean Zay

Jean Zay représente le dernier supercalculateur convergé récemment acquis par le ministère de l'Enseignement supérieur, de la Recherche et de l'Innovation grâce à la collaboration de la société GENCI (Grand équipement national de calcul intensif).

Ces travaux ont bénéficié d'un accès aux moyens de calcul de l'IDRIS au travers de l'allocation de ressources attribuée par GENCI au compte uzk68zl.

Grâce aux GPUs V100 disponibles sur Jean Zay, nous avons pu faire tourner nos modèles 4 fois plus vite qu'avec notre GPU en local, avec l'avantage supplémentaire de pouvoir lancer plusieurs entraînements en même temps. Nous y avons eu accès assez tard dans le stage, si bien que nous n'en avons bénéficié que pour l'entraînement des modèles avec les mécanismes d'attention et les modèles spécifiques aux ethnies et aux genres.

### 4.3 Langage, librairies, Pages GitHub

- Toute implémentation d'architectures de Deep Learning demande d'utiliser une bibliothèque d'auto-différentiation. Notre choix a été Pytorch [38].
- Nous avons choisi le langage Python car il est à un niveau d'abstraction adapté à nos objectifs dans le cadre du stage (pas si simple qu'on ne peut pas aller dans le détail, mais assez abstrait pour ne pas perdre de temps).
- Nous avons à diverses utilisés du code provenant de pages GitHub. En voici la liste (tous les termes sont définis plus tard dans le rapport) :

Pour la distance perceptuelle LPIPS : [lien GitHub](#)

Pour le modèle pré-entraîné AttGAN : [lien GitHub](#)

Pour une implémentation différentiable de l'index de similarité SSIM : [lien GitHub](#)

## 5 Méthode générale

### 5.1 Bases de données

Dans ce stage, nous avons utilisé deux bases de données de visages.

- La base de données CFD [30] est constituée d'environ 800 images standardisées de visages, répartis par ethnie et par sexe. Pour chaque visage, un score de beauté a été attribué par un jury.
- La base de données Fairface est constituée d'une centaine de miliers d'images de visages non-standardisées collectées sur le Web, classées dans des catégories d'ethnie et de sexe de taille à peu près égale. [22]

L'utilisation exclusive d'images de visages est questionnable car on obtiendrait un meilleur modèle du cortex visuel en entraînant notre modèle sur une base de données représentative des images naturelles en général.

### 5.2 Modélisation de la fluence de traitement neuronal (métriques)

#### 5.2.1 Mesures de sparsité

Une manière intuitive de mesurer la simplicité du traitement de l'information dans le cerveau est la sparsité des activations neuronales. La sparsité indique à quel point les activations sont localisées sur un nombre restreint de neurones. Typiquement, si quelques neurones s'activent fortement et tous les autres restent à zéro, les activations sont sparses. Au contraire, si tous les neurones sont à moitié activés, les activations ne sont pas sparses.

Comme mesure de sparsité, nous avons utilisé les deux métriques suivantes :

1. L'index de gini : une mesure de l'inégalité des richesses dans une population. Un index de gini de 1 correspond à l'inégalité absolue, et un index de gini de 0 correspond à l'égalité absolue. (voir figure 6).
2. La norme  $L_\epsilon$  : elle est calculée en comptant le nombre de neurones qui sont activés au delà d'un seuil  $\epsilon$ . Si beaucoup de neurones se sont activés au delà de ce seuil, on estimera que le motif d'activation neuronale n'était pas sparse.

Dans [19], les auteurs proposent un certains nombre de critères que doit respecter une mesure de sparsité. Ils comparent un certain nombre de mesures différentes, pour en conclure que la seule qui remplit tous leurs critères est *l'index de Gini*.

#### 5.2.2 Qualité de reconstruction

Nous avons utilisé la qualité de reconstruction comme mesure de l'efficacité du traitement neuronal : plus le VAE reconstruit bien une image, plus il est jugé que l'encodeur (le modèle du cortex visuel) en a bien extrait l'information nécessaire. Comme mesure de la qualité de la reconstruction, nous utilisons la distance perceptuelle LPIPS [59] entre l'image d'entrée, et l'image en sortie du VAE. Plus la distance est petite, plus il est estimé que l'encodeur a bien compris l'image en entrée.

#### 5.2.3 Sharpness Aware Minimization

##### Intuition :

La *Sharpness Aware Minimization* (SAM) [7] est une version modifiée de la descente de gradient où l'on cherche non seulement à minimiser la fonction de coût, mais aussi à se diriger vers un minimum "plat", c'est-à-dire que les valeurs de la fonction de coût dans le voisinage du minimum restent basses (voir figure 8 pour une explication visuelle). L'intuition est qu'un minimum local "pointu" (en opposition à "plat") indique que le moindre changement des paramètres du modèle causerait un effondrement de ses connaissances. En évitant cette fragilité du modèle, on garantirait notamment une meilleure capacité de généralisation.

##### Fonctionnement :

Au lieu de chercher à minimiser directement la fonction de coût  $\mathcal{L}(W)$  ( $W$  les paramètres du modèle), on cherche à minimiser une version modifiée de la fonction de coût :

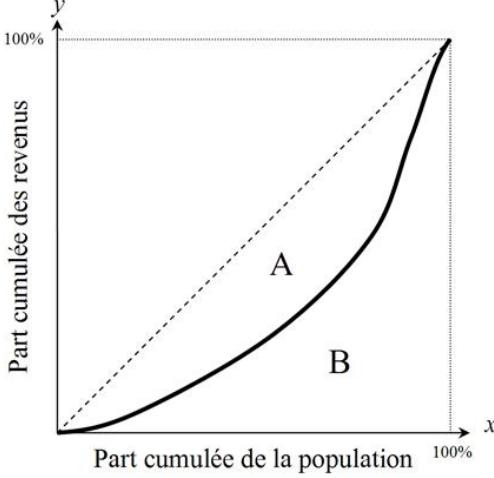


Figure 6: L’index de gini provient de l’économie, où il est utilisé pour mesurer l’inégalité dans une distribution. Il est égal à l’aire A sur le dessin.

$$\mathcal{L}_{\text{SAM}}(W) = \max_{\|\epsilon\| < \rho} (\mathcal{L}(W + \epsilon))$$

En mots, cette expression signifie que l’on minimise le maximum de la fonction de coût dans une zone autour de  $W$  (la taille de la zone étant donnée par  $\rho$ , un paramètre que nous choisissons).

Néanmoins, un problème se pose : l’opérateur  $\max$  utilisé n’est pas dérivable, or pour entraîner notre modèle, nous avons besoin de dériver la fonction de coût. Voyons comment nous pouvons malgré tout approximer la valeur suivante :

$$\nabla_W [\max_{\|\epsilon\| < \rho} (\mathcal{L}(W + \epsilon))]$$

L’astuce est la suivante : si on peut trouver le  $\hat{\epsilon}$  qui maximise la fonction de coût dans la zone définie par  $\rho$ , on peut simplement remplacer l’expression par :

$$\nabla_W [\max_{\|\epsilon\| < \rho} (\mathcal{L}(W + \epsilon))] = \nabla_W [\mathcal{L}(W + \hat{\epsilon})]$$

Nous pouvons ensuite suivre le raisonnement suivant :

1. Le gradient d’une fonction indique la *direction locale d’augmentation maximale de la valeur de la fonction*.
2. Dans la mesure où  $\rho$  est petit (et c’est le cas pour nous), notre recherche de  $\hat{\epsilon}$  peut être considérée comme locale.
3. Conclusion : nous pouvons approximer  $\hat{\epsilon}$  par :

$$\hat{\epsilon} \approx \rho * \frac{\nabla_W [\mathcal{L}(W)]}{\|\nabla_W [\mathcal{L}(W)]\|} \quad (4)$$

Et l’algorithme de descente de gradient modifiée devient alors :

1. Passer une image dans le modèle pour en obtenir les outputs.
2. Pour ces outputs, calculer la fonction de coût  $[\mathcal{L}(W)]$ .
3. Par rétropropagation, trouver le gradient  $\nabla_W [\mathcal{L}(W)]$  de la fonction de coût par rapport à chacun des paramètres  $w$  du modèle.
4. Avec ce gradient, calculer  $\hat{\epsilon}$  selon l’équation (4).
5. Ajouter  $\hat{\epsilon}$  à  $W$  et calculer la fonction de coût  $\mathcal{L}(W + \hat{\epsilon})$  avec ces nouveaux paramètres.
6. Par rétropropagation, trouver le gradient  $\nabla_W [\mathcal{L}(W + \hat{\epsilon})]$ .

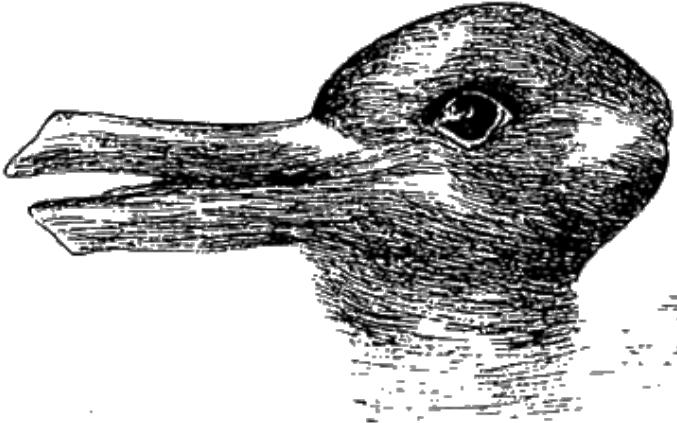


Figure 7: Démonstration de l’importance de l’attention visuelle : si l’on fait attention au pointu des formes à gauche, on voit un canard, mais si l’on fait attention à la protubérance du crâne du canard à droite, on voit soudainement un lapin. Additionnellement, au moment où l’on finit par comprendre l’image, on ressent du plaisir, ce qui est expliqué par la théorie de la fluence comme le plaisir du gain d’information.

7. Retirer  $\hat{\epsilon}$  des paramètres du modèle pour revenir à  $W$ .
8. Changer chacun des poids  $w$  avec la règle  $w \leftarrow w - \nabla_w [\mathcal{L}(W + \hat{\epsilon})] * \lambda$ .
9. Recommencer les étapes 1 à 8 jusqu’à convergence de la fonction de coût.

Une représentation visuelle de cet algorithme est donnée à la figure 8.

#### Utilisation de SAM dans notre cas :

Nous avons tout d’abord testé l’apport de SAM en capacité de généralisation des modèles (voir figure 9 pour les détails sur le déroulement de l’entraînement). Pour deux modèles entraînés sur Fairface, l’un avec SAM et l’autre sans, il n’y avait pas de différence en termes de la qualité de reconstruction sur CFD. Donc dans le cas de nos VAE, SAM n’apportait rien à l’entraînement.

Néanmoins SAM ne nous intéressait pas comme moyen d’améliorer nos modèles, mais comme inspiration pour une mesure de la compréhension des images par le modèle. Si  $\mathcal{L}(W + \hat{\epsilon})$  est le maximum local de la fonction de coût, et  $\mathcal{L}(W)$  en est le minimum local, alors la valeur  $\mathcal{L}(W + \hat{\epsilon}) - \mathcal{L}(W)$  mesure l’amplitude de changement local de la fonction de coût – en d’autres termes, son acuité (*sharpness*) locale.

Si, de plus, l’acuité locale est une mesure de la robustesse du modèle au point  $W$  et pour une image en entrée, nous pouvons utiliser cette mesure d’acuité locale comme une mesure d’*à quel point le modèle a bien compris une image en entrée*. Nous pourrions donc utiliser cela comme une mesure de l’efficacité du traitement neuronal et donc de la fluence, qui pourrait être un bon prédicteur de la beauté.

#### 5.2.4 Attention

**Attention en rapport à l’esthétique** Nous nous sommes intéressés à l’effet de l’attention sur l’expérience de la beauté. Un exemple parlant en terme d’attention visuelle en relation à la théorie de la Fluence est celui des illusions d’optique (voir figure 7). Selon Shaeffer [47], l’attention esthétique est qualifiée par trois caractéristiques qui la séparent de l’attention “normale” :

- La *Densification attentionnelle* : pour une dimension donnée du stimulus (par exemple la couleur), un sujet en état d’attention esthétique percevra des différences plus fines sur cette

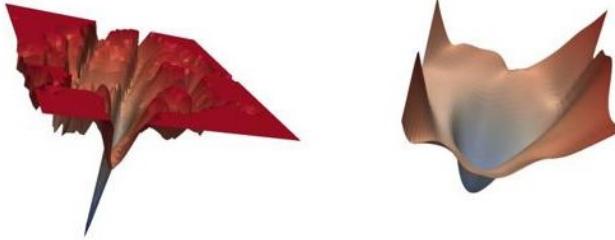


Figure 8: Si l'on représente l'erreur de reconstruction comme une heightmap à deux dimensions, qui sont deux paramètres du modèle, une courbe plus lisse (à droite) indique une plus grande robustesse de la reconstruction à un changement des paramètres, ce qui indique une meilleure compréhension de l'image, et ainsi une plus haute "efficacité" du traitement de l'information.

dimension, c'est-à-dire que le nombre de valeurs (couleurs) qu'il parviendra à différencier sera plus grand.

- La *Saturation attentionnelle* : le sujet portera attention à plus de dimensions du stimulus (couleur, contour, présence physique, texture...) à la fois qu'il ne l'aurait fait autrement.
- Le *contournement des schèmes perceptuels* : les schèmes perceptuels sont des motifs abstraits selon lesquels nous catégorisons les objets que nous percevons. Avec l'aide des schèmes perceptuels, nous pouvons obtenir toute l'information nécessaire sur un stimulus en un regard (par exemple, pour voir une chaussure il nous suffit d'en voir la forme générale, et à moins que la situation le demande, nous n'allons pas chercher plus loin). Dans le cadre de l'attention esthétique, les schèmes perceptuels seraient outrepassés, l'attention se portant sur l'expérience de perception plutôt que sur la catégorisation de l'objet.

Ces caractéristiques de l'interaction esthétique avec un objet impliquent l'abandon des racourcis et l'examen minutieux. Elles semblent donc contraires à l'idée que les stimuli les plus beaux seraient les plus facilement traités (Fluence). [11] propose un modèle de l'esthétique qui prend en compte cette dimension supplémentaire de l'esthétique associée à l'effort et l'apprentissage.

### 5.3 Développement d'un Autoencodeur Variationnel

Il existe dans l'état de l'art des VAE qui ont de très bonnes reconstructions [3] [31]. Par exemple, on peut voir des reconstructions du modèle nVAE[53] à la figure 10. Cependant, ceux-là sont des modèles hiérarchiques avec des connexions résiduelles et de l'échantillonnage à différents niveaux, et nous avons des doutes sur leur utilité comme modèles du cerveau. L'objectif était donc ici de créer un VAE qui

- Possède un encodeur semblable aux CNN qui ont été testés comme modèles du cerveau (AlexNet [25], VGG16 [49]...)
- Donne des reconstructions de qualité.

Il n'y a pas à notre connaissance de modèle dans l'état de l'art qui réunisse ces deux qualités.

Nous avons commencé par implémenter et entraîner sur Fairface un Autoencodeur Variationnel (VAE) avec l'architecture donnée dans l'article d'origine [23]. Pour lier la qualité de reconstruction à la *compréhension* de l'image par l'encodeur, il faut que les reconstructions soient bonnes. Or, ce n'était pas le cas avec l'architecture de base.

Nous avons testé beaucoup de techniques pour améliorer la qualité des reconstructions. Nous listons ici toutes ces techniques avec une brève explication et les figures montrant leur effet positif ou négatif.

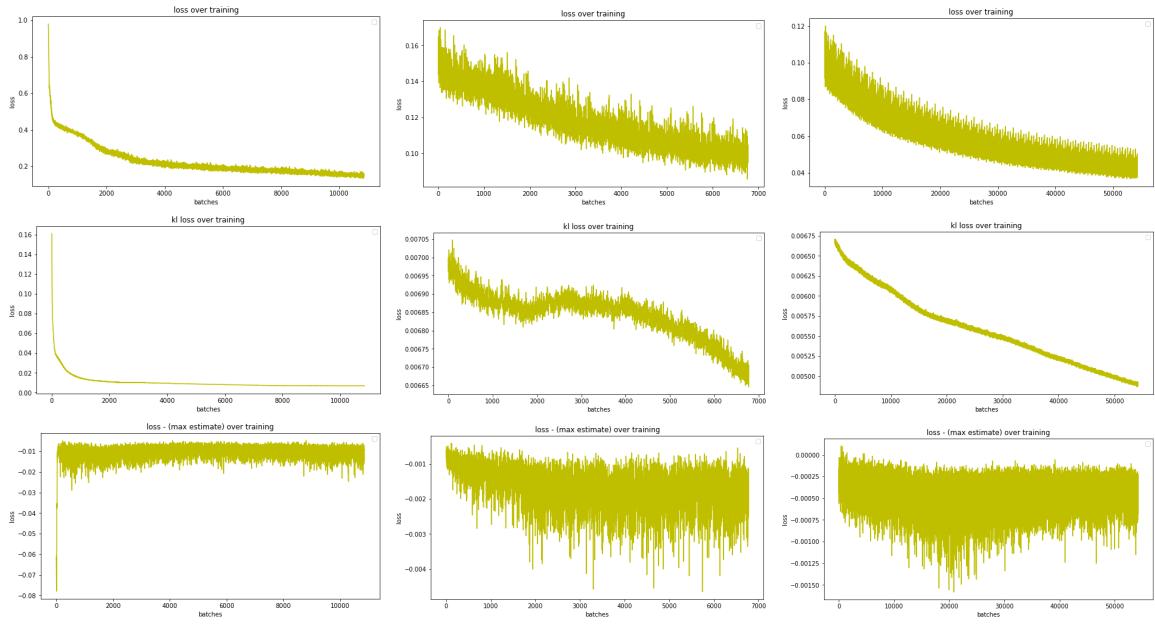


Figure 9: Fonctions de coût pendant l’entraînement du VAE avec Sharpness Aware Minimization – colonnes : Trois phases d’entraînement (le modèle a été entraîné en trois fois) – ligne 1 : erreur de reconstruction ; ligne 2 : valeur de la divergence de Kullback-Leibler de la contrainte de sparsité ; ligne 3 : la valeur  $L(\epsilon) - L(\hat{\epsilon})$ . Nous voyons qu’elle reste négative, ce qui indique que le gradient est toujours une bonne estimation de la direction d’augmentation maximale d’augmentation de la fonction de coût.



