Université de Caen Normandie
Master 2 Informatique - Images et données multimédia

Mémoire de stage

Génération de contenus contrastants pour améliorer l'accessibilité aux documents webs

---

Présenté et soutenu par
François Ledoyen

le 3 Septembre 2021

Stage réalisé du 3 Mars 2021 au 27 Août 2021
dans l'équipe IMAGE du laboratoire GREYC de Caen

Encadrants :

M. Youssef Chahir, GREYC, Université de Caen
M. Gaël Dias, GREYC, Université de Caen
M. Fabrice Maurel, GREYC, Université de Caen
M. Alexis Lechervy, GREYC, Université de Caen

Table des matières

I Introduction **3**

1 Contexte **3**

2 Plan du mémoire **5**

II Accessibilité et audit **6**

1 Notions d'accessibilité numérique **6**

2 Audit RGAA (Référentiel général d'amélioration de l'accessibilité) **7**
2.1 Présentation du RGAA 7
2.2 Réalisation d'un audit RGAA 7

III État de l'art **11**

1 Apprentissage supervisé **11**
1.1 Objectif d'un modèle supervisé 11
1.2 Régression et classification 11
1.3 Apprentissage d'un modèle supervisé 12

2 Architectures des réseaux de neurones **13**
2.1 Perceptron 13
2.2 Perceptron multi-couches (MLP) 13
2.3 Réseau de neurones convolutif (CNN) 14
2.4 Auto-encodeur (AE) 15
2.5 Réseau génératif adverse (GAN) 15

3 Modèles de manipulation sémantique **16**

IV Contributions **20**

1 Création d'un jeu de données pour la génération de contraste **20**
1.1 Choix de la base initiale 20
1.2 Construction du jeu de contraste 21

2 Modèle de génération de contraste **22**
2.1 Représentation des images 23
2.2 Génération de contenu 24

V Conclusion **26**

Première partie

Introduction

1 Contexte

Malgré l'augmentation de la place du numérique dans notre quotidien - médias, démarches administratives, divertissement, etc. - l'accès au web pour les personnes handicapées n'a pas suivi la même courbe. Il a même régressé. En 2018, 66% des personnes naviguant au lecteur d'écran considèrent que l'accès au web a stagné ou régressé [Fra18]. Mais heureusement, l'accessibilité numérique est normée par des règles inscrites dans le *web content accessibility guidelines* (WCAG) du W3C et dans le référentiel général d'amélioration de l'accessibilité (RGAA) édité par la direction interministérielle du numérique française.

La société Koena a pour rôle (1) d'auditer et de proposer des améliorations d'accessibilité à ces services qui trop souvent ne respectent pas les réglementations, et (2) de former à l'accessibilité numérique les personnes en charge de leurs développements. La tâche d'audit est essentiellement manuelle via l'utilisation de tableurs ou d'extensions de navigateurs, et nécessite un haut niveau d'expertise incluant notamment l'appréhension du contexte. Ainsi, seulement 30% des tests d'audit sont actuellement automatisés, car une grande partie de ceux restants dépendent de l'analyse du contexte dans lequel se trouve l'élément étudié. Cette prise en compte dans les outils d'audit automatique n'est pas encore traitée. De plus l'auditeur doit aussi proposer des corrections et améliorations du document qu'il évalue.

Au cours de ces six mois j'ai eu l'opportunité de suivre leur formation d'une semaine pour devenir auditeur RGAA ce qui m'a permis de mieux cerner les contraintes en lien avec l'automatisation pour en extraire des pistes de recherche qui seront approfondis lors de ma thèse CIFRE avec Koena. Voici deux critères du RGAA représentatifs des contraintes à l'automatisation citées précédemment et sur lesquels je vais articuler mes futurs travaux :

— **Critère 1.1** *Chaque image* ***porteuse d'information*** [1] *a-t-elle une alternative textuelle* [2] *?*

— **Critère 1.3** *Pour chaque image porteuse d'information ayant une alternative textuelle, cette alternative est-elle* ***pertinente*** *?*

