

UNIVERSITÉ DE PARIS  
UFR MATHÉMATIQUES ET INFORMATIQUE

---

# Projet TER - StyleGANov'

---

Master 1 Vision Machine Intelligente

Jiixin XUE – Billal IHADDADEN  
Encadré par Olivier RISSER-MAROIX

Année universitaire 2020-2021

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>État de l’art</b>	<b>2</b>
2.1	La similarité visuelle en science cognitives . . . . .	2
2.2	La similarité visuelle et l’apprentissage profond . . . . .	2
2.3	Les GANs et le transfert du style . . . . .	3
<b>3</b>	<b>Méthodes étudiées</b>	<b>4</b>
3.1	Generative Adversarial Network . . . . .	4
3.2	Pix2Pix . . . . .	4
3.3	BigGAN et espace de latent . . . . .	4
<b>4</b>	<b>Méthodes expérimentales</b>	<b>5</b>
<b>5</b>	<b>Conclusion et Perspectives</b>	<b>6</b>
<b>6</b>	<b>Rédaction en L<sup>A</sup>T<sub>E</sub>X</b>	<b>7</b>
6.1	Découpage d’un chapitre . . . . .	7
6.2	Listes . . . . .	8
6.2.1	Listes à puces . . . . .	8
6.2.2	Listes numérotées . . . . .	8
6.2.3	Listes descriptives . . . . .	8
6.2.4	Listes imbriquées . . . . .	8
6.3	Mathématiques . . . . .	9
6.4	Figures . . . . .	10
6.4.1	Inclusion d’images . . . . .	10
6.4.2	Dessin . . . . .	10
6.5	Tables . . . . .	10
6.6	Théorèmes . . . . .	11
6.7	Algorithmes et code . . . . .	11
6.7.1	Algorithmes en pseudo-code . . . . .	11
6.7.2	Listings : extraits de code source . . . . .	12
6.8	Bibliographie . . . . .	13
<b>7</b>	<b>Conclusion</b>	<b>14</b>
<b>A</b>	<b>Tables</b>	<b>15</b>

# Chapitre 1

## Introduction

Le jugement de la similarité d'images par l'humain s'appuie sur beaucoup de choses notamment les éléments de la scène et les aspects culturels. Pour nous les humains il nous est facile de juger de la similarité entre deux images, cependant la prédiction de la similarité perceptive humaine est un sujet de recherche difficile. Le processus visuel sous-jacent à cet aspect de la vision humaine fait appel à plusieurs niveaux différents d'analyse visuelle (formes, objets, texture, disposition, couleur etc). Dans le cas de ce projet la similarité purement visuelle sera traitée sans prendre en compte la sémantique. Le transfert de style et la génération d'image par des architectures modernes de réseaux de neurones (VAE, AdaIN, PIX2PIX, CycleGAN, BigGAN etc). La première partie consiste à expérimenter la génération d'image avec pix2pix en utilisant les images de l'ensemble de données PASCAL avec et sans l'arrière-plan. Ensuite Nous allons expérimenter le BigGAN pré entraîné sur l'ensemble de données d'ImageNET.

# Chapitre 2

## État de l'art

Un chapitre dédié à l'état de l'art doit décrire les concepts et méthodes déjà existant(e)s, en lien avec le travail que vous avez réalisé.

### 2.1 La similarité visuelle en science cognitives

Les objets peuvent être caractérisés selon un grand nombre de critères possibles, mais certaines caractéristiques sont plus utiles que d'autres pour donner un sens aux objets qui nous entourent. Dans [5] ils ont développé un modèle informatique de jugements de similarité pour les images du monde réel de 1854 objets.