Figure 1: 256×256-pixel samples generated by NVAE, trained on CelebA HQ [28].

Figure 10: Reconstructions données par le modèle nVAE. elles sont très bonnes, mais l’architecture est éloignée des modèles du cerveau.

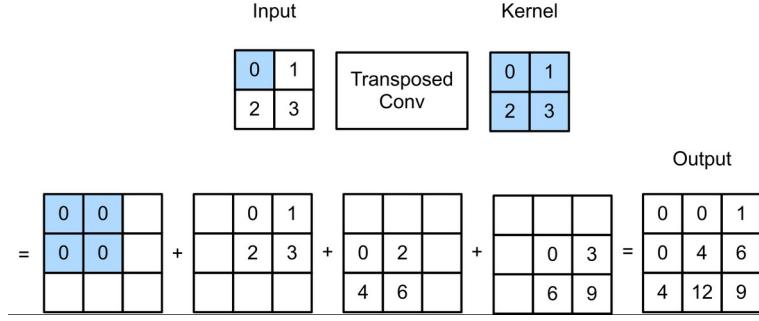


Figure 11: Fonctionnement de ConvTranspose2d : un kernel est multiplié par la valeur d'un pixel à la fois et à chaque fois, le résultat est ajouté dans l'image de sortie aux pixels autour de la position du pixel en question.

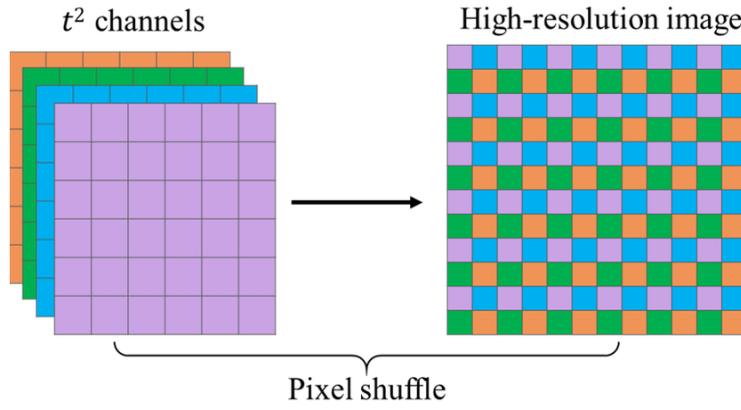


Figure 12: Le principe de PixelShuffle est de réaliser des convolutions avec de nombreux filtres pour avoir beaucoup de featuremaps différentes. Ensuite, les featuremaps sont combinées pour former une plus grande image, comme montré sur la figure.

### 5.3.1 Méthodes d'upsampling :

voir figure 16 pour les résultats

*Convtranspose* : C'est la méthode la plus répandue pour l'upsampling dans les méthodes de reconstruction ou génération d'images. Elle ressemble à une convolution, mais fonctionne différemment (voir figure 11)

*Upsample traditionnel suivi d'une convolution* : La ConvTranspose est connue pour causer des motifs indésirables de damier sur les images à la sortie. Une solution proposée [34] est de pratiquer un upsampling traditionnel sans paramètres (par exemple par interpolation linéaire), suivi d'une convolution classique qui permet d'avoir des paramètres dans l'opération.

*Pixelshuffle* : [48] est une autre méthode alternative à la ConvTranspose. Elle consiste à appliquer un grand nombre de filtres de convolution pour obtenir des cartes de caractéristiques. Ensuite, ces filtres sont combinés entre eux pour former une plus grande image (voir figure 12 pour une explication visuelle).

### 5.3.2 Différences dans l'espace latent :

*Espace latent par fully connected ou par convolution* : Dans l'architecture de l'article d'origine, les vecteurs de l'espace latent sont obtenus par une couche fully-connected (une multiplication de matrice dont tous les paramètres sont appris pendant l'entraînement). Nous avons essayé cette alternative, ainsi que la possibilité d'obtenir ces vecteurs par une couche de convolution (ce qui permet de prendre

en compte l'information spatiale tout en réduisant le nombre de paramètres).

*Taille de l'espace latent* : Le choix de la taille de l'espace latent est important : trop petit, il contraint trop le modèle et les reconstructions sont mauvaises. Trop grand, il rend la tâche plus facile au modèle, qui n'est plus forcée d'apprendre bien (voir en addition figure 21).

### 5.3.3 Fonctions de coût :

voir figure 17 pour les résultats

*Mesures simples de distance (normes L1 et L2)* : Les normes L1 (somme des valeurs absolues des différences entre les pixels) et L2 (somme des carrés des différences entre les pixels) sont les plus utilisées à l'entraînement d'Autoencodeurs.

*SSIM* : L'index de similarité structurelle SSIM[55], est une mesure très répandue de la similarité entre deux images. Il est basé sur l'extraction de caractéristiques par sous-parties des deux images et comparaison de ces caractéristiques.

*Fonction de coût du Deep feature consistent VAE (DFCVAE)[16]* : L'idée de cette fonction de coût est d'imiter le jugement de similarité humain avec un réseau de neurones qui sert de modèle du cortex visuel. Pour ce faire, on passe les deux images à comparer dans le réseau pour obtenir les représentations extraites, et on calcule alors une norme L2 entre les deux représentations. Nous avons essayé cette technique avec différents réseaux comme modèles du système visuel : attGAN [13], VGGFace[49], et un VAE basique.

*LPIPS* : la distance LPIPS [59] (Learned perceptual image patch similarity) repose sur la même idée la fonction de coût du DFCVAE, avec toutefois quelques ajouts, entre autres une multiplication des représentations sur les différentes couches par des scalaires pour mieux se rapprocher du jugement humain.

### 5.3.4 Méthodes de régularisation :

*Batch Normalisation* : La Batch Normalisation [21] consiste à normaliser l'output de chaque couche du modèle en z-transform), pour ensuite la multiplier par une moyenne et un écart-type appris. Cela permet entre autres de pailler au problème du *vanishing gradient* ainsi qu'à réduire l'interdépendance entre les couches. L'*Instance Normalisation* [52] fait la même chose, mais avec une moyenne et un écart-type par image et par couche. Nous ne montrons pas les résultats ici car leur utilisation n'y change rien.

### 5.3.5 Connections résiduelles :

L'idée principale des connections résiduelles [12] dans le cadre du Deep Learning est d'ajouter l'entrée d'une couche à sa sortie. Si l'entrée est  $x$ , et la couche non-residuelle donne la sortie  $f(x)$ , la couche résiduelle donnera  $f(x) + x$ . L'intuition primaire derrière ce mécanisme est : les couches de convolution ont tendance à apprendre la fonction  $f(x) = 0$  plus facilement que la fonction  $f(x) = x$ . L'ajout de  $x$  dans la sortie de la couche permet au réseau de se concentrer sur les petites différences plutôt qu'à conserver l'information. Nous n'avons pas utilisé cette technique dans l'encodeur pour le garder aussi semblable aux modèles type VGG que possible.

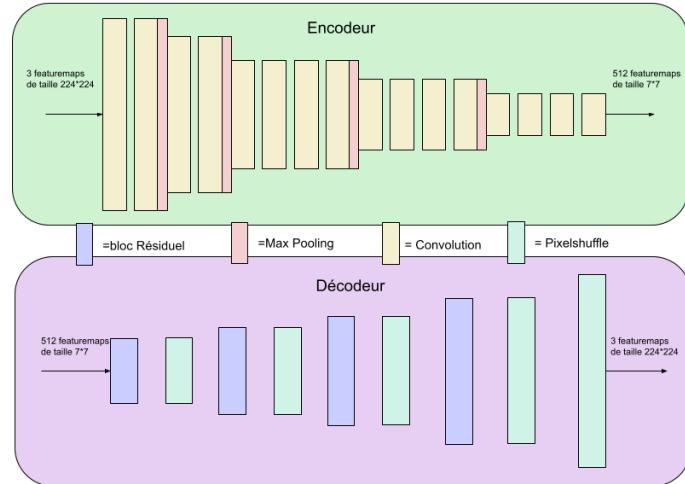
### 5.3.6 Différents modèles

Une fois que nous avions choisi quelles techniques garder parmi toutes celles qui figurent au dessus, nous avons expérimenté avec différentes tailles pour l'encodeur et le décodeur. Nous avons testé plus d'architectures qu'il est pertinent de détailler ici ; trois exemples sont fournis à la figure 13.

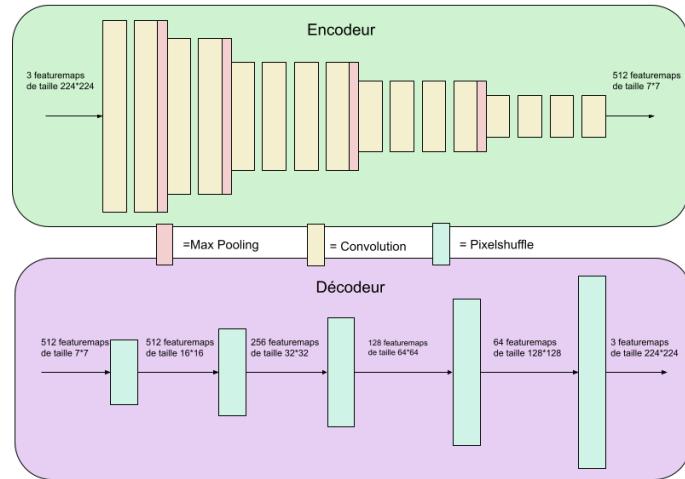
### 5.3.7 Étude des représentations

Les VAE sont capables d'extraire, de manière non supervisée, de la structure entre les différentes images qu'on leur donne en entrée. Ainsi on peut interpoler dans l'espace latent entre les images, et obtenir dans l'espace de sortie une interpolation non-linéaire pertinente (voir figure 18). De plus, grâce à cette structure, on pourrait s'attendre à ce que dans l'espace latent, les visages d'une certaine ethnique

### Convolutions de VGG16 avec décodeur à 10 couches



### Convolutions de VGG16 avec décodeur à 5 couches



### Encodeur à 5 couches et décodeur à 5 couches

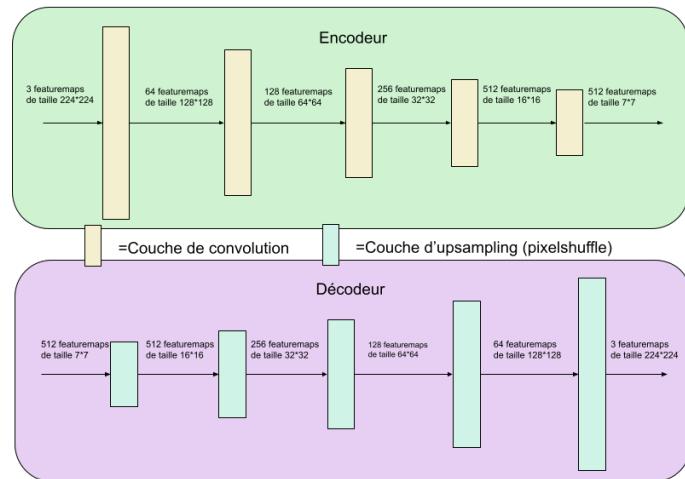


Figure 13: schémas représentant les 3 architectures principales testées

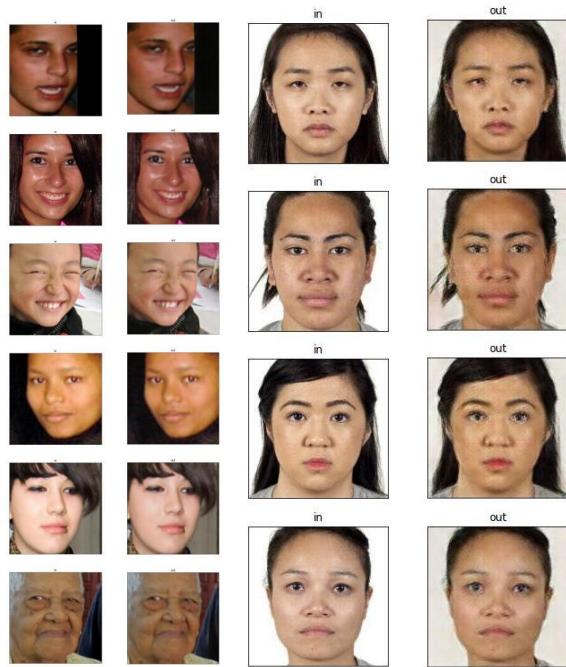


Figure 14: Reconstructions du petit modèle (5 couches à l'encodeur et au décodeur) sur Fairface à gauche, et CFD à droite. On montre à chaque fois l'image originale suivie de la reconstruction.

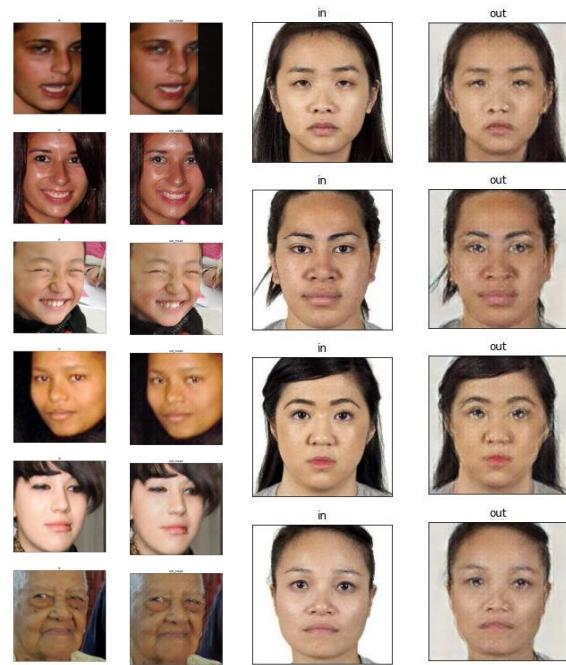


Figure 15: Reconstructions du modèle avec VGG en encodeur et 5 couches au décodeur sur Fairface à gauche, et CFD à droite.

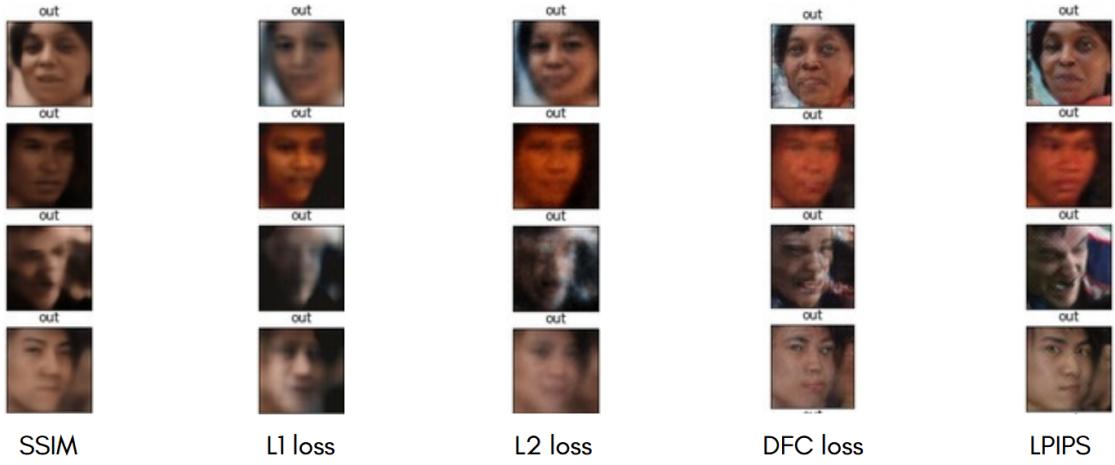


Figure 16: Effet de différentes méthodes d’upsampling sur les reconstructions du petit modèle. LPIPS semblait être la meilleure pour nos trois architectures.