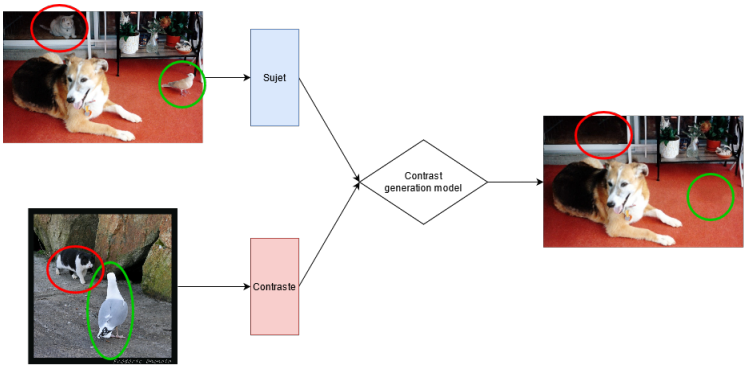
A terme l'outil d'audit automatique devra être capable de prendre en compte le contexte d'un document web afin d'améliorer la tâche d'analyse et de générer des

1. Image qui véhicule une information nécessaire à la compréhension du contenu auquel elle est associée.

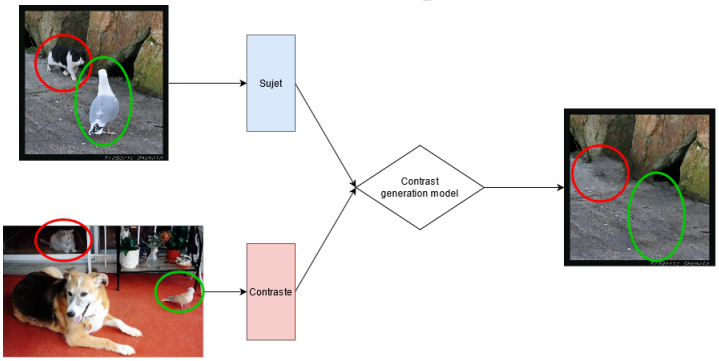
2. Texte qui décrit l'image qui restitué par les technologies d'assistance pour les éléments graphiques dont font parties les images

propositions d'améliorations cohérentes avec la *sémantique morpho-dispositionnelle* du document, c'est à dire le sens véhiculé par des configurations graphiques exploitant des marques de ponctuation et de mise en forme [Mau04], *i.e.* le contexte de l'image dans notre cas.

En lien avec cette thèse et sa partie portant sur la proposition de nouveaux contenus permettant d'améliorer la compréhension des pages webs, mon stage avait pour objectif de construire un modèle d'apprentissage profond auto-supervisé capable de générer des images mettant en avant le contraste entre deux images. Nous avons décidé de définir le contraste comme étant la chose qui distingue une image d'une autre, plus formellement c'est la différence "ensembliste" entre les éléments sémantiques, *i.e* les contenus des deux images. La figure 1 illustre le résultat attendu : le modèle prend en entrée une image sujet et une image de contraste ; sur la figure 1a les éléments communs - chat et oiseau - sont supprimés de l'image sujet pour ne laisser que le chien et l'arrière plan. Alors que pour la figure 1b le modèle ne retourne que l'arrière plan.



(a) Le contraste entre l'image sujet et l'image de contraste est le chien et l'arrière plan



(b) Le contraste entre l'image sujet et l'image de contraste est l'arrière plan

FIGURE 1 – Exemples d'une génération de contraste désirées

2 Plan du mémoire

Ce mémoire est organisé comme suit : la deuxième partie présente la tâche d'audit RGAA pour la quelle je vais proposer des outils d'analyse sémantique de contenu. La troisième partie est consacrée à la présentation de l'état de l'art sur le quel je me suis appuyé. La quatrième partie est centrée sur le jeu de données et le modèle de génération de contraste dévelopés au cours de ces six derniers mois. Finalement, la dernière partie propose un bilan de ce stage et donne des perspectives pour la poursuite de ce travail.

Deuxième partie

Accessibilité et audit

1 Notions d'accessibilité numérique

Tout d'abord définissons de ce qu'est le handicap, L'article L. 114 du code de l'action sociale et des familles nous dit : "toute limitation d'activité ou restriction de participation à la vie en société subie dans son environnement par une personne en raison d'une altération substantielle, durable ou définitive d'une ou plusieurs fonctions physiques, sensorielles, mentales, cognitives ou psychiques, d'un polyhandicap ou d'un trouble de santé invalidant".

L'accessibilité numérique signifie que les sites web, technologies et outils sont conçus et développés de façon à ce que les personnes handicapées puissent les utiliser. Ces services doivent être :

— **perceptibles** : par exemple, proposer des équivalents textuels à tout contenu non textuel ;

— **utilisables** : par exemple, rendre toutes les fonctionnalités accessibles au clavier ; laisser à l'utilisateur suffisamment de temps pour lire et utiliser le contenu ; l'aider à accéder rapidement au contenu ;

— **compréhensibles** : par exemple, faire en sorte que les pages fonctionnent de manière prévisible et contrôlable ; aider l'utilisateur à corriger les erreurs de saisie.