### 2.2 La similarité visuelle et l'apprentissage profond

Les progrès récents des réseaux de neurones artificiels ont révolutionné la vision par ordinateur, mais ces systèmes de conception sont toujours surpassés par les humains, dans [1] ils ont comparé la perception des objets par le cerveau humain et par les machines (certain nombre de modèles informatiques, par exemple : Tal, Gabor, Hog, split-half etc.). Ils ont recueilli un vaste ensemble de données comprenant 26 675 mesures de dissimilarité perçue pour 2801 objets visuels chez 269 sujets humains. Afin de mesurer la dissimilarité chez les humains, ils ont demandé de localiser une bille étrange dans un tableau contenant un objet parmi de multiples occurrences de l'autre. La réciproque du temps de recherche visuelle a été considérée comme une estimation de la dissimilarité perçue. Cette mesure se comporte comme une distance mathématique, elle présente une somme linéaire de plusieurs caractéristiques, elle explique la catégorisation visuelle rapide et est fortement corrélée avec les évaluations subjectives de la dissimilarité. Ils ont testé l'ensemble de mesures sur des modèles de calcul très répandus. Le meilleur modèle était un CNN mais il a été surpassé par la combinaison de tous les autres modèles. Leur conclusion était que tous les modèles informatiques montrent des modèles similaires de décart par rapport à la perception humaine.

Dans [2] ils voulaient savoir est-ce que l'apprentissage automatique peut expliquer les jugements humains de similarité de forme des objets visuels. Alors ils ont analysé la performance des systèmes d'apprentissage métrique (distance ou similarité) y compris les DNN, sur un nouvel ensemble de données de jugement de similarité de forme des objets visuels humains.

Contrairement aux autres études où ils demandaient aux participants de juger de la similarité lorsque les objets ou les scènes étaient rendus à partir d'un seul point de vue, eux ils ont utilisé un rendu à partir de plusieurs vues et ils ont demandé aux participants de juger de la similarité de forme de manière variable. Ils ont trouvé que les DNN ne parviennent pas à expliquer les données expérimentales, mais une métrique entraînée avec une représentation variable basée sur des parties produit un bon ajustement, ils ont aussi constaté que même si les DNN puissent apprendre à extraire la représentation basée sur les parties et devrait être capable d'apprendre à modéliser leurs données. Les réseaux entraînés avec une fonction triplet loss basée sur le jugement de similarité ne donne pas un bon résultat. Le mauvais résultat du DNN est causé par la non-convexité du problème d'optimisation dans l'espace des poids du réseau. Ils concluent que l'insensibilité du point de vue est un aspect critique de la perception de la forme visuelle humaine, et que les réseaux de neurones et d'autres méthodes d'apprentissage automatique devront apprendre des représentations insensibles au point de vue afin de rendre compte des jugements de similarité de forme des objets visuels des humains.

La comparaison des représentations formées par les DNN avec celles utilisées par les humains est un défi, car les représentations psychologiques humaines ne peuvent pas être observées directement. Dans [3] ils ont évalué et proposé une amélioration de la correspondance entre les DNN et les représentations humaines. Leur approche consiste à résoudre le problème de comparaison en exploitant la relation étroite entre représentation et similarité, ce qui veut dire que pour chaque fonction de similarité sur un ensemble de paires de points de données correspond à une représentation implicite de ces points. Ce qui offre une base empirique pour la première évaluation de DNN en tant qu'une approximation des représentations psychologiques humaines.

Dans [4] ils ont démontré leur méthode sur le jeu de données CUB-200-2011 et Stanford Cars en appliquant leur architecture du DNN ProtoPNet. Leur expérience a montré que l'architecture de DNN qu'ils ont créé pouvait atteindre une précision comparable à avec ses analogues. Et lorsque plusieurs ProtoPNet sont combinés en un réseau plus vaste, celui-ci peut atteindre une précision équivalente à celle de certains des modèles profonds les plus performants. De plus, leur modèle offre un niveau d'interprétabilité qui est absent dans les modèles existants.

## 2.3 Les GANs et le transfert du style

# Chapitre 3

## Méthodes étudiées

Depuis que Ian Goodfellow a proposé le GAN (Generative Adversarial Network) en 2014, la recherche sur le GAN est très active. Diverses variantes du GAN ne cessent d'apparaître. Yann LeCun a même déclaré que le GAN était adversarial training is the coolest thing since sliced bread. Nous nous donc focaliserons dans un sur la génération de paires d'images visuellement similaires via des GANs.