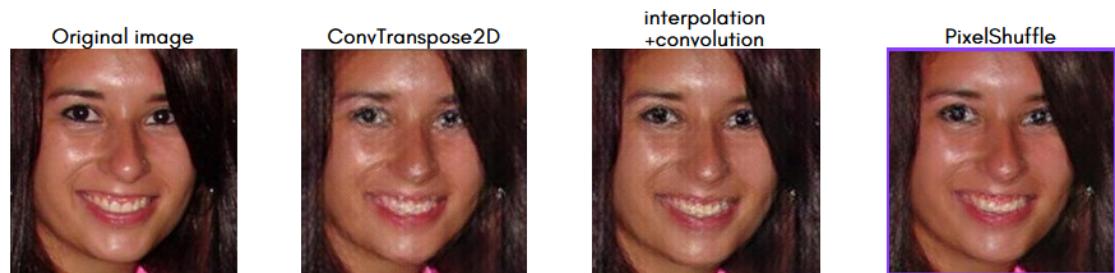


Figure 17: Effet de différentes fonctions de coût à l’entraînement sur les reconstructions du petit modèle. Nous avons choisi d’utiliser PixelShuffle au final.



Figure 18: Interpolation linéaire dans l'espace latent entre les représentations des images de Fairface avec le petit modèle. On décode chaque point de l'interpolation pour obtenir les transitions ci-dessus. Il semblerait que le modèle tente toujours de représenter un visage, ce qui montre une certaine robustesse du modèle (même s'il existe de meilleurs résultats dans l'état de l'art).

se trouvent dans la même zone, ceux d'une autre se trouvant un peu à côté, etc. Nous avons donc obtenu la figure 19 (à gauche) pour tester cela, et voir si de la structure n'était pas extraite en relation aux scores de beauté (les visages beaux se trouvant plus dans une certaine zone que dans une autre). Aucune structure n'apparaît dans l'espace latent, mais avec un espace latent quatre fois plus petit, on obtient de la séparation entre les ethnies et sexes (voir figure 19 (à gauche)). Toutefois, aucune structure en lien avec la beauté n'émerge.

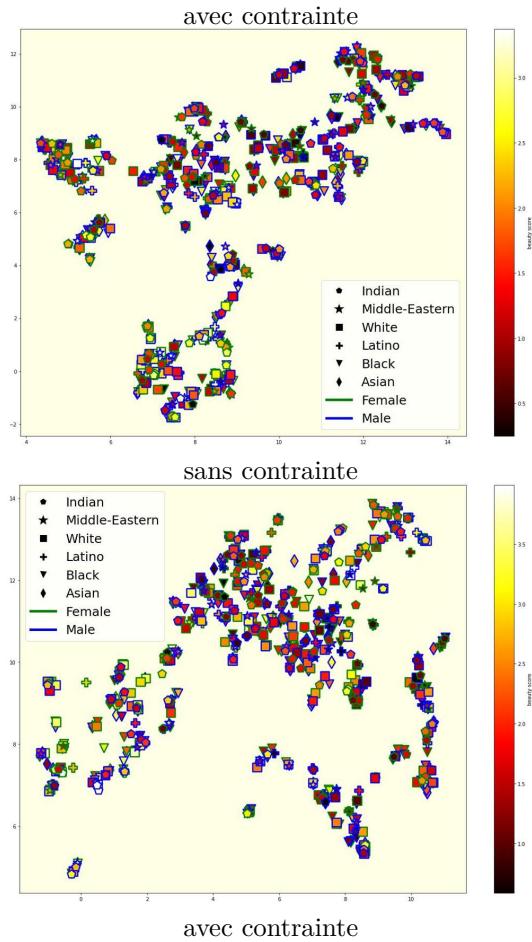


Figure 19: Structure extraite de manière non-supervisée par le VAE avec VGG en encodeur, et 10 couches au décodeur. Le graphe est obtenu par projection UMAP sur 2 dimensions des activations dans l'espace latent en réponse à chaque image de CFD. Nous comparons avec et sans la contrainte – il n'y a rien d'extraït dans les deux cas.

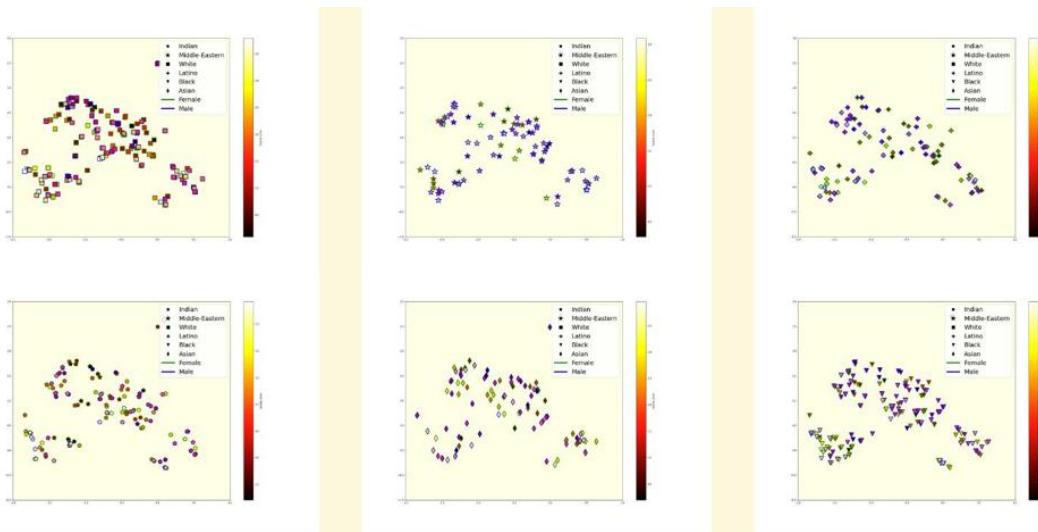


Figure 20: Avec le même modèle et la même technique (sans la contrainte), nous regardons aussi par ethnies si la structure est extraite. Ce n'est pas le cas.

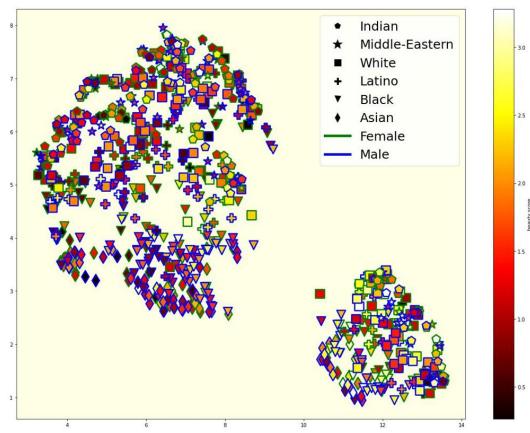


Figure 21: Avec un espace latent 4 fois plus petit, nous voyons cette fois-ci une structure apparaître.

## 6 Tâches

### 6.1 Reprise du stage de l'année précédente

- Le stagiaire de l'année dernière, Melvin BARDIN, a travaillé sur la modélisation de la Fluence comme vraisemblance de la réponse des CNN à une image en entrée : plus la réponse du CNN est typique, plus on considère que l'image a été traitée de manière fluente. Il a ensuite utilisé cette mesure de la Fluence pour prédire des scores de beauté.
- Il y avait un certain nombre de résultats manquants dans ce stage, qu'il était intéressant d'obtenir ; ma première mission a été de le faire. Normalement, il suffisait de lancer quelques scripts depuis la ligne de commande, mais en réalité il fallait aller changer des choses dans le code, où il n'y avait pas d'interface, si bien qu'au final il a fallu écrire des lignes supplémentaires et se familiariser avec les miliers de lignes de code qui avaient été écrites avant de pouvoir lancer quoi que ce soit. Cela a été très chronophage.

### 6.2 Mécanismes d'attention

#### Approche Deep Learning de l'attention :

Les mécanismes d'attention sont un composant majeur du fonctionnement du cerveau. Ils permettent de "recycler" la fonctionnalité de certaines zones du cerveau en modifiant leur fonctionnement selon la tâche à réaliser. Dans la littérature de Deep Learning, les mécanismes d'attention sont aussi omniprésents. Ils permettent en général de donner aux modèles un moyen explicite d'ignorer l'information qui ne leur est pas importante.

Il semblerait qu'il y ait un lien, bien qu'imparfait, entre les mécanismes d'attention dans le cerveau, et dans les réseaux de neurones artificiels[27].

**Expérience :** Notre idée était de modéliser l'attention dans le cerveau par l'attention dans les réseaux de neurones, et de regarder si les phénomènes décrits par Schaeffer émergeaient pour les images les plus belles.

La plupart des méthodes d'attention de l'état de l'art consistent à calculer *à la volée* des cartes d'attention qui sont ensuite appliquées à l'image (ou à la représentation latente). Le lien est difficile à faire entre ces méthodes et la description de Schaeffer, qui implique des rétrocontroles cognitifs dans le temps (car les symptômes de l'attention esthétique sont amenés par l'expérience précédente, soit avec le même stimulus, soit avec un autre).

Nous avons choisi d'utiliser la méthode CBAM [58] (voir figure 22) pour modéliser l'attention visuelle. Dans une première version, nous avons utilisé CBAM avec les cartes d'attention en profondeur (le long des featuremaps) et spatiales, avec une contrainte de sparsité sur les cartes d'attention spatiale (voir figure 23 pour les résultats). Face à ces résultats, nous avons choisi de nous concentrer sur l'attention en profondeur car elle représente mieux le nombre de "features" auxquelles le réseau a du donner de l'importance. Nous avons modifié CBAM de deux manières pour extraire ces cartes d'attention en profondeur.

#### 6.2.1 Attention en profondeur avec sigmoïde et contrainte de sparsité (pas de résiduel)

Pour cette première modification de CBAM, nous avons retiré l'attention spatiale. Ensuite, nous avons ajouté une contrainte L1 (ajout de la somme des valeurs des cartes d'attention à l'objectif de minimisation) sur ces cartes d'attention pour les inciter à être plus sparses (le modèle est forcé d'être plus sélectif dans ce à quoi il porte attention). Les résultats avec cette méthode sont donnés à la figure 24.

#### 6.2.2 Attention en profondeur avec softmax à température variée

Le Softmax est défini ainsi : pour une liste de valeurs  $a_1, a_2, a_3, \dots$  en entrée, il modifie chaque valeur comme ceci :

$$\text{softmax}(a_i) = \frac{\exp(a_i)}{\sum_j \exp(a_j)}$$

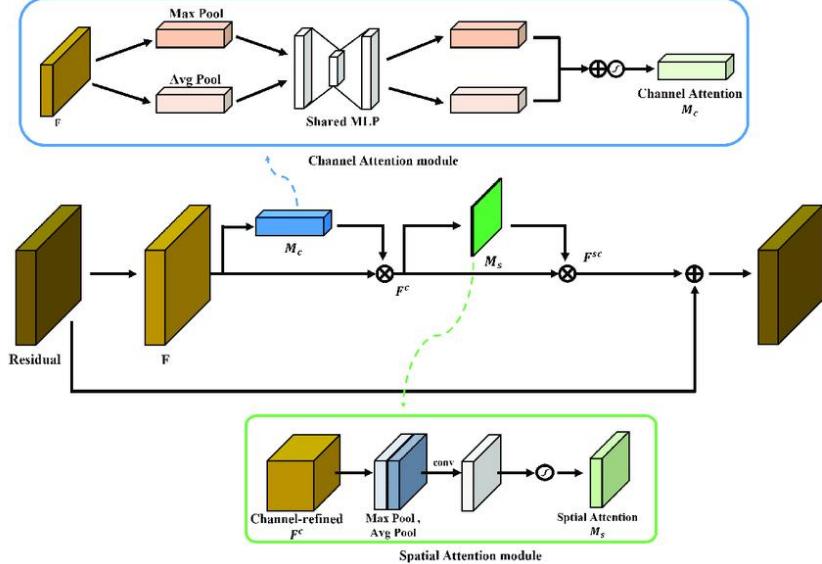


Figure 22: Le principe essentiel de CBAM est d’appliquer de l’attention en profondeur puis de l’attention spatiale. Les scores d’attention qui sont attribués à chaque featuremap (dans le cas de l’attention en profondeur) et à chaque pixel (dans le cas de l’attention spatiale) sont à chaque fois calculés avec un pooling suivi d’un perceptron multi-couches (plusieurs multiplications de matrices suivies d’une ReLU), et d’une sigmoïde pour ramener les scores entre 0 et 1.

Bien que la sigmoïde dans CBAM (et par extension le CBAM modifié avec norme L1 que nous venons de décrire) donne des valeurs entre 0 et 1, il serait intéressant d’utiliser un Softmax à la place de la sigmoïde pour les raisons suivantes :

1. Le Softmax nous donne des cartes d’activations dont la somme est toujours 1 ce qui les rend plus interprétables
2. Le Softmax (grâce à l’exponentielle) a tendance à forcer la sparsité des activations (d’où le nom *soft-max*, un opérateur max dérivable).

Un Softmax basique nous semblait très contraignant pour le réseau, ainsi nous avons choisi d’ajouter une *température t* au softmax, ce qui se traduit par :

$$\text{softmax}_t(a_i) = \frac{\exp(a_i/t)}{\sum_j \exp(a_j/t)}$$

De plus, nous avons choisi de ne mettre le softmax et les cartes d’attention que sur certaines couches du réseau. On peut voir à la figure 25 les résultats de l’entraînement pour différentes valeurs de *t* et différentes couches sur lesquelles on met l’attention.

### 6.2.3 Limites de notre approche sur l’attention :

Les méthodes calculant l’attention *à la volée* ne sont pas suffisantes pour capturer l’attention biologique. Pour améliorer cela, il faudrait y ajouter une dimension globale (comme [54]) et temporelle.

Pour l’instant notre mesure d’attention ressemble plutôt à d’autres approches [46] pour prédire la complexité des images en calculant l’importance que le modèle attribue aux différentes caractéristiques.

## 6.3 Spécialisation des modèles visuels aux ethnies et sexes

### 6.3.1 Origine de l’idée

Le stagiaire de l’année dernière (qui mesurait la fluence comme la typicalité des activations du CNN en réponse au stimulus) avait essayé de spécialiser ses distributions d’activations (dans lesquelles la typicalité est calculée comme la vraisemblance) sur des groupes ethniques/de sexe. Aussitôt, sa mesure de fluence était devenue un meilleur prédicteur de la beauté.

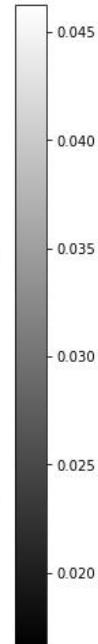
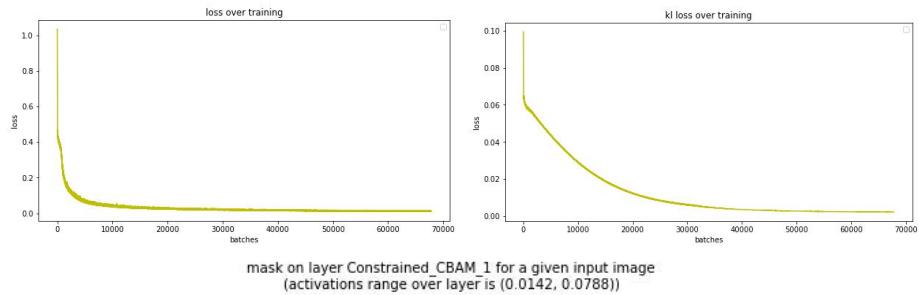


Figure 23: résultats avec CBAM complet avec juste une contrainte de sparsité sur les featuremaps spatiales. En noir et blanc, nous voyons les cartes d'attention spatiale extraites pour chaque image. Celles-là ne sont pas très interprétables ; nous avons donc choisi de n'utiliser que de l'attention en profondeur. Nous montrons aussi la valeur des fonctions de coût (reconstruction à gauche, et contrainte de sparsité sur l'espace latent) pendant l'entraînement.

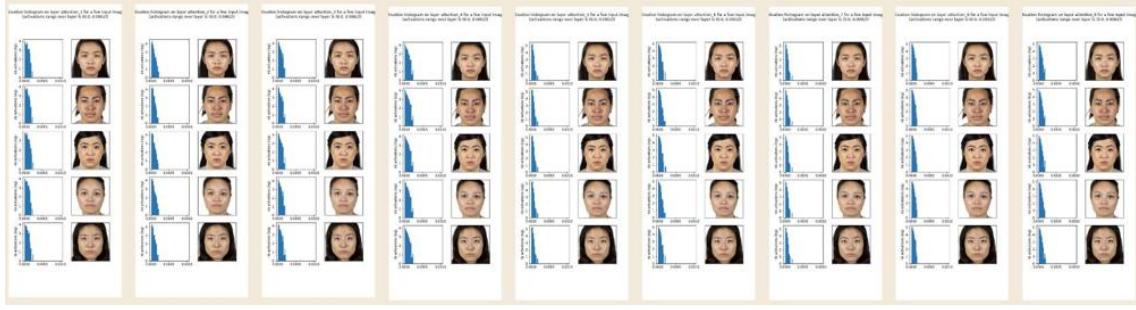


Figure 24: Résultats avec juste l’attention en profondeur et une contrainte de sparsité dessus : nous voyons que la contrainte de sparsité a tiré les histogrammes des activations vers 0, et les reconstructions sont un peu moins bonnes. Ici nous avons utilisé l’attention sur 10 couches, et les histogrammes des valeurs d’attention sont donnés. Les reconstructions ne variant pas par couches, ce sont les mêmes à chaque fois.