— **robustes** : par exemple, optimiser la compatibilité avec les technologies d'assistance (lecteur d'écran, etc.).

Les services webs concernés par cette normes sont ceux du secteur public (services gouvernementaux, etc.), les entités privées d'intérêt public (banques, services pour personnes handicapées etc.) et les grandes entreprises dont le chiffre d'affaire est supérieur à 250 millions d'euros. En cas de non conformité et de refus d'aménagement raisonnable pour rendre le ou les services accessibles, l'entité pourra être condamnée à payer 25 000 € d'amende par service non conforme.

2 Audit RGAA (Référentiel général d'amélioration de l'accessibilité)

2.1 Présentation du RGAA

En France, la norme qui régit l'accessibilité est le *Référentiel général d'amélioration de l'accessibilité* (RGAA).

Le RGAA propose un cadre opérationnel de vérification de la conformité, *i.e.* d'audit, aux exigences d'accessibilité. Ce cadre est composé de 106 critères répartis en 13 catégories : images, cadres, couleurs, multimédia, tableaux, liens, scripts, éléments obligatoires, structuration de l'information, présentation de l'information, formulaires, navigation et consultation. Plus globalement, elles concernent la bonne utilisation des balises HTML, la description des contenus permettant de les rendre accessibles, l'aide à la navigation et les interactions que l'utilisateur peut avoir avec un élément dynamique. Afin de réduire la marge d'interprétation lors de leurs évaluations ces critères sont composés de tests indiquant les éléments à vérifier et les différents cas de figure, nous verrons plus tard comment un audit se déroule. Voici quelques exemples :

— *Critère 1.1.* Chaque image porteuse d'information a-t-elle une alternative textuelle ?

— *Critère 1.2.* Chaque image de décoration est-elle correctement ignorée par les technologies d'assistance ?

— *Critère 3.1.* Dans chaque page web, l'information ne doit pas être donnée uniquement par la couleur. Cette règle est-elle respectée ?

— *Critère 3.2.* Dans chaque page web, le contraste entre la couleur du texte et la couleur de son arrière-plan est-il suffisamment élevé (hors cas particuliers) ?

— *Critère 8.9.* Dans chaque page web, les balises ne doivent pas être utilisées uniquement à des fins de présentation. Cette règle est-elle respectée ?

— *Critère 9.2.* Dans chaque page web, la structure du document est-elle cohérente (hors cas particuliers) ?

2.2 Réalisation d'un audit RGAA

Un audit se déroule en trois étapes :

1. sélection des pages à auditer ;
2. l'auditeur analyse chacune des pages selon les critères évoqués plus tôt et propose des corrections et des améliorations pour l'accessibilité du service ;
3. édition de la déclaration d'accessibilité

2.2.1 Sélection des pages à auditer

Les pages obligatoires dans un échantillon d'audit sont les pages :

— d'accueil ;
— de contact ;
— les mentions légales ;
— la déclaration d'accessibilité ;
— le plan du site ;
— d'aide ;
— d'authentification.

En plus de ces dernières il nous est demandé d'inspecter, si elles existent, une page pertinente de chaque service et les fonctionnalités principales du site :

— au moins une page pertinente pour chaque type de service fourni ;
— la fonctionnalité de recherche ;
— au moins un document téléchargeable pertinent, pour chaque type de service fourni ;
— les pages constituant un processus, comme une transaction en ligne ;
— des exemples de pages ayant un aspect sensiblement distinct ou présentant un type de contenu différent, comme des tableaux de données ou des contenus multimédias.

Un troisième ensemble de pages s'ajoute au pages décrites ci-dessus, ce sont les pages sélectionnées au hasard, il doit représenter au moins 10 % des pages auditées.

Il est important de noter que la sélection des pages auditées ainsi que leur nombre doivent être représentatifs du service de communication au public en ligne. Le nombre de visiteurs par page peut être pris en compte lors de la constitution de l'échantillon.

2.2.2 Outils de l'auditeur

L'outil le plus utile reste l'inspecteur de code que propose chaque navigateur. Il permet de rechercher facilement les éléments et les attributs requis pour un test d'accessibilité. Il existe aussi des extensions de navigateur telles que *Web Developer Toolbar*, *HeadingsMap*, *ColorContrastChecker*, qui respectivement nous permettent de faciliter le repérage visuels de certains éléments du document, de vérifier l'arborescence du document et de valider si le contraste entre la couleur d'un texte et de son fond est assez prononcé.