### 3.1 Generative Adversarial Network

Generative Adversarial Network (GAN) est une méthode d'apprentissage non supervisé, consistant à faire jouer deux réseaux neuronaux l'un contre l'autre. Cette méthode a été proposée en 2014 par Ian Goodfellow[[goodfellow2014generative](#)]. Les réseaux adversariens génératifs se composent d'un réseau génératif et d'un réseau discriminatif. Le réseau génératif prend en entrée des échantillons aléatoires de l'espace latent, et sa sortie doit imiter autant que possible les échantillons réels de l'ensemble d'apprentissage. L'entrée du réseau discriminant est l'échantillon réel ou la sortie du réseau génératif, et son but est de distinguer autant que possible la sortie du réseau génératif de l'échantillon réel. Le réseau génératif, quant à lui, doit tromper le réseau discriminant autant que possible. Les deux réseaux jouent l'un contre l'autre et ajustent constamment leurs paramètres, dans le but ultime de rendre le réseau discriminant incapable de déterminer si la sortie du réseau génératif est vraie.

### 3.2 Pix2Pix

### 3.3 BigGAN et espace de latent

# Chapitre 4

## Méthodes expérimentales

# Chapitre 5

## Conclusion et Perspectives

Ce chapitre doit présenter les concepts et méthodes que vous avez étudié(e)s, et décrire vos résultats. En particulier, dans le cadre d'une étude expérimentale, discutez vos résultats et comparez, le cas échéant, votre approche avec l'état de l'art.



# Chapitre 6

## Rédaction en L<sup>A</sup>T<sub>E</sub>X

Dans ce chapitre, nous décrivons quelques bases sur l'utilisation de L<sup>A</sup>T<sub>E</sub>X pour la rédaction de votre rapport. Si vous êtes débutant en L<sup>A</sup>T<sub>E</sub>X, de nombreux tutoriaux sont disponibles en ligne (notamment [ce tutorial disponible sur Overleaf](#)). De nombreux symboles mathématiques sont disponibles L<sup>A</sup>T<sub>E</sub>X. Certains sont présentés dans ce document, nous vous recommandons également d'utiliser « The Comprehensive L<sup>A</sup>T<sub>E</sub>X Symbol List » [[pakin2020comprehensive](#)] pour trouver les symboles dont vous avez besoin.

Ce template est basé sur [Sleek Template](#), un template L<sup>A</sup>T<sub>E</sub>X libre qui fournit un certain nombre de commandes pour faciliter la rédaction d'un rapport. Nous décrivons ici une partie qui doit être suffisante, mais vous pouvez aussi vous référer à la documentation en ligne pour découvrir plus de commandes.

Le format de la page de garde est généré automatiquement, en fonction de la valeur placée dans les champs suivants : `\logo`, `\institute`, `\faculty`, `\department`, `\title`, `\subtitle`, `\author`, et `\date`.

Pensez à modifier `\title` et `\author`, et à choisir le bon `\subtitle` en fonction de votre parcours (IAD ou VMA). Ne modifiez pas les autres champs.

---

```
\logo{./resources/img/up_maths-info.jpg}
\institute{Université de Paris}
\faculty{UFR Mathématiques et Informatique}
\title{Titre du rapport}
\subtitle{Master 1 Intelligence Artificielle Distribuée}
%\subtitle{Master 1 Vision et Machine Intelligente}
\author{Prénom \textsc{Nom1} -- Prénom \textsc{Nom2}}
\date{2021}
```

---

Listing 6.1 – Code L<sup>A</sup>T<sub>E</sub>X de la page de garde

### 6.1 Découpage d'un chapitre

Au sein d'un chapitre, les commandes `\section`, `\subsection` et `subsubsection`, notamment, permettent d'organiser le contenu. De nombreux éléments (notamment les sections, les algorithmes, les définitions,...) peuvent être étiquetés au moyen de la commande

`\label`. Cela permet d’y faire référence, par exemple `\ref{section:liste}` permet de faire référence à la Section 6.2.

## 6.2 Listes

Cette section décrit comment réaliser des listes (et donne un exemple de section découpée en sous-sections). Pour chaque type de liste, les éléments sont identifiés par la commande `\item`. Le label peut être modifié, soit individuellement avec `\item[newLabel]`, soit pour tout l’environnement avec l’option `label=newLabel`.

### 6.2.1 Listes à puces

Une liste à puce est définie via l’environnement `itemize` :

- un item,
- un autre item,
- \* un dernier item, avec un label modifié.