### 6.3.2 Interprétation biologique

L’interprétation de ce phénomène était que notre perception de la beauté serait conditionnelle de la catégorie dans laquelle nous plaçons le sujet. Ainsi, la fluence du traitement de l’information serait influencée par une attente vis-à-vis du stimulus due à la catégorie de ce dernier, et en spécifiant le modèle visuel selon cette catégorie, on modélise mieux la fluence et par extension la beauté.

### 6.3.3 Mise en place

Nous avons voulu tester cette approche de spécialisation des modèles visuels sur notre méthode. Pour cela, nous avons entraîné un VAE par sous-groupe de l’espace ethnie\*genre, que nous considérons comme le modèle du système visuel en réaction à une catégorie d’images.

Il faut noter que cette approche est un peu moins pertinente puisque nous ne comparons pas les activations à une distribution représentant **l’attente** du système visuel pour certaines images. Néanmoins, nos modèles spécialisés pourraient représenter le "chemin" que prend l’information dans le cerveau conditionnellement de la nature de l’objet perçu, ou bien une modification directe du système visuel par des rétrocontrôles cognitifs (de l’attention ou encore du predictive coding [32]) qui l’adapterait dynamiquement au type de stimulus détecté.

Nous n’avons pas encore les résultats de l’entraînement de nos modèles sur chaque sous-classe car les modèles ont été entraînés sur Jean Zay mais il faut encore générer les figures.

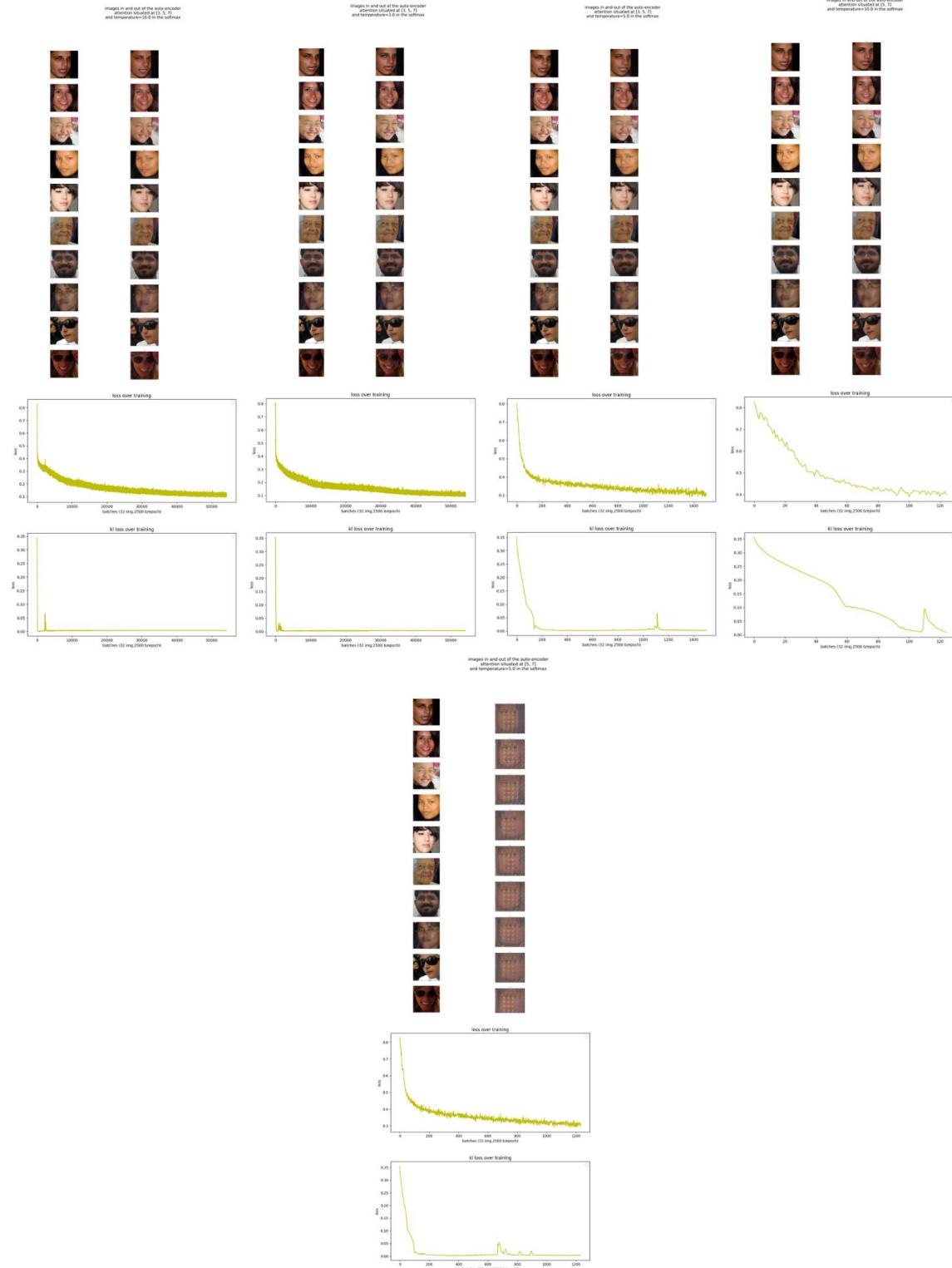


Figure 25: Reconstructions et fonctions de coût (reconstruction au dessus, et contrainte de sparsité sur l'espace latent en dessous) avec l'attention en profondeur avec softmax, située pendant l'entraînement à différentes couches, et avec différentes températures.

Pour beaucoup de configurations non montrées ici, la loss était NaN au bout de quelques epochs (comme dans le cas 57-5.0 ici où l'on voit que les reconstructions sont mauvaises et la fonction de coût stagne au dessus de 0.3)

## 7 Ajouts à la méthode

### 7.1 Tentatives de reconstruction sans décodeur

#### 7.1.1 La motivation :

Comme expliqué plus haut, nous cherchons à modéliser la fluence du traitement neuronal par deux métriques : l'économie, et l'efficacité du traitement de l'information. En particulier, pour l'efficacité du traitement neuronal, nous voulons savoir quelle quantité d'information importante l'encodeur (notre modèle du cerveau) a réussi à extraire. Un problème avec la mesure de cela avec l'erreur de reconstruction est le biais introduit par le décodeur. Il serait intéressant d'avoir un moyen d'obtenir des reconstructions, sans pour autant introduire un décodeur dans la balance.

#### 7.1.2 La méthode :

La méthode que nous avons adoptée pour reconstruire une image traitée par l'encodeur du VAE sans utiliser le décodeur est la suivante :

1. Passer l'image d'entrée  $X$  dans l'encodeur pour obtenir la représentation latente  $Z_X$ .
2. Passer une image de bruit  $N$  dans l'encodeur pour obtenir la représentation latente  $Z_N$ .
3. Calculer la distance L2 entre ( $Z_X$  et  $Z_N$ ), notée  $D$ . Avec l'autodifférentiation de PyTorch, calculer  $\frac{\delta D}{\delta N}$ .
4. Modifier  $N$  pour que sa représentation latente se rapproche de celle de  $X$  :  $N \leftarrow N - \lambda * \frac{\delta D}{\delta N}$ .
5. Recommencer les étapes 2 à 4 jusqu'à convergence de  $D$ .

#### 7.1.3 Le problème rencontré :

En pratique, quand nous mettons en place cette méthode, nous obtenons les résultats donnés à la figure 26.  $D$  devient très petit, ce qui indique que la représentation de  $N$  devient la même que celle de  $X$  dans l'espace latent du VAE. Néanmoins, dans l'espace des images,  $N$  reste très différente de  $X$ . Qu'est-ce qui cause ce comportement ?

#### 7.1.4 Exemples adversariaux :

Un des plus grands problèmes avec les CNN est leur taille gigantesque, largement suffisante pour apprendre par cœur toute la base de données d'entraînement. Un problème associé à cela est que les CNN se basent sur des motifs imperceptibles dans images en entrée pour faire leurs décisions. Si l'on introduit artificiellement ces motifs dans une image, comme ils sont imperceptibles, on peut se retrouver avec une interprétation par le CNN de l'image complètement différente de celle de l'humain. Par exemple, une image de chat peut être modifiée de manière imperceptible pour l'œil humain, de manière à ce qu'un CNN croie qu'il s'agit d'un hélicoptère. Une telle image s'appelle un *exemple adversarial* [10]. Dans le cas d'un Autoencodeur, cette fausse interprétation de l'image se traduit par ce que l'on voit à la figure 27.

#### Clever Hans :

Pour mieux expliquer le problème des exemples adversariaux, Ian Goodfellow compare cela au cas de l'âne "Clever Hans". *Clever Hans était un âne qui semblait capable de compter, car quand on le mettait sur la scène et lui énonçait un calcul arithmétique basique, il tapait du sabot le nombre de coups correspondant à la solution du calcul. Cependant, un jour où on lui murmura à l'oreille le calcul, il fut soudain incapable de le résoudre. On se rendit alors compte de la stratégie de Clever Hans : il tapait du sabot jusqu'à ce que l'audience retienne son souffle, ce qui indiquait qu'il était arrivé au bon nombre de coups.* Clever Hans se basait sur un indice accidentel qui lui permettait une bonne performance, en ignorant totalement la nature du problème à résoudre. Les exemples adversariaux sont-ils la preuve que les CNN se comportent de la même manière ? Il n'y a pas de consensus là-dessus. Par exemple, dans [20], les auteurs montrent que ces motifs imperceptibles sur lesquels se basent les CNN sont les mêmes d'une base de données à une autre, ce qui indique qu'ils ne sont pas si *accidentels* que cela.

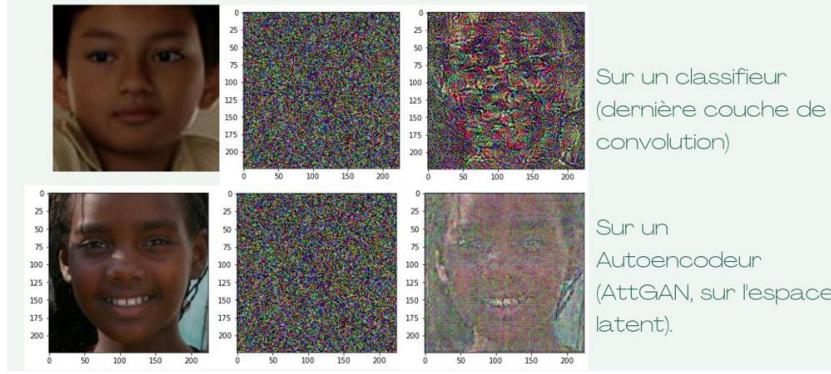


Figure 26: reconstruction par descente de gradient, sur VGGface et sur AttGAN – pour chaque ligne, dans l’ordre : originale, point de départ, image après convergence. Il est intéressant de noter que les reconstructions, bien qu’insuffisantes, sont meilleures pour un autoencodeur que pour un classifieur, ce qui indique une représentation latente plus complète.

Quoi qu’il en soit, dans notre cas, ce défaut des CNN rend notre méthode de reconstruction sans décodeur défaillante.

## 7.2 Utilisation d’une contrainte de sparsité

Une des tâches réalisées pendant le stage a été la mise en place d’une contrainte à l’entraînement du modèle pour forcer ses activations à être plus sparses.

### 7.2.1 Motivation biologique :

L’hypothèse du codage efficace [1] de Barlow énonce que le cerveau, sous une contrainte de dépense d’énergie, aurait tendance à encoder l’information de manière sparse. Ce n’est pas le cas pour nos CNN. Il a été montré [35] que l’ajout d’une contrainte L1 (en ajoutant à l’objectif de minimisation la norme L1 des activations) fait émerger des filtres semblables à ce que l’on trouve dans le cortex visuel. Puisque nous sommes intéressés par la modélisation du cerveau par les CNN, il est intéressant de leur ajouter une telle contrainte pour potentiellement amplifier la ressemblance au cerveau. Additionnellement, la norme  $L_\epsilon$  que nous utilisons comme mesure de sparsité demande que les distributions des activations soient resserrées autour de 0 et avec une queue gauche beaucoup plus courte que la queue droite.

### 7.2.2 La contrainte mise en place

Nous avons mis en place la contrainte de minimisation de la distance de Kullback-Leibler entre la distribution des activations, et une distribution uniforme localisée sur une valeur  $\rho$  (contrainte directement tirée de [37]). Concrètement, nous ajoutons à l’objectif de minimisation la valeur

$$\sum_n (\rho \log(\frac{\rho}{\rho_n}) + (1 - \rho) \log(\frac{1-\rho}{1-\rho_n}))$$

Où  $n$  correspond à un neurone (on boucle sur tous les neurones d’une couche),  $\rho$  à la localisation de la loi uniforme, et  $\rho_n$  à l’activation moyenne du neurone  $n$  pour un certain nombre d’images en entrée (dans notre cas 32 en général). De plus, nous utilisons un paramètre  $\beta$  par lequel nous multiplions cette contrainte dans l’objectif de minimisation, afin de lui accorder plus ou moins d’importance (si  $\beta$  est très grand, on minimisera la contrainte de sparsité plus que l’erreur de reconstruction, et inversement).



Figure 27: Démonstration de l'exemple adversarial : gauche : rappel des reconstructions avec attGAN. Seconde colonne : reconstruction sans décodeur avec un VAE. Troisième colonne : images d'origine. Dernière colonne : reconstruction par le VAE des images de la seconde colonne.

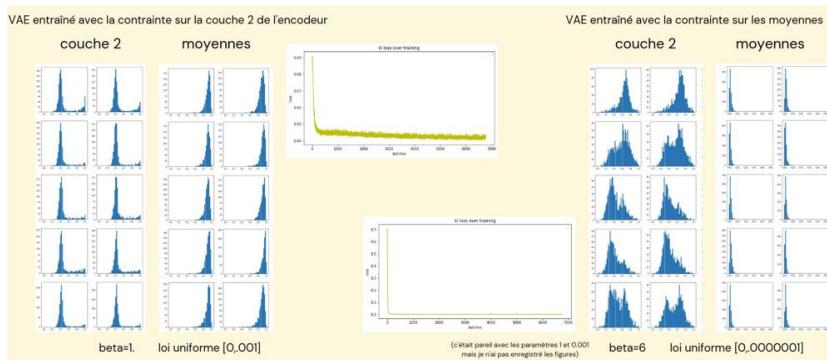


Figure 28: Les histogrammes montrés sont les histogrammes de fréquence d'activations sur toutes les images de CFD, pour un seul neurone à chaque fois. Nous voyons ici que la contrainte fait son effet (les distributions tendent vers 0) sur la couche des moyennes, mais pas sur la couche 2 (sur le petit modèle). Cela était en fait du à l'Instance Normalization qui se trouvait sur les 4 premières couches et empêchait la contrainte de faire son effet. Cela paraît difficile à expliquer.

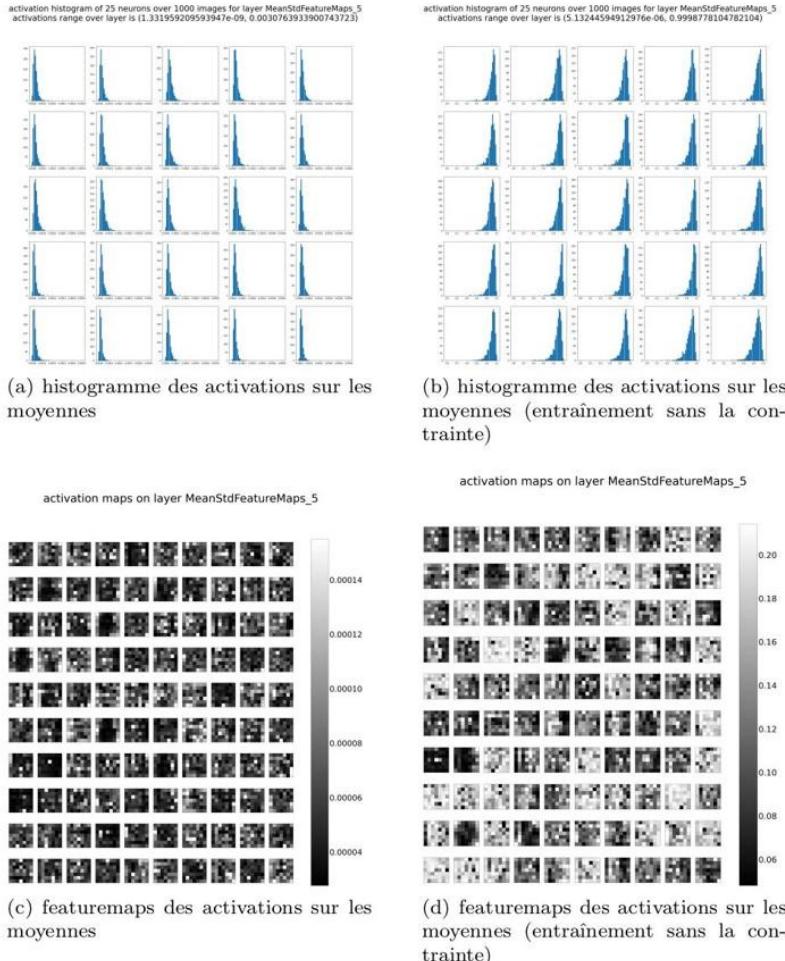


Figure 29: Effet de la contrainte sur les activations : elle réduit grandement l'amplitude en tirant les distributions vers une constante, mais augmente aussi la sparsité. Au niveau des Featuremaps, nous voyons avec la contrainte moins de blanc et plus de noir, ce qui indique que les activations sont plus localisées.