Des services plus complets, tels que *Wave*, permettent une analyse complète du document selon les critères ne faisant pas appels à une prise en compte du contexte de l'élément pour l'analyser.

Bien que ces outils permettent un gain de temps, ils ne couvrent que 30% des 106 critères. Augmenter ce pourcentage permettra à l'auditeur de se concentrer sur son rôle de conseiller à l'amélioration de l'accessibilité du service.

2.2.3 Analyse d'une page

Pour une page, chaque critère est statué à l'aide de différents tests. L'auditeur lui donne le statut de *Conforme*, *Non applicable* ou *Non conforme*. De plus, il se doit d'ajouter, si nécessaire, des propositions d'améliorations ou de corrections pour ces différents critères. C'est à partir de son travail que les concepteurs des sites pourront les améliorer. Il référence pour chaque page ces informations dans un tableur (*cf.* 2) qui sera fourni au commanditaire avec le bilan final.

RGAA 4.1 – GRILLE D'ÉVALUATION
Accueil : https://koena.net/ressources-formation/site_exercice_autisme/non_conforme/index.html

Thématique	Critère	Recommandation	Statut	Dérogation	Modifications à apporter	Commentaires en cas de dérogations
LIENS	6.1	Chaque lien est-il explicite (hors cas particuliers) ?	C	N	Conseil : ajouter une note indiquant, pour les liens dans ce cas, qu'ils s'ouvrent dans un nouvel onglet. Par exemple, <intitulé du lien> (nouvelle page)'. pour le lien allant vers le fichier pdf ajouter '(pdf)" à l'intitulé du lien	
	6.2	Dans chaque page web, chaque lien a-t-il un intitulé ?	C	N		
SCRIPTS	7.1	Chaque script est-il, si nécessaire, compatible avec les technologies d'assistance ?	NC	N	cf doc	
	7.2	Pour chaque script ayant une alternative, cette alternative est-elle pertinente ?	NA	N		
	7.3	Chaque script est-il contrôlable par le clavier et par tout dispositif de pointage (hors cas particuliers) ?	NC	N	cf doc	
	7.4	Pour chaque script qui initie un changement de contexte, l'utilisateur est-il averti ou en a-t-il le contrôle ?	C	N	oui il est inité par un <button>	
	7.5	Dans chaque page web, les messages de statut sont-ils correctement restitués par les technologies d'assistance ?	NA	N		
ÉLÉMENTS OBLIGATOIRES	8.1	Chaque page web est-elle définie par un type de document ?	NC	N	cf doc	
	8.2	Pour chaque page web, le code source généré est-il valide selon le type de document spécifié (hors cas particuliers) ?	NC	N	cf doc	
	8.3	Dans chaque page web, la langue par défaut est-elle présente ?	NC	N	cf doc	
	8.4	Pour chaque page web ayant une langue par défaut, le code de langue est-il pertinent ?	NA	N		
	8.5	Chaque page web a-t-elle un titre de page ?	C	N		
	8.6	Pour chaque page web ayant un titre de page, ce titre est-il pertinent ?	C	N	cf doc	
	8.7	Dans chaque page web, chaque changement de langue est-il indiqué dans le code source (hors cas particuliers) ?	NC	N	cf doc	
	8.8	Dans chaque page web, le code de langue de chaque changement de langue est-il valide et pertinent ?	NC	N	cf doc	

FIGURE 2 – Exemple de grille d'audit

2.2.4 Édition de la déclaration d'accessibilité

Suite à l'audit d'un échantillon de pages d'un site web l'état de conformité est calculé. Il correspond au pourcentage des critères respectés sur l'ensemble des pages. Par exemple, si un critère est respecté sur sur trois pages sur cinq il sera considéré comme non valide. Un second score est aussi mentionné : le *taux moyen de conformité* qui considère chaque critère de chaque page indépendemment. Le service peut-être en :

— conformité totale : si tous les critères de contrôle du RGAA sont respectés ;

— conformité partielle : si au moins 50 % des critères de contrôle du RGAA sont respectés ;

— non-conformité : si moins de 50 % des critères de contrôle du RGAA sont respectés.

Ce résultat doit être publié officiellement dans une déclaration d'accessibilité et rendu disponible sur le site concerné et dans un format accessible.

Troisième partie

État de l'art

Comme présenté en introduction le but de ce stage était de construire un modèle de génération d'images dont la sémantique est contrôlée par