### 6.2.2 Listes numérotées

Les listes numérotées sont définies avec `enumerate` :

- (a) un item,
- (b) un autre item,
- (c) un dernier item.

Dans ce cas, le nouveau label peut être une expression spéciale (*cf.* Table A.3 en Annexe), comme dans l’exemple précédent qui remplace la numérotation standard par « (a), (b), (c), ... ».

### 6.2.3 Listes descriptives

Une liste descriptive est définie par `description` :

**Item 1** Description de l’item 1,

**Item 2** Description de l’item 2,

**Item 3** Description de l’item 3.

### 6.2.4 Listes imbriquées

Voici un exemple d’imbrication de listes, qui inclut aussi des personnalisations, et des éléments de mathématiques (voir la Section 6.3).

It is also possible to write nested lists. Here follows a very condensed example.

- Lorem ipsum dolor sit amet, consectetur adipiscing elit, sed do eiusmod tempor incididunt ut labore et dolore magna aliqua.  
Arcu ac tortor dignissim convallis aenean et tortor. In eu mi bibendum neque egestas congue quisque.

- + Semper quis lectus nulla at volutpat diam ut. Felis eget velit aliquet sagittis id. Blandit aliquam etiam erat velit scelerisque in dictum non consectetur.

$$a^2 + b^2 = c^2 \tag{6.1}$$

- Nibh sed pulvinar proin gravida hendrerit lectus. Pretium aenean pharetra magna ac placerat vestibulum lectus mauris. Non consectetur a erat nam at lectus urna duis.
  - i. Nibh tortor id aliquet lectus. Sit amet justo donec enim diam vulputate ut pharetra sit.
  - iv. Condimentum id venenatis a condimentum vitae. Quis eleifend quam adipiscing vitae proin sagittis nisl.
  - xx. Proin sagittis nisl rhoncus mattis rhoncus urna neque viverra.
- Elit scelerisque mauris pellentesque pulvinar pellentesque habitant morbi tristique senectus.

**Ridiculus** mus mauris vitae ultricies leo. Mollis aliquam ut porttitor leo a diam. Velit egestas dui id ornare arcu odio ut sem nulla.

**Nullam vehicula** ipsum a arcu. Nibh sit amet commodo nulla facilisi nullam. At erat pellentesque adipiscing commodo elit. Libero volutpat sed cras ornare arcu dui.

## 6.3 Mathématiques

Ce template utilise les packages `amsmath` et `amssymb`, qui sont standard pour l'écriture d'éléments mathématiques en L<sup>A</sup>T<sub>E</sub>X. Quelques commandes personnalisées ont également été définies, comme `\rbk`, `\sbk` et `\cbk` pour les parenthèses, les crochets et les accolades, `\abs` pour la valeur absolue, `\norm` pour la norme, et `\fact` pour la factorielle. En voici des exemples

$$\left(\frac{\pi}{2}\right), \quad \left[\frac{\pi}{2}\right], \quad \left\{\frac{\pi}{2}\right\}, \quad \left|\frac{\pi}{2}\right|, \quad \left\|\frac{\pi}{2}\right\|, \quad n! = \prod_{i=1}^n i$$

On peut faire référence à une équation ou une formule mathématique particulière, comme dans l'exemple de l'équation [6.1](#).

Voici quelques exemples qui illustrent les possibilités du template.

D'abord, une équation non numérotée avec l'environnement `equation*` :

$$e = \sum_{n=0}^{\infty} \frac{1}{n!}$$

Deux sous-équations avec leur propre label ([6.2a](#) et [6.2b](#)) :

$$\frac{dx}{dt} = \alpha x - \beta xy \tag{6.2a}$$

$$\frac{dy}{dt} = \delta xy - \gamma y \tag{6.2b}$$

Un système d'équations :

$$\begin{cases} x = 2 \times y + z \\ y = 3 \times x^2 - 2 \times z \\ z = x - y \end{cases} \quad (6.3)$$

## 6.4 Figures

### 6.4.1 Inclusion d'images

Les figures, aux formats classiques d'images, peuvent être incluses avec `\includegraphics{}`, notamment `jpg`, `png`, `bmp`,... Cependant, il est préférable d'utiliser un format vectoriel quand c'est possible (`pdf` or `eps`).