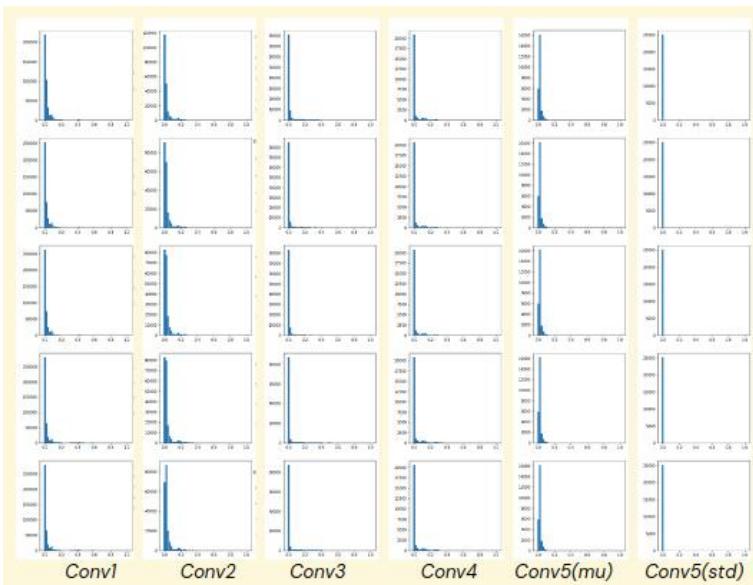


Figure 30: Histogrammes des activations par neurones sur toutes les images de CFD, sur le modèle avec 5 couches au décodeur et à l'encodeur, sans l'instance normalization, et avec la contrainte de sparsité. Nous voyons que la contrainte fonctionne pour ramener les activations vers 0, mais qu'elle ne crée aucune quantification séparation activation/non-activation (ce que nous espérions).

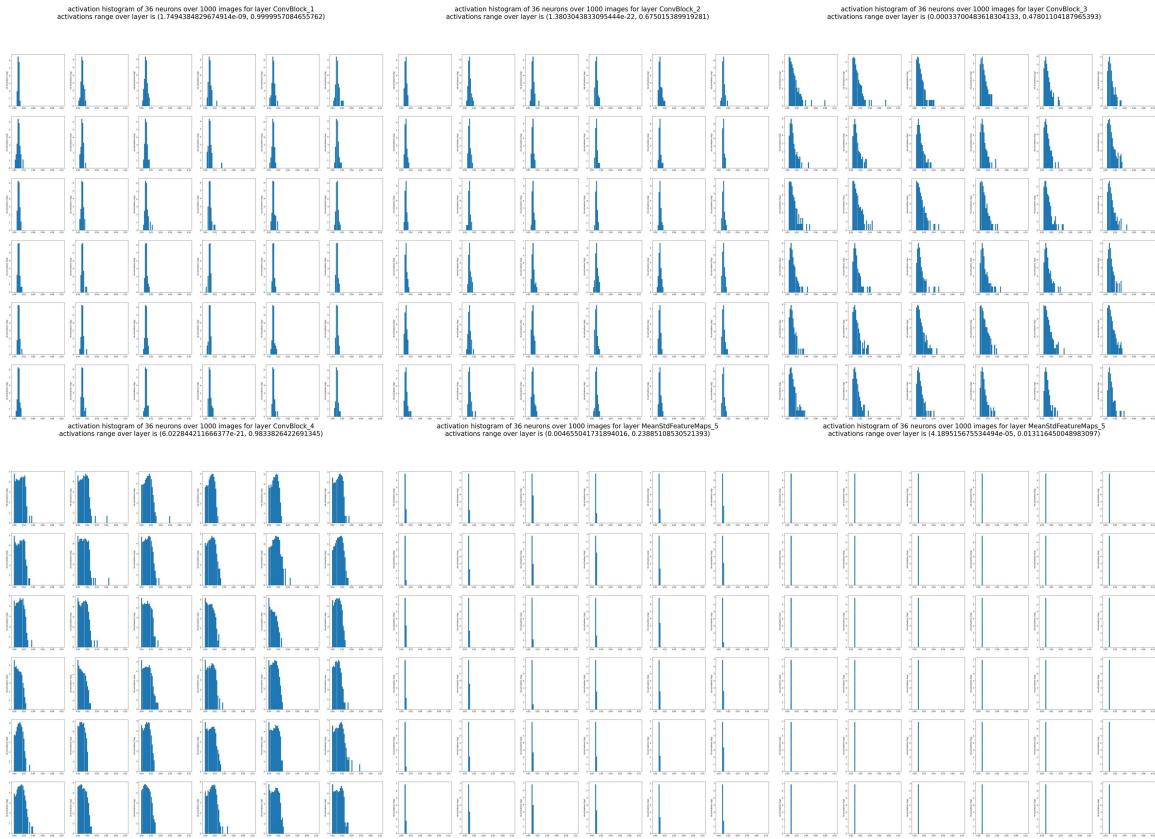


Figure 31: Histogrammes des activations sur le petit modèle quand la contrainte est située en 0.000001. Nous donnons les histogrammes d’activations pour 25 neurones, (matrice de 5x5), sur toutes les images de CFD. Ce sont donc les histogrammes de fréquence d’activations sur toutes les images de CFD, pour un seul neurone à chaque fois. Nous voyons que le fait de rapprocher la loi uniforme de 0 n’a pas d’effet positif sur les distributions. Il y a 6 matrices de 25 activations, à lire dans le sens de lecture : couche 1, 2, 3, 4, puis moyennes et finalement écart-type.

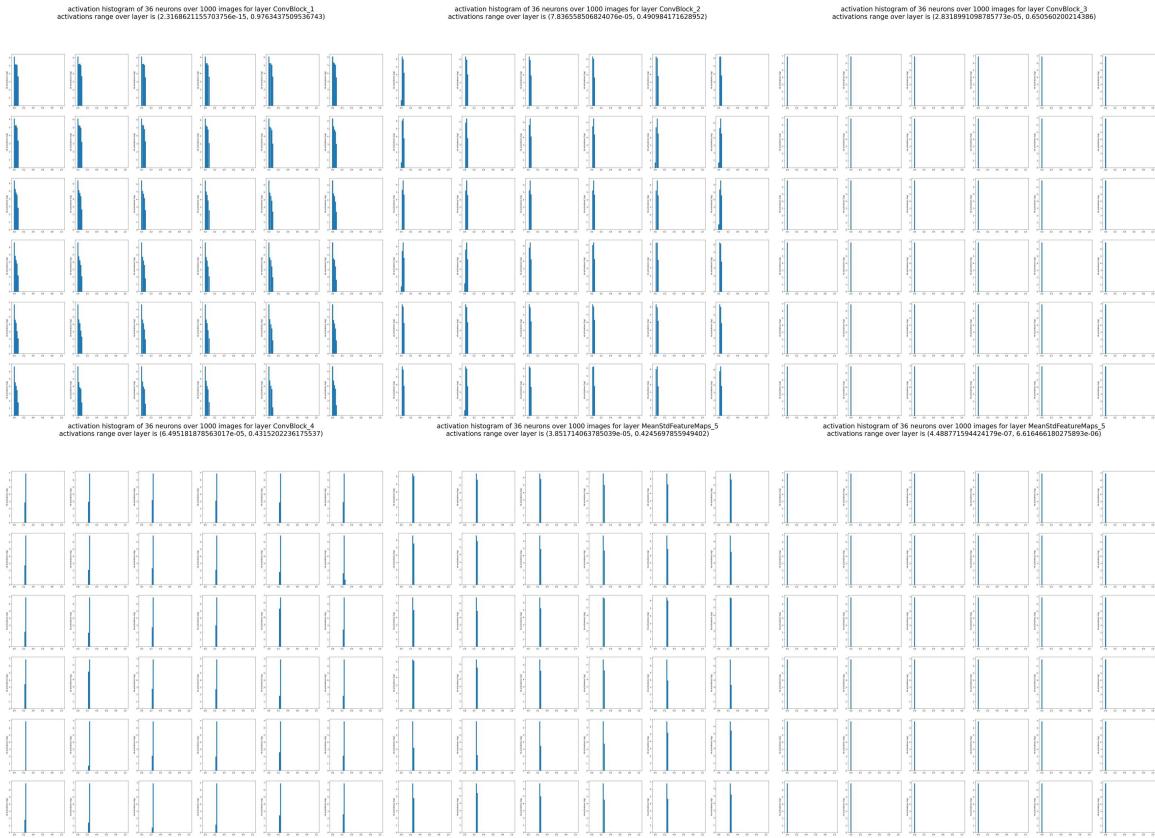


Figure 32: même chose que la figure 31 quand la contrainte est multipliée par 3 dans la fonction finale de coût. Nous ne voyons toujours pas la séparation espérée.

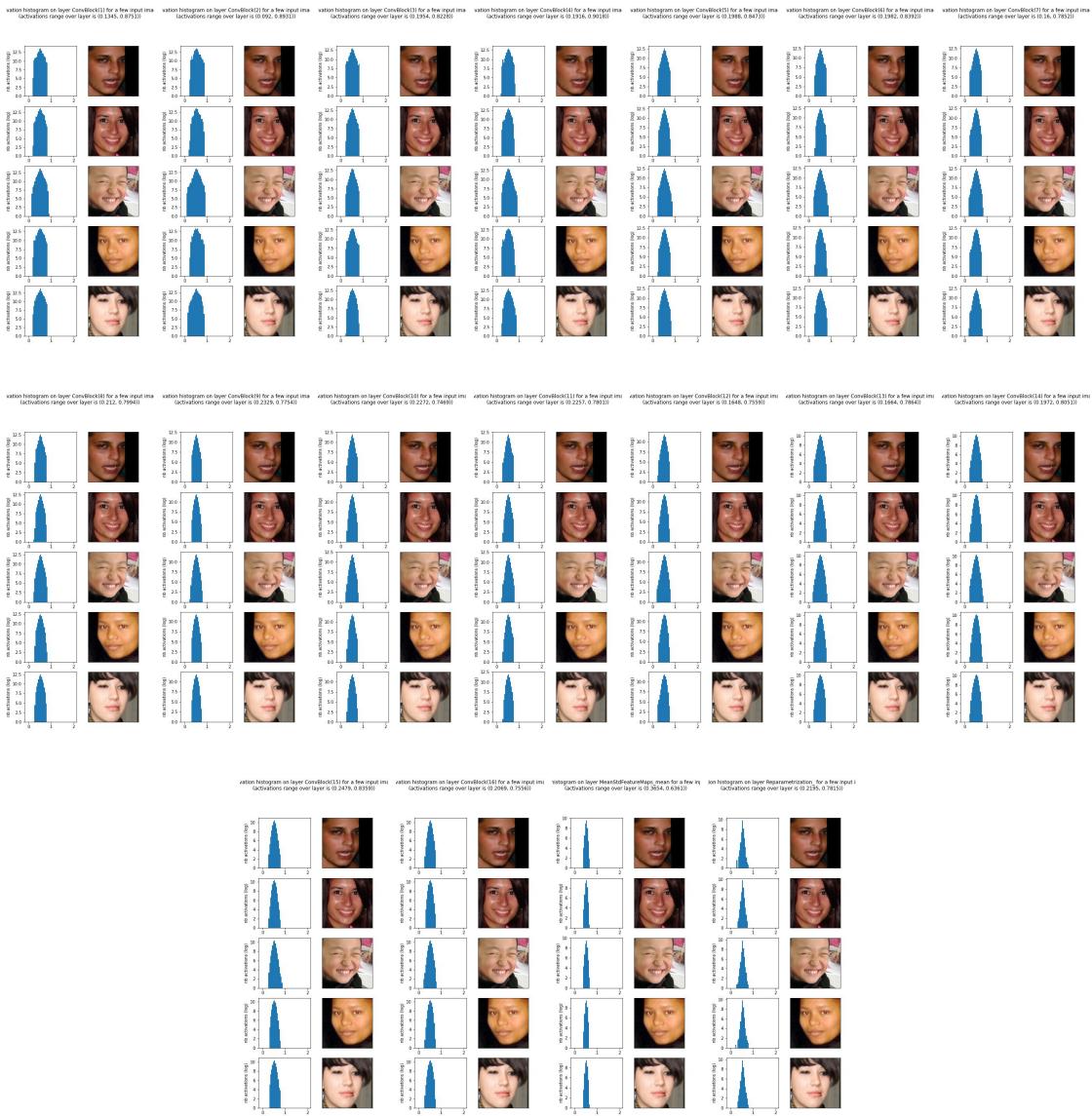


Figure 33: Histogrammes (cette fois-ci les histogrammes pour toute la couche, sur une seule image, au lieu de le faire sur un seul neurone pour toutes les images) sur le VAE avec VGG en encodeur et 10 couches au décodeur, sans la contrainte. Il y a 5 histogrammes par couche de l'encodeur, et ils sont donnés dans l'ordre des couches ici.

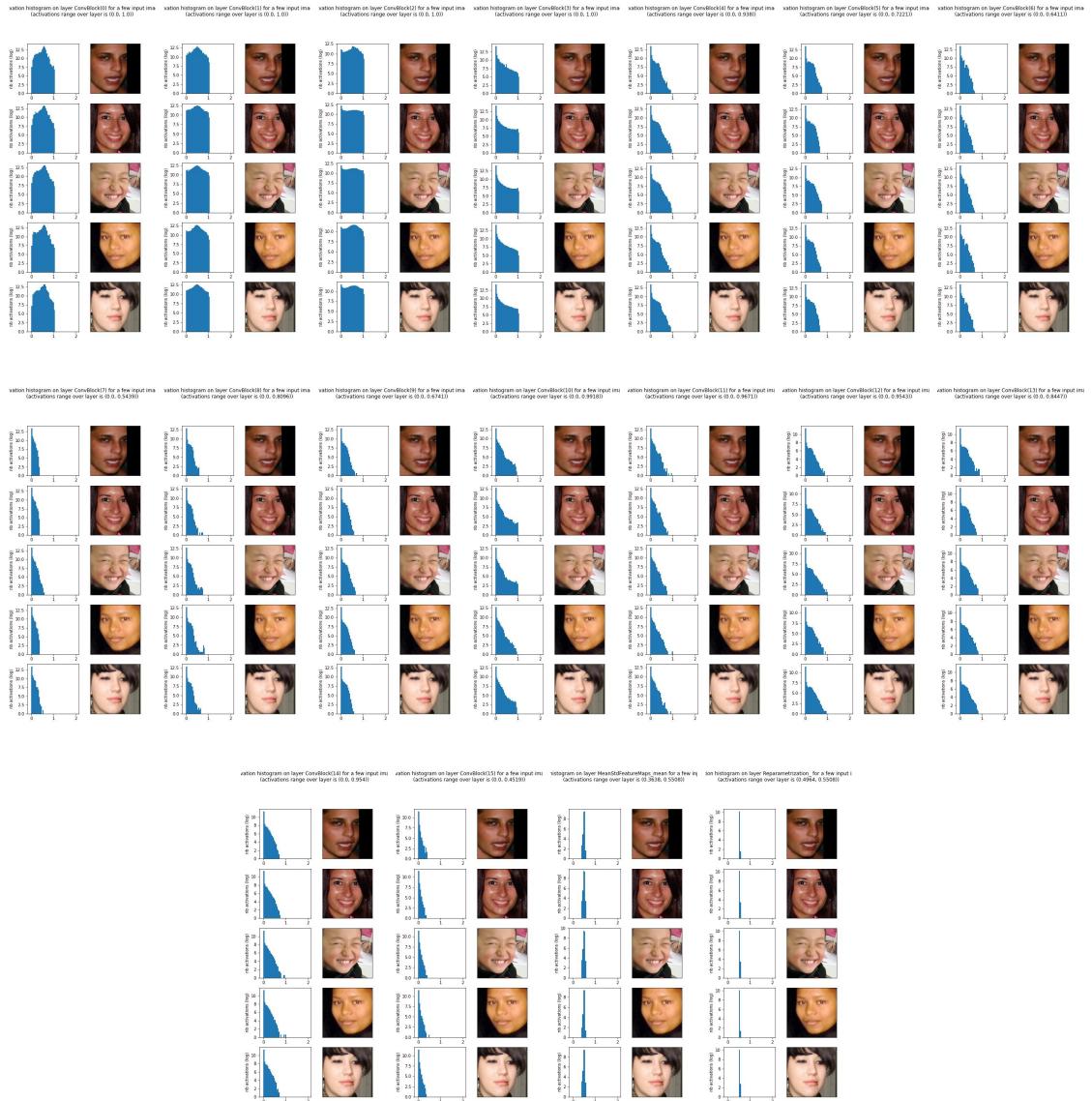


Figure 34: Histogrammes sur le VAE avec VGG en encodeur et 10 couches au décodeur, avec la contrainte. Il semblerait que la contrainte marche un peu moins bien ici, peut-être parce que l'encodeur, plus long, a besoin de conserver plus l'information, et peut se permettre moins de sparsité. Ici aussi, les histogrammes sont donnés dans l'ordre des couches de l'encodeur.

## 8 Pistes abandonnées

### 8.1 Pistes abandonnées

Nous présentons ici brièvement un certain nombre de pistes que nous avons considérées plus ou moins profondément (parfois même testées) au cours du stage, pour finalement les laisser tomber, par manque de temps ou parce que ça ne marchait pas.

#### 8.1.1 Normalisation Spectrale

##### Idée de la Normalisation Spectrale

L'algorithme de descente de gradient reposant sur la validité du gradient comme direction locale d'augmentation maximale de la fonction de coût, il serait intéressant de pouvoir s'assurer que le gradient soit stable et pas trop grand. Une manière de formuler cela est :

$$\|\mathcal{L}(x) - \mathcal{L}(y)\| \leq K * \|x - y\|$$

Où  $x$  et  $y$  sont des vecteurs quelconques de nombres réels, et  $K$  est une constante. Si  $\mathcal{L}$  respecte cette condition, on dit qu'elle est  $K$ -Lipschitz. L'heuristique de la Spectral Normalisation est la suivante : *si la fonction de coût est 1-Lipschitz, l'entraînement sera plus stable.*

##### Fonctionnement

La *norme spectrale* d'une fonction est la valeur maximale de toutes les valeurs singulières de sa jacobienne en tous les points de l'espace. En pratique, nous appliquerons la normalisation spectrale individuellement sur chacune des couches de notre modèle, sans la fonction d'activation, ce qui représente donc une application linéaire qui aura la même jacobienne pour tous les points de l'espace.

Considérons les deux résultats suivants :

- La norme spectrale  $N_s$  d'une application linéaire est égale à sa constante de Lipschitz (c'est-à-dire que l'application est  $N_s$ -Lipschitz). (preuve non fournie)
- Il est assez intuitif que la norme spectrale de l'application linéaire multipliée par un scalaire  $\lambda$  est égale à  $\lambda$  fois sa norme spectrale.

Nous pouvons alors concevoir qu'en calculant la norme spectrale des couches de notre réseau et en divisant les poids par elle, on aura des couches 1-Lipschitz. Comme nous utilisons des ReLU comme fonctions d'activation, que la ReLU est 1-Lipschitz, et que la composition de fonctions 1-Lipschitz est aussi 1-Lipschitz, en réalisant cette opération de division des poids par leur norme spectrale, nous rendons le réseau en entier 1-Lipschitz.

Le dernier élément est comment calculer la norme spectrale. Pour cela, la *méthode de la puissance itérée* est communément utilisée.

##### Utilité

Après son invention, la Normalisation Spectrale a d'abord été appliquée sur des Réseaux de Neurones Antagonistes Génératifs [9] où elle permettait de diversifier les images générées par le générateur. Nous avons utilisé cette technique dans notre VAE (en appliquant la normalisation à toutes les couches) car les auteurs du nVAE [53] affirment que c'est utile pour les VAE. Néanmoins, comme on peut le voir à la figure 35, dans notre cas cela ne changeait rien.

#### 8.1.2 Autres contraintes de sparsité

Nous avons ici fait un choix d'une contrainte de sparsité parmi un certain nombre de possibilités. Notamment, une contrainte conçue pour la sparsification de réseaux de neurones en vue de les compresser [29], et une norme L1 [35] comme nous l'avons discuté plus haut, ont été considérées comme alternatives pendant le stage.



Figure 35: Les reconstructions avec la spectral normalization ne sont ni meilleures, ni pires que sans (petit modèle ici).

### 8.1.3 Generative Invertible Flows

Les Generative Invertible Flows [24] sont un modèle génératif à variable latente (comme les VAE) dont le principe est de créer une transformation inversible vers l'espace latent. L'inversion ne se fait pas à l'aide d'un décodeur, mais grâce aux propriétés des opérations utilisées pour le passage de l'entrée à l'espace latent (voir figure 36 pour des résultats de ce type de modèle).

Les Generative Invertible Flows auraient été une alternative intéressante aux VAE dans la mesure où ils permettraient de contourner le biais inhérent au décodeur. Néanmoins, en se renseignant, nous sommes tombés sur plusieurs résultats qui nous ont détournés de leur utilisation :

- Les opérations inversibles utilisés sont un peu éloignés des opérations de convolution de nos VAE, ce qui nous éloignerait de notre modèle biologique.
- L'espace latent a tendance à avoir une haute dimensionnalité dans les Generative Invertible Flows par rapport aux VAE (donc potentiellement moins d'information extraite explicitement (voir figure 21 comparé à 19)).

### 8.1.4 Information mutuelle

#### Le principe de l'information mutuelle :

L'information mutuelle entre deux variables aléatoires représente à quel point une distribution dépend probabilistiquement de l'autre. On peut en obtenir une vision intuitive en regardant la formule. L'information mutuelle entre les variables aléatoires X et Y est :

$$I(X;Y) = \sum_{x,y} P(x,y) * \log\left(\frac{P(x,y)}{P(x)P(y)}\right)$$

Nous voyons que l'information mutuelle est en fait la divergence de Kullback-Leibler entre la distribution jointe des deux variables, et le produit des deux distributions marginales. En se rappelant que si deux événements sont indépendants on a  $P(x,y) = P(x)P(y)$ , on peut voir que l'information mutuelle va augmenter plus  $x$  et  $y$  seront des événements dépendants.

#### Utilité dans notre cas :

Toujours dans le même objectif de mesurer l'efficacité du traitement de l'information par l'encodeur en se passant du décodeur, nous voulions utiliser l'information mutuelle entre l'image en entrée et les représentations latentes. Un avantage supplémentaire de cette méthode aurait été de pouvoir obtenir une mesure pour chaque couche plutôt qu'avoir une seule reconstruction pour tout le modèle. Avoir cette mesure d'information mutuelle entre la représentation latente à la sortie de chaque couche, et



Figure 36: Samples du modèle Glow (pas des reconstructions, mais bien des samples artificiels tirés aléatoirement de l'espace latent).

l'image en entrée, aurait permis différentes analyses sur l'effet des différentes couches sur la perception de la beauté.

#### Problème :

L'information mutuelle se calculant entre deux variables aléatoires, il n'est pas possible de la calculer sur une seule image (il faut plusieurs images et plusieurs représentations latentes correspondantes pour créer des distributions). Même avec plusieurs échantillons, les estimateurs ne sont pas parfaits [50]. Pour notre utilisation, il aurait fallu un mesure par image, donc nous avons abandonné cette piste.

#### 8.1.5 Convolutions Séparables en Profondeur

**Le principe :** Dans le domaine du traitement des images, des filtres de convolution en deux dimensions sont souvent utilisés pour lisser des images, ou encore extraire les contours. Une manière d'optimiser l'opération de convolution est d'exprimer le filtre comme le produit extérieur entre deux vecteurs. Par exemple, le filtre basique de détection de contours

$$\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix}$$

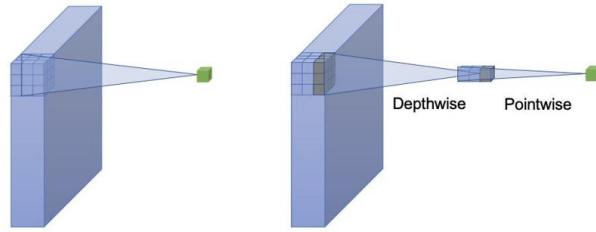
peut s'écrire :

$$\begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \otimes \begin{bmatrix} -1 \\ 0 \\ 1 \end{bmatrix}$$

L'opération de convolution avec un des petits vecteurs puis avec l'autre est équivalente à la convolution avec le filtre tout entier, mais elle prend moins de mémoire et moins de temps de calcul.

Les *Convolutions Séparables en Profondeur* sont une technique provenant de l'architecture MobileNet [17], qui consiste à exprimer les filtres à trois dimensions que l'on utilise dans nos couches de convolution (hauteur, largeur, features) comme une opération de convolution avec un filtre en deux dimensions (hauteur,largeur) et un filtre à une dimension (features) (voir 37). L'avantage est une diminution forte du temps de calcul et de la mémoire nécessaire. Le prix à payer est que les filtres séparables sont limités en termes d'expressivité, mais en pratique cela ne semble pas déranger énormément.

Dans nVAE [53], les auteurs recommandent l'usage des convolutions séparables en profondeur, car elles permettent d'augmenter la taille des filtres de convolution sans prendre plus de place en mémoire. Nous avons testé cette technique (qui nous a permis de passer de filtres 3x3 à des filtres 7x7), mais elle n'améliore pas les résultats dans notre cas (voir figure 38).



**Figure 3: Standard convolution and depthwise separable convolution.**

Figure 37: Le principe des convolutions séparables en profondeur est de transformer une convolution avec un filtre 3D (gauche) en une convolution avec un filtre 2d suivie d'une convolution avec un filtre 1D (droite).



Figure 38: Reconstructions du petit modèle avec des filtres de taille 7x7 au lieu de 3x3 en prenant la même place en mémoire, grâce aux convolutions séparables en profondeur (c'est pire)

### 8.1.6 Pistes pour la reconstruction sans décodeur

Après le premier mois et demi du stage, nous avions abandonné l'idée de reconstruire les images sans le décodeur, en optimisant une image pour qu'elle ait la même représentation latente que l'image cible. Pendant le reste du stage, nous sommes tombés sur différentes pistes d'amélioration de cette technique, qu'il aurait été intéressant d'explorer (notamment, peut-être qu'une combinaison de ces techniques aurait pu grandement améliorer les reconstructions).

#### Approches anti-adversarielles

Les exemples adversariels pouvant présenter un problème de sécurité dans des applications basées Deep Learning, différentes méthodes ont été créées pour lutter contre cette faiblesse.

Il est par exemple possible de générer des exemples adversariels *pendant l'entraînement* et d'apprendre au réseau à les classer néanmoins dans la bonne classe. Il existe des bases de données augmentées avec des exemples adversariels [14].

Une approche qui paraît plus élégante et un peu moins *ad-hoc* est l'approche *Local Winner Takes All* (LWTA[36]), où un mécanisme de compétition locale entre les neurones du CNN, inspiré par le fonctionnement du cerveau, est ajouté, ce qui lui donne une robustesse aux exemples adversariels sans avoir besoin d'en ajouter dans la base d'entraînement.

#### Détection de sortie de la distribution

Dans le domaine des VAE, un problème récurrent est le fait que les VAE vont créer dans l'espace latent une distribution des images de la base d'entraînement, et placer au milieu de cette distribution des images n'ayant rien à voir avec (par exemple, une image de bruit gaussien se retrouve juste à côté d'une image d'un visage). Cela pourrait être une bonne explication de ce qui nous arrive quand nous essayons de faire une reconstruction sans le décodeur. Il existe des méthodes pour rendre l'espace latent robuste à ces "sorties de distribution" [6] [40].

#### Restriction de l'espace des images :

Une autre manière d'aborder le problème est dans l'espace des images plutôt que dans l'espace latent. La raison pour laquelle notre optimisation sans décodeur donne sur des images non-réalistes est que notre encodeur n'est pas injectif : pour la même représentation latente, il existe plusieurs images en entrée. Seulement un sous-ensemble de ces images seront réalistes. Si nous pouvions ajouter un a-priori (par exemple sous la forme d'une contrainte sur l'image optimisée) sur l'espace des images qui le restreint aux images réalistes, notre optimisation devrait tomber sur de meilleures reconstructions. La version la plus connue de cette méthode est DeepDream [33] (voir figure 39).

Néanmoins, deux défauts de cette approche se présentent : premièrement, il est doutable que nous obtiendrions les reconstructions parfaites dont nous avons besoin, et deuxièmement, avec cette approche, nous prendrions le risque d'introduire un nouveau biais dans l'a priori, et donc de nous débarrasser du biais du décodeur pour introduire un nouveau biais autre part.



Figure 39: Deep dream fait une attaque adversarielle avec descente de gradient, seulement avec un apriori sur l'espace des images. Qu'est-ce que ça aurait donné sur un autoencodeur, qui plus est spécialisé sur les visages ?

## 9 Résultats

### 9.1 Analyse statistique en lien avec la beauté

#### 9.1.1 Corrélations

##### Liste des métriques

Au fil du stage, différentes pistes ont été explorées et différentes métriques de la fluence développées. En voici une liste :

- L'acuité SAM
- L'erreur LPIPS de reconstruction
- L'index de gini des activations des couches
- La norme  $L_\epsilon$  des activations des couches (dans la mesure où la contrainte était appliquée à l'entraînement)
- La norme L1 des cartes d'attention.

##### Attentes

Nous expliquons ici, dans le cadre de notre sujet sur la fluence et la beauté, quelle est notre attente de la corrélation que chaque métrique devrait avoir avec les scores de beauté sur CFD.

- L'acuité SAM : comme nous l'utilisons comme mesure du point auquel le réseau a *compris* l'image, elle serait une mesure d'efficacité du traitement de l'information et donc positivement corrélée aux scores de beauté.
- L'erreur LPIPS de reconstruction : elle nous permet de tester la qualité de la représentation latente extraite par l'encodeur, en regardant si le décodeur reconstruit bien l'image. Ainsi, c'est une mesure d'efficacité du traitement de l'information qui devrait être positivement corrélée aux scores de beauté.
- L'index de gini des activations des couches : nous nous en servons comme une mesure de l'économie d'effort pendant le traitement de l'information. Un index de gini haut indique une haute sparsité, et un traitement aisément de l'information. Il serait attendu que l'index de gini soit positivement corrélé avec les scores de beauté.
- La norme  $L_\epsilon$  des activations des couches (dans la mesure où la contrainte était appliquée à l'entraînement) : plus la norme  $L_\epsilon$  est haute, plus on considère que de nombreux neurones se sont activés, et donc que l'effort de traitement de l'information a été grand. La norme  $L_\epsilon$  serait donc négativement corrélée avec les scores de beauté dans le cadre de la théorie de la Fluence.
- La norme L1 des cartes d'attention : puisqu'elle mesure la quantité de caractéristiques différentes que le modèle a du prendre en compte pour traiter l'information, une haute valeur indiquerait, dans le cas de la Fluence, une image difficile à traiter. Cette métrique serait donc négativement corrélée à la beauté.

#### Méthode

Pour obtenir les corrélations entre les scores de beauté et les métriques, nous avons passé chacune des images de la base de données CFD dans les encodeurs de nos différents réseaux, en gardant en mémoire les représentations intermédiaires à la sortie de chaque couche. Nous avons calculé les mesures gini et  $L_\epsilon$  sur ces représentations intermédiaires. Ensuite, nous avons obtenu les reconstructions et calculé la valeur LPIPS, et finalement l'acuité SAM. Pour les modèles où nous avions ajouté des mécanismes d'attention, nous avons extrait les cartes d'attention et calculé leur norme L1.

Ainsi, nous obtenions autant de valeurs pour chaque métrique qu'il y a de valeurs de beauté dans la base de données CFD. Entre le vecteur de chaque métrique et le vecteur des scores de beauté, nous pouvions ainsi calculer une corrélation.

Dans le cas de gini,  $L_\epsilon$  et des cartes d'attention, nous pouvions calculer une corrélation pour chaque représentation latente.

Nous avons aussi regardé les P-values de nos corrélations (voir figure 40 pour un exemple), et en général elles sont inférieures à .05 quand la corrélation a une valeur supérieure à .1.

De plus nous avons regardé les corrélations de Spearman (corrélation de Pearson après remplacement des valeurs par leur index dans la liste triée des valeurs), pour vérifier que nous ne passions pas à côté de relations non-linéaires nous avons à chaque fois regardé les corrélations de Spearman en même temps que les corrélations de Spearman. Nous n'avons pas trouvé d'informations supplémentaires de cette manière.

Finalement, comme la corrélation de Spearman peut aussi passer à côté de certaines relations (par exemple, une distribution à deux variables qui forme un cercle dans l'espace aura une corrélation de Spearman de 0 entre les deux axes), nous avons aussi regardé les graphes des distributions jointes entre les valeurs des métriques, et les scores de beauté (voir figure 45 pour un exemple). Encore une fois, nous n'avons rien trouvé de particulier de cette manière.

## Résultats

La figure 47 donne les corrélations de Pearson entre l'index de gini des activations et les scores de beauté, pour toutes les couches de l'encodeur du modèle VGG19 avec 10 couches au décodeur, et avec un mécanisme d'attention (la version softmax) sur la couche 5.

La figure 48 montre la corrélation entre l'index de gini calculé sur toutes les couches à la fois, ainsi que la mesure d'acuité SAM, et les scores de beauté sur CFD (sur le modèle VGG19 avec 10 couches au décodeur).

La figure 49 montre les corrélations de Pearson entre les métriques gini et  $L_\epsilon$ , et les scores de beauté, pour les 5 couches de l'encodeur du petit modèle. La corrélation entre les scores de beauté et l'erreur de reconstruction y est aussi donnée. On y voit deux versions de chaque métrique : une sur un modèle entraîné avec la contrainte, et une autre avec un modèle entraîné sans.

La figure 50 donne la même chose que la figure 49, mais sans l'erreur de reconstruction (ce modèle ne reconstruisait pas assez bien), et sur le modèle VGG19 avec 5 couches au décodeur. Les couches sont rangées dans l'ordre de lecture (en haut à gauche , couche 1, juste à côté, couche 2, etc...).

## Interprétation

Il semblerait que les corrélations soient en général assez basses. Même parmi celles qui sont au dessus de 0.1, il est difficile de détecter un motif particulier. Nous passons ici les métriques en revue et commentons sur leur effet.