FIGURE 6.1 – Logo de l'UFR Maths-Info

### 6.4.2 Dessin

Pour les plus courageux, il est possible de dessiner directement, notamment avec le package `tikz`. Voici un exemple de graphe dirigé :

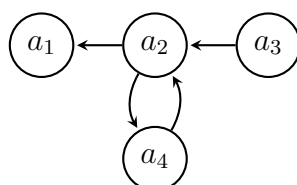


FIGURE 6.2 – Un graphe dirigé

Voir [TikZ pour l' impatient](#) pour plus de détails.

## 6.5 Tables

Voici des exemples de tableaux :

a			qrs
b	ef	jkl	tuvx
cd	ghi	mnp	wyz

TABLE 6.1 – Exemple tableau avec une cellule sur plusieurs colonnes

a	b	c
	de	fg
	hij	klm
nopq	rstuv	wxyz

TABLE 6.2 – Exemple tableau avec une cellule sur plusieurs lignes

## 6.6 Théorèmes

Le template fournit plusieurs environnements de type « théorème », notamment `thm` (théorème), `lem` (lemme), `prop` (proposition), `proof` (démonstration), `defn` (définition), ou `expl` (exemple).

**Théorème 1** (Inégalité triangulaire). *Étant donné un triangle dans un espace euclidien, la somme des longueurs de deux de ses côtés est supérieure ou égale à la longueur du troisième côté.*

*Démonstration.* Soient  $a$ ,  $b$  et  $c$  les longueurs des côtés d’un triangle dans un espace euclidien, et  $\alpha$ ,  $\beta$ ,  $\gamma$  leurs angles opposés respectifs. D’après le théorème de Pythagore généralisé, on a

$$\begin{aligned}
 c^2 &= a^2 + b^2 - 2ab \cos \gamma \\
 &\leq a^2 + b^2 + 2ab \\
 &\leq (a + b)^2 \\
 \Leftrightarrow c &\leq a + b
 \end{aligned}$$

Par conséquent, dans n’importe quel triangle, la somme des longueurs de deux côtés est toujours supérieure ou égale à la longueur du troisième côté. ■

**Définition 1** (Graphe dirigé). Un graphe dirigé est un couple  $G = \langle N, E \rangle$  où  $N$  est l’ensemble des noeuds, et  $E \subseteq N \times N$  est l’ensemble des arcs.

*Exemple 1.* Le graphe  $G = \langle N, E \rangle$  avec  $N = \{a_1, a_2, a_3, a_4\}$  et  $E = \{(a_2, a_1), (a_2, a_4), (a_3, a_2), (a_4, a_2)\}$  est montré en Figure 6.2.

## 6.7 Algorithmes et code

Il est possible d’afficher des algorithmes et extraits de code dans un document L<sup>A</sup>T<sub>E</sub>X.

### 6.7.1 Algorithmes en pseudo-code

Voici un exemple d’algorithme écrit avec le package `algorithm2e`, nous vous invitons à voir sa [documentation](#) pour plus d’informations. Il est possible de décrire le comportement

d'un algorithme en faisant référence aux lignes numérotées, par exemple la ligne 8.

```
1 Entrées une liste d'entiers  $l$ ;  
   Résultat : le plus grand élément de  $l$   
2  $tmpMax = l[0]$ ;  
3  $i = 1$ ;  
4 while  $i < longueur(l)$  do  
5   | if  $l[i] > tmpMax$  then  
6   |   |  $tmpMax = l[i]$ ;  
7   | end  
8   |  $i = i + 1$ ;  
9 end  
10 return  $tmpMax$ ;
```

**Algorithme 1** : Recherche de l'élément maximal d'une liste

### 6.7.2 Listings : extraits de code source

Le package `sleek-listings` permet d'afficher des extraits de code source, en respectant la coloration syntaxique pour différents langages : `c`, `cpp`, `matlab`, `python` and `java` are implemented, with basic color-maps. Il est possible de personnaliser le style des listings pour chacun de ces langages :

- `\NumberStyle{stylename}` crée un style `stylenameNumber` avec les lignes numérotées;
- `\FrameStyle{stylename}` crée un style `stylenameFrame` avec un cadre autour du listing;
- `\FrameTBStyle{stylename}` crée un style `stylenameFrameTB` avec une ligne au dessus et en dessous du listing;
- `\FrameNumberStyle{stylename}` et `\FrameTBNumberStyle{stylename}` combinent les lignes numérotées et le cadre/les lignes.

Par exemple, la commande `\FrameTBStyle{python}` crée un nouveau style appelé `pythonFrameTB`, qui donne le rendu suivant pour du code Python :

---

```
import numpy as np # Unnecessary import  
  
a, b = 69., .420  
  
def f(a: float, b: float) -> float:  
    r"""  
    Sum two numbers  
  
    Parameters  
    -----  
    a: first number  
    b: second number  
  
    Returns  
    -----  
    the sum of 'a' and 'b'  
    """  
  
    return a + b
```

```
c = f(a, b)
```

```
print('{:f} + {:f} equals {:f}'.format(a, b, c))
```

---

## 6.8 Bibliographie

Nous recommandons l'usage de [BibTeX](#) pour la gestion de la bibliographie. Ajoutez les entrées correspondant à vos références dans le fichier `resources/bib/references.bib`. Vous pouvez obtenir les entrées BibTeX sur [Google Scholar](#) ou [DBLP](#). La commande `\cite` vous permet de citer une (ou plusieurs) référence(s) dans le document, par exemple `[pakin2020comprehensive, RusselNorvig]`.

# Chapitre 7

## Conclusion

La conclusion de votre rapport doit brièvement résumer ce que vous avez fait, en mettant en lumière les points forts et les points faibles de votre travail. Enfin, il faut décrire les perspectives de poursuite qui peuvent être envisagées.



# Annexe A

## Tables

Les chapitres d'annexes sont situés après l'utilisatoin de la commande `\appendix`, et se structurent comme des chapitres normaux.

<code>\emph{abcABC123}</code>	<i>abcABC123</i>
<code>\bfseries{abcABC123}</code>	<b>abcABC123</b>
<code>\itshape{abcABC123}</code>	<i>abcABC123</i>
<code>\lowercase{abcABC123}</code>	abcab123
<code>\normalfont{abcABC123}</code>	abcABC123
<code>\textbf{abcABC123}</code>	<b>abcABC123</b>
<code>\textit{abcABC123}</code>	<i>abcABC123</i>
<code>\textsc{abcABC123}</code>	ABCABC123
<code>\textsf{abcABC123}</code>	abcABC123
<code>\textsl{abcABC123}</code>	<i>abcABC123</i>
<code>\textsubscript{abcABC123}</code>	abcABC123
<code>\textsuperscript{abcABC123}</code>	abcABC123
<code>\texttt{abcABC123}</code>	abcABC123
<code>\underline{abcABC123}</code>	<u>abcABC123</u>
<code>\uppercase{abcABC123}</code>	ABCABC123

TABLE A.1 – Available text fonts in L<sup>A</sup>T<sub>E</sub>X.

<code>\$\mathcal{abcABC123}\$</code>	$\mathcal{ABC}\infty\in\exists$
<code>\$\mathit{abcABC123}\$</code>	<i>abcABC123</i>
<code>\$\mathnormal{abcABC123}\$</code>	<i>abcABC123</i>
<code>\$\mathrm{abcABC123}\$</code>	abcABC123
<code>\$\mathbb{abcABC123}\$</code>	$\mathbb{ABC}\mathbb{K}\mathbb{H}$
<code>\$\mathfrak{abcABC123}\$</code>	$\mathfrak{abcABC123}$

TABLE A.2 – Available math fonts in L<sup>A</sup>T<sub>E</sub>X and AMS.

Expression	Description
<code>\arabic*</code>	Arabic numbers (1, 2, 3, ...)
<code>\alph*</code>	Lowercase letters (a, b, c, ...)
<code>\Alph*</code>	Uppercase letters (A, B, C, ...)
<code>\roman*</code>	Lowercase Roman numerals (i, ii, iii, ...)
<code>\Roman*</code>	Lowercase Roman numerals (I, II, III, ...)

TABLE A.3 – Special expressions for the label of `enumerate` environments.