- L'erreur LPIPS de reconstruction : cette métrique non plus semble ne donner aucune
- L'acuité SAM : cette métrique semble ne donner aucune information sur les scores de beauté. Puisqu'elle est une mesure de l'acuité de l'erreur de reconstruction, nous aurions pu nous y attendre.
- L'index de gini des activations des couches : Sur les trois modèles, il est difficile de dire s'il est généralement positivement ou négativement corrélé avec la beauté. Une chose potentiellement intéressante est que sur le petit modèle et sur le modèle VGG19 avec 5 couches au décodeur, il semblerait que le signe de la corrélation alterne d'une couche à la suivante de manière systématique. Cependant, il est difficile de dire ce que cela voudrait dire dans le cadre de la théorie de la Fluence. L'index de gini sur toutes les couches à la fois, sur le modèle
- La norme  $L_\epsilon$  des activations des couches : tout comme pour l'index de gini, on ne retrouve pas de tendance à être positivement ou négativement corrélé, même si on retrouve aussi ce motif d'alternance d'une couche à l'autre.
- La norme L1 des cartes d'attention : ne semble pas porter d'information sur la beauté.

**Effet de la contrainte :** *Sur vgg19 avec 5 couches au décodeur* : la présence de la contrainte à l'entraînement semble réduire la corrélation entre gini et les scores de beauté.

*Sur le petit modèle* : La contrainte change la corrélation totalement. Il y a plus de corrélations avec que sans la contrainte, mais avec la contrainte, les corrélations sont négatives, ce qui est l'inverse de notre attente. Nous n'avons pas entraîné le modèle VGG19 avec 10 couches au décodeur avec la contrainte. Effet des différents modèles : Le vgg à 10 couches au décodeur semble se démarquer des autres dans ses motifs de corrélation. Peut-être que c'est causé par l'ajout du mécanisme d'attention.

### 9.1.2 Modèles linéaires

#### Méthode

Les corrélations nous permettent de savoir si nos métriques et la beauté ont tendance à varier ensemble. Néanmoins, elles ne nous permettent pas directement de vérifier leur capacité à prédire les scores de beauté. Il serait en effet intéressant de savoir, parmi les variations entre les scores de beauté pour différentes images, quelle proportion pourrait être prédictive grâce à nos métriques. Pour cela, nous utilisons des modèles linéaires.

Un modèle linéaire est un ensemble de poids qui constituent une combinaison linéaire de vecteurs dont le résultat se rapproche autant que possible d'un vecteur cible. Les vecteurs dans la combinaison sont formés des échantillons de nos *variables prédictives* (ou indépendantes) et le vecteur cible est formé des échantillons de notre *variable cible* (ou variable dépendante). Sous forme plus compacte, un modèle linéaire est constitué d'un ensemble de poids  $\beta_1, \beta_2, \dots, \beta_n$  tels que la distance entre

$$\beta_1 * V_1 + \beta_2 * V_2 + \dots + \beta_n \text{ et } V_c$$

sera minimale (où  $V_c$  est le vecteur cible, et  $V_1, V_2, \dots, V_{n-1}$  sont les vecteurs prédictifs).

Si  $\hat{V}_c$  est le vecteur produit par notre modèle linéaire, nous mesurons la force prédictive de notre modèle linéaire comme suit :

- $SR = \sum_i (\hat{V}_c - V_c)^2$  est une mesure de la variabilité entre les prédictions et le vrai vecteur cible.
- $SS = \sum_i (V_c - \bar{V}_c)^2$  est une mesure de la variabilité au sein du vrai vecteur cible.
- $\frac{SR}{SS}$  est alors la proportion de variabilité entre les scores de beauté qui n'est pas expliquée par les prédictions.
- $R^2 = 1 - \frac{SR}{SS}$  est finalement la proportion de variabilité entre les scores de beauté qui est expliquée par les prédictions.

Nous utilisons ce score  $R^2$  pour mesurer la capacité des métriques à expliquer la beauté.

Les métriques qui nous intéressent ici sont celles qui peuvent être extraites par couche : il n'est pas très intéressant de faire un modèle linéaire sur juste l'erreur de reconstruction, ou juste l'acuité SAM. Ainsi, nous entraînons nos modèles linéaires sur les métriques Gini,  $L_\epsilon$  et la norme L1 des cartes d'attention.

#### Attentes

Si nos métriques sont de bons prédicteurs de la beauté, ce qui irait dans le sens de l'hypothèse de la théorie de la Fluence, nous nous attendrions à avoir des  $R^2$  plus hauts. Par exemple, dans [43], un score  $R^2$  de 0.17 a été trouvé entre des scores de beauté et une mesure de fluence. Dans le travail de Melvin BARDIN (stage de l'année dernière), sur différentes sous-classes de CFD, les scores variaient de 0.04 à 0.3.

#### Résultats

On peut voir à la figure 52, pour le petit modèle, les scores moyens sur 10 modèles linéaires entraînés pour différentes versions du petit modèle, et les métriques L-epsilon et gini (il y a donc 5 vecteurs indépendants, pour les 5 couches de l'encodeur).

Pour les métriques d'attention, lorsqu'on entraîne le modèle VGG19 avec 10 couches au décodeur, et des cartes d'attention après chaque couche entre la couche 4 et 9 de l'encodeur, et qu'on les utilise comme vecteurs indépendants dans notre modèle linéaire, on trouve un  $R^2$  d'environ 0.108 avec la première version de l'attention qui n'a que l'attention en profondeur et une contrainte L1 et 0.0942 avec la seconde qui avait un softmax.

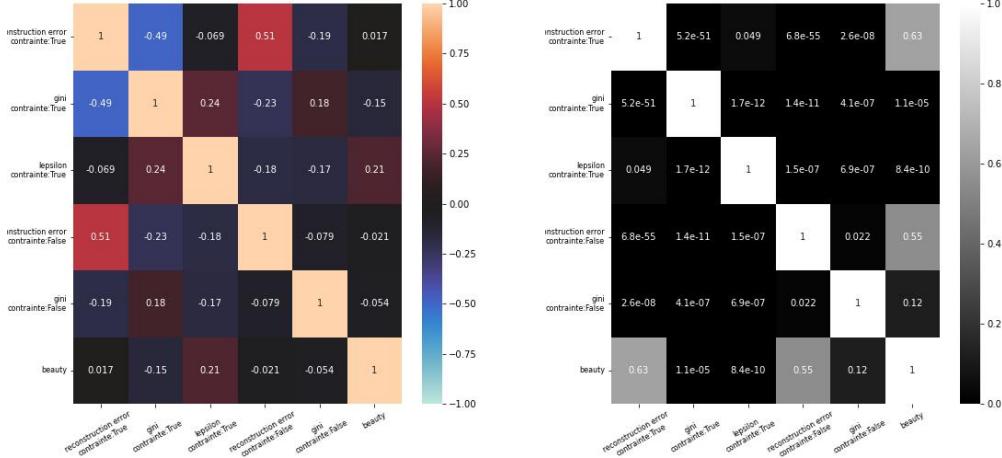


Figure 40: Pour chaque matrice de corrélation, nous regardons aussi la matrice des Pvalues pour s'assurer que les résultats sont significatifs.

### Interprétation

Il semblerait que nos métriques ne contiennent pas beaucoup d'information sur les scores de beauté. Plusieurs hypothèses peuvent être émises en conséquence :

- Les réseaux n'étaient pas des bons modèles du cerveau,
- Les métriques ne sont pas avisées comme mesures de la fluence,
- Les bases de données ne mesurent pas le type de beauté qui serait prédit par la fluence,
- La fluence prédit environ 10 pourcent de la beauté en général.

## Conclusion

### Conclusion sur les résultats

Pour conclure, dans ce stage nous avons établi une méthodologie pour étudier le lien entre des métriques de fluence et des scores de beauté, à travers des autoencodeurs comme modèles du cerveau. Les tâches principales étaient le développement de 5 métriques différentes, le test de l'utilisation d'une contrainte de sparsité, la tentative de nous passer du décodeur, la création et l'entraînement d'une nouvelle architecture de VAE, et l'analyse statistique. L'apport du stage est d'abord méthodologique, étant donnés les résultats statistiques : nous n'avons pas trouvé de motif très clair malgré des corrélations avec pvalue inférieure à 01, mais l'étude n'a pas été assez exhaustive pour rejeter l'hypothèse. Ainsi, la méthode développée sera ré-utilisée dans le futur et l'objectif du dernier mois de stage est de développer, en collaboration avec des personnes qui pourraient l'utiliser, une interface pratique pour l'entraînement des modèles sur de nouvelles bases de données, et l'extraction des métriques. Il serait par exemple intéressant de faire des vidéos pour montrer l'utilisation de l'interface, ou encore ajouter une documentation avec l'outil Sphinx.

### Capacités développées

- Plus d'expérience dans l'implémentation et entraînement en Pytorch
- Meilleure maîtrise du côté théorique du Deep Learning
- Introduction à une grande quantité de techniques de DL jusqu'ici inconnues

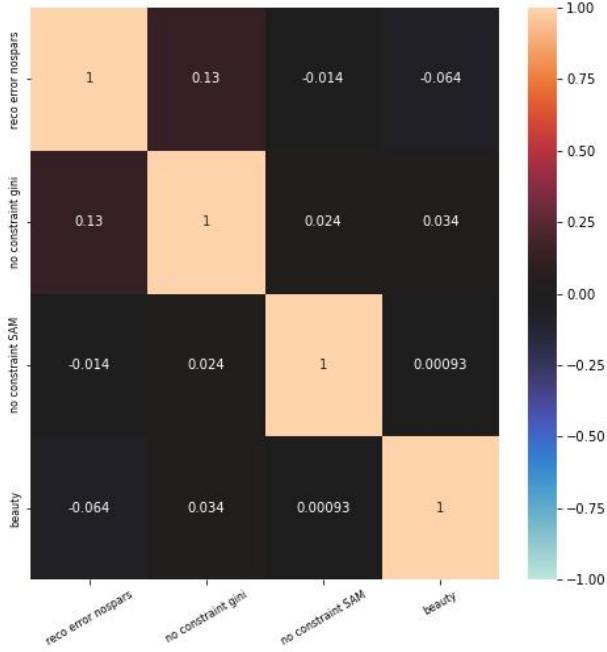


Figure 41: Corrélation de la beauté (dernière ligne de la matrice) avec la métrique SAM et gini sur toutes les couches du petit modèle. Les corrélations sont presque nulles, il n'y a pas de lien entre ces métriques et le score de beauté.

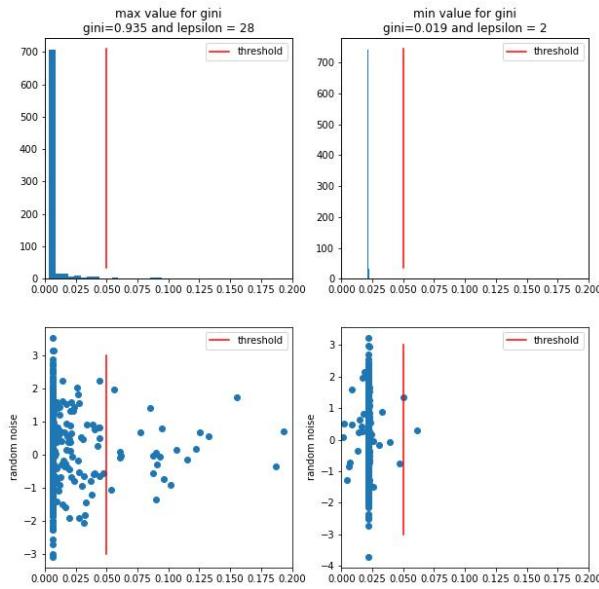


Figure 42: Comportement contre-intuitif des métriques gini et  $L_\epsilon$  : dans certains cas de figure, gini et  $L_\epsilon$  peuvent être positivement corrélées, ce qui est contre-intuitif.

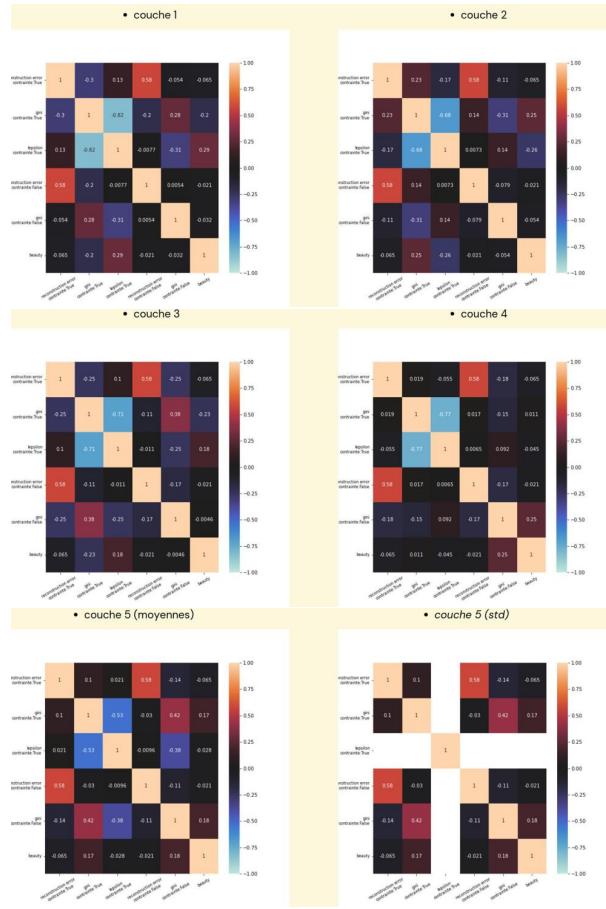


Figure 43: Corrélation de pearson entre les métriques de fluence, et les scores de beauté sur CFD, pour le petit vae avec 5 layers au décodeur. Nous donnons les corrélations pour les métriques sur un modèle entraîné avec et sans la contrainte. Nous observons une alternance de corrélations sur les premières couches, ce qui n'est pas très interprétable. Autrement, nous n'avons pas de corrélation aux scores de beauté. La contrainte semble faire ressortir des corrélations avec une plus haute magnitude, peut-être parce qu'elle renforce le modèle du cerveau.

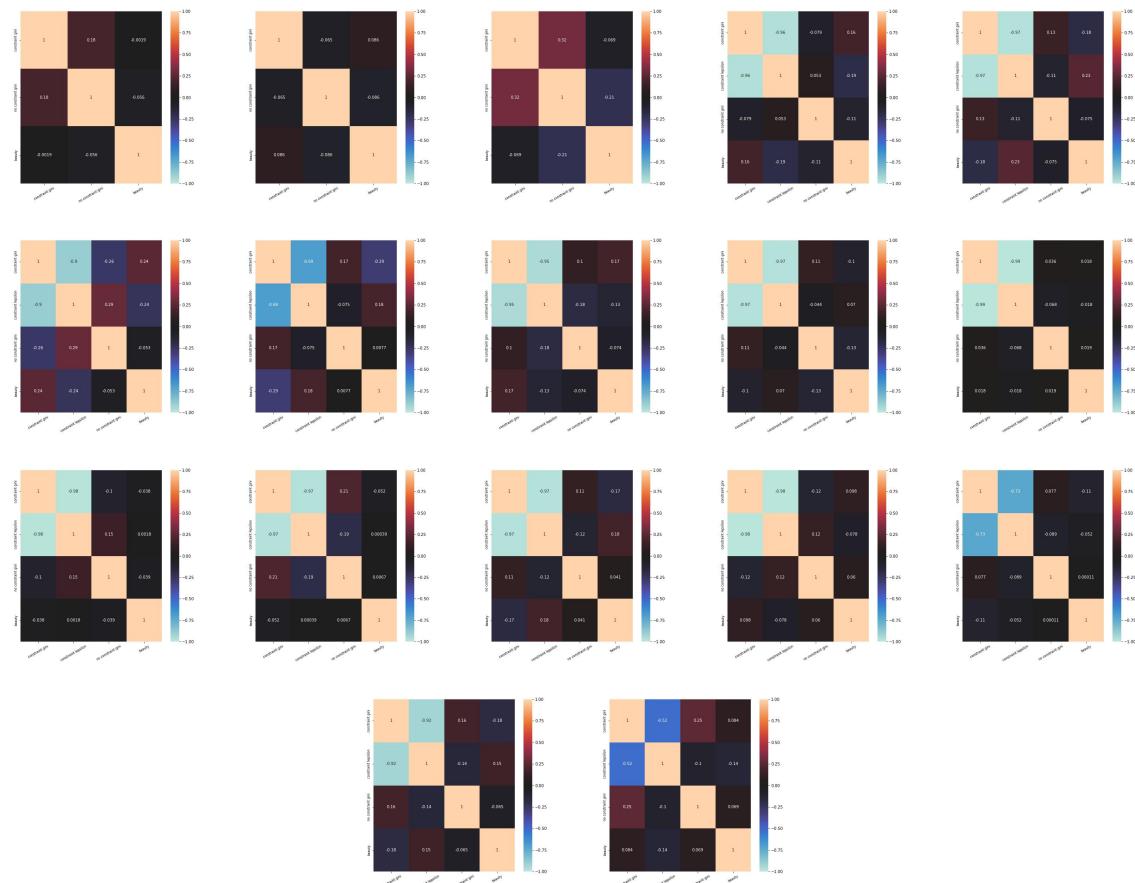


Figure 44: Corrélations entre les métriques de fluence, et les scores de beauté sur CFD, pour le vgg avec 5 layers au décodeur. Nous constatons la même oscillation des scores de corrélation, entourée de corrélations nulles.

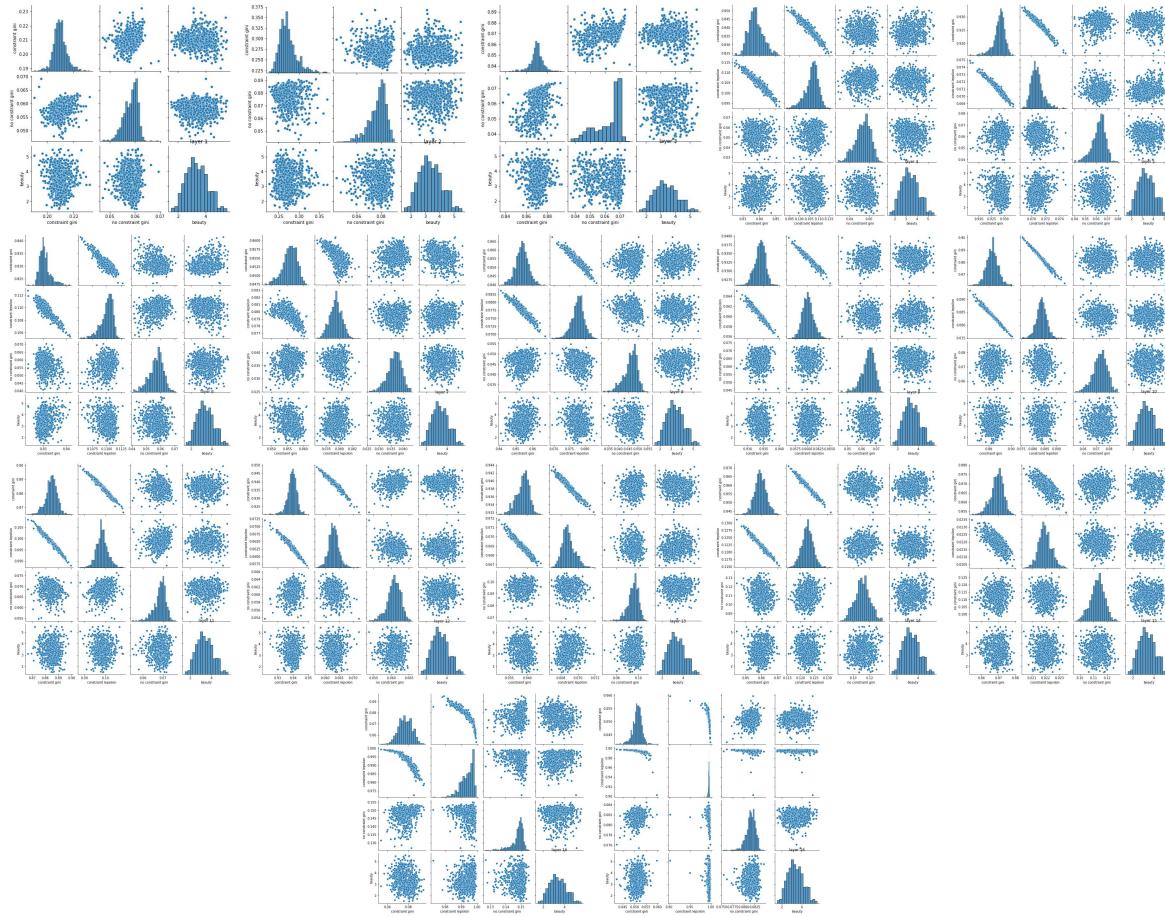


Figure 45: distributions jointes entre les métriques de fluence, et les scores de beauté sur CFD, pour le vgg vae avec 5 layers au décodeur (pour vérifier qu'il n'y a pas de relations non-linéaires)

$R^2$ ridge regression	sparsité partout	sparsité partout sauf convl	sparsité partout sauf convl et moyennes	pas de sparsité
lepsilon	0.106	0.103	0.0538	intolérable
gini	0.0740	0.0759	0.0925	0.0948

Figure 46: Résultats des régressions linéaires avec normalisation l2 (moyenne du  $r^2$  sur 10 modèles), pour le petit modèle entraîné avec la contrainte à différents endroits.

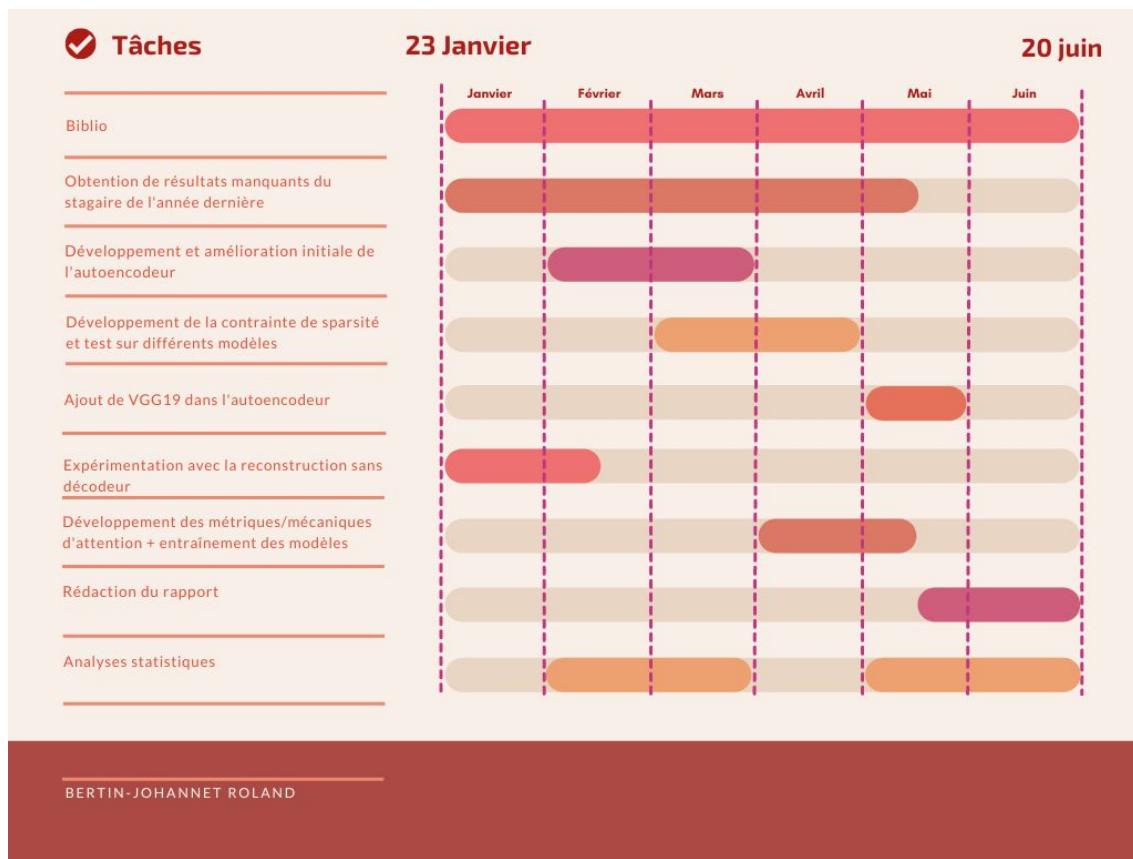


Figure 47: Allocation de temps aux différentes tâches du stage.

- Prise en connaissance d'une partie de la bibliographie dans le domaine des neurosciences computationnelles
- Eveil à plusieurs théories de l'esthétique
- Pratique la méthode scientifique
- Application de statistiques pour répondre à de nouvelles questions
- Collaboration et discussion avec des profils variés (Mathématiciens, écologues, neuroscientifiques, spécialistes de Deep Learning)..

## Perspectives futures pour ce sujet

Il serait intéressant dans le futur, même si cela sort du sujet du stage, d'ajouter une dimension temporelle à nos métriques. Cela pourrait amener à une attention plus réaliste et plus biologiquement poussée et pourrait être réalisé avec du reinforcement learning

## References

- [1] Horace B Barlow et al. Possible principles underlying the transformation of sensory messages. *Sensory communication*, 1(01):217–233, 1961.
- [2] Aenne A Briellmann and Peter Dayan. A computational model of aesthetic value. *Psychological Review*, 2022.
- [3] Rewon Child. Very deep vaes generalize autoregressive models and can outperform them on images. *arXiv preprint arXiv:2011.10650*, 2020.
- [4] Denis Dutton. *The art instinct: Beauty, pleasure, & human evolution*. Oxford University Press, USA, 2009.
- [5] Ronald A Fisher. The evolution of sexual preference. *The Eugenics Review*, 7(3):184, 1915.
- [6] Griffin Floto, Stefan Kremer, and Mihai Nica. The tilted variational autoencoder: Improving out-of-distribution detection. In *The Eleventh International Conference on Learning Representations*, 2023.
- [7] Pierre Foret, Ariel Kleiner, Hossein Mobahi, and Behnam Neyshabur. Sharpness-aware minimization for efficiently improving generalization. *arXiv preprint arXiv:2010.01412*, 2020.
- [8] Kunihiko Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological cybernetics*, 36(4):193–202, 1980.
- [9] Ian J. Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks, 2014.
- [10] Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.
- [11] Laura K. M. Graf and Jan R. Landwehr. A dual-process perspective on fluency-based aesthetics: the pleasure-interest model of aesthetic liking. *Personality and Social Psychology Review: An Official Journal of the Society for Personality and Social Psychology, Inc*, 19(4):395–410, November 2015.
- [12] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.
- [13] Zhenliang He, Wangmeng Zuo, Meina Kan, Shiguang Shan, and Xilin Chen. Attgan: Facial attribute editing by only changing what you want. *IEEE transactions on image processing*, 28(11):5464–5478, 2019.

- [14] Dan Hendrycks, Kevin Zhao, Steven Basart, Jacob Steinhardt, and Dawn Song. Natural adversarial examples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15262–15271, 2021.
- [15] Alexander Hepburn, Valero Laparra, Raul Santos-Rodriguez, Johannes Ballé, and Jesús Malo. On the relation between statistical learning and perceptual distances, 2022.
- [16] Xianxu Hou, Linlin Shen, Ke Sun, and Guoping Qiu. Deep feature consistent variational autoencoder. In *2017 IEEE winter conference on applications of computer vision (WACV)*, pages 1133–1141. IEEE, 2017.
- [17] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861*, 2017.
- [18] David H Hubel and Torsten N Wiesel. Receptive fields of single neurones in the cat’s striate cortex. *The Journal of physiology*, 148(3):574, 1959.
- [19] N. Hurley and S. Rickard. Comparing Measures of Sparsity. *IEEE Transactions on Information Theory*, 55(10):4723–4741, October 2009.
- [20] Andrew Ilyas, Shibani Santurkar, Dimitris Tsipras, Logan Engstrom, Brandon Tran, and Aleksander Madry. Adversarial examples are not bugs, they are features, 2019.
- [21] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. pmlr, 2015.
- [22] Kimmo Karkkainen and Jungseock Joo. Fairface: Face attribute dataset for balanced race, gender, and age for bias measurement and mitigation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1548–1558, 2021.
- [23] Diederik P Kingma and Max Welling. Auto-encoding variational bayes, 2022.
- [24] Durk P Kingma and Prafulla Dhariwal. Glow: Generative flow with invertible 1x1 convolutions. *Advances in neural information processing systems*, 31, 2018.
- [25] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60(6):84–90, 2017.
- [26] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.
- [27] Grace W Lindsay. Attention in psychology, neuroscience, and machine learning. *Frontiers in computational neuroscience*, 14:29, 2020.
- [28] Grace W Lindsay. Convolutional neural networks as a model of the visual system: Past, present, and future. *Journal of cognitive neuroscience*, 33(10):2017–2031, 2021.
- [29] Christos Louizos, Max Welling, and Diederik P Kingma. Learning sparse neural networks through  $l_0$  regularization. *arXiv preprint arXiv:1712.01312*, 2017.
- [30] Debbie S. Ma, Joshua Correll, and Bernd Wittenbrink. The Chicago face database: A free stimulus set of faces and norming data. *Behavior Research Methods*, 47(4):1122–1135, December 2015.
- [31] Lars Maaløe, Marco Fraccaro, Valentin Liévin, and Ole Winther. Biva: A very deep hierarchy of latent variables for generative modeling. *Advances in neural information processing systems*, 32, 2019.
- [32] Beren Millidge, Anil Seth, and Christopher L Buckley. Predictive coding: a theoretical and experimental review, 2022.
- [33] Alexander Mordvintsev, Christopher Olah, and Mike Tyka. Inceptionism: Going deeper into neural networks, 2015.

- [34] Augustus Odena, Vincent Dumoulin, and Chris Olah. Deconvolution and checkerboard artifacts. *Distill*, 2016.
- [35] Bruno A Olshausen and David J Field. Emergence of simple-cell receptive field properties by learning a sparse code for natural images. *Nature*, 381(6583):607–609, 1996.
- [36] Konstantinos P. Panousis, Sotirios Chatzis, and Sergios Theodoridis. Stochastic local winner-takes-all networks enable profound adversarial robustness, 2021.
- [37] Johanna Pasquet, Jérôme Pasquet, Marc Chaumont, and Dominique Fouchez. Pelican: deep architecture for the light curve analysis. *Astronomy & Astrophysics*, 627:A21, 2019.
- [38] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga, and Adam Lerer. Automatic differentiation in pytorch, 2017.
- [39] Richard O Prum. *The evolution of beauty: How Darwin’s forgotten theory of mate choice shapes the animal world-and us.* Anchor, 2018.
- [40] Xuming Ran, Mingkun Xu, Lingrui Mei, Qi Xu, and Quanying Liu. Detecting out-of-distribution samples via variational auto-encoder with reliable uncertainty estimation. *Neural Networks*, 145:199–208, 2022.
- [41] Rolf Reber, Norbert Schwarz, and Piotr Winkielman. Processing Fluency and Aesthetic Pleasure: Is Beauty in the Perceiver’s Processing Experience? *Personality and Social Psychology Review*, 8(4):364–382, November 2004.
- [42] Julien P Renoult. The evolution of aesthetics: A review of models. *Aesthetics and Neuroscience: Scientific and Artistic Perspectives*, pages 271–299, 2016.
- [43] Julien P. Renoult, Jeanne Bovet, and Michel Raymond. Beauty is in the efficient coding of the beholder. *Royal Society Open Science*, 3(3):160027, March 2016.
- [44] Julien P Renoult and Tamra C Mendelson. Processing bias: extending sensory drive to include efficacy and efficiency in information processing. *Proceedings of the Royal Society B*, 286(1900):20190165, 2019.
- [45] Michael J Ryan. A taste for the beautiful. In *A Taste for the Beautiful*. Princeton University Press, 2018.
- [46] Elham Saraee, Mona Jalal, and Margrit Betke. Visual complexity analysis using deep intermediate-layer features. *Computer Vision and Image Understanding*, 195:102949, 2020.
- [47] Jean-Marie Schaeffer. *L’expérience esthétique*. Editions Gallimard, March 2015. Google-Books-ID: JsDmBgAAQBAJ.
- [48] Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P. Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network, 2016.
- [49] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition, April 2015. arXiv:1409.1556 [cs].
- [50] Jiaming Song and Stefano Ermon. Understanding the limitations of variational mutual information estimators. *arXiv preprint arXiv:1910.06222*, 2019.
- [51] Michele Svanera, Andrew T Morgan, Lucy S Petro, and Lars Muckli. A self-supervised deep neural network for image completion resembles early visual cortex fmri activity patterns for occluded scenes. *Journal of Vision*, 21(7):5–5, 2021.
- [52] Dmitry Ulyanov, Andrea Vedaldi, and Victor Lempitsky. Instance normalization: The missing ingredient for fast stylization. *arXiv preprint arXiv:1607.08022*, 2016.

- [53] Arash Vahdat and Jan Kautz. Nvae: A deep hierarchical variational autoencoder. *Advances in neural information processing systems*, 33:19667–19679, 2020.
- [54] Rufin VanRullen and Andrea Alamia. Gattanet: Global attention agreement for convolutional neural networks. In *Artificial Neural Networks and Machine Learning–ICANN 2021: 30th International Conference on Artificial Neural Networks, Bratislava, Slovakia, September 14–17, 2021, Proceedings, Part I*, pages 281–293. Springer, 2021.
- [55] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.
- [56] Piotr Winkielman, Jamin Halberstadt, Tedra Fazendeiro, and Steve Catty. Prototypes are attractive because they are easy on the mind. *Psychological Science*, 17(9):799–806, September 2006.
- [57] Piotr Winkielman, Norbert Schwarz, Tedra Fazendeiro, and Rolf Reber. The Hedonic marking of Processing Fluency: implications for evaluative judgment. In *The Psychology of Evaluation: Affective Processes in Cognition and Emotion*. Psychology Press, January 2003.
- [58] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. Cbam: Convolutional block attention module. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018.
- [59] Richard Zhang, Phillip Isola, Alexei A Efros, Eli Shechtman, and Oliver Wang. The unreasonable effectiveness of deep features as a perceptual metric. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 586–595, 2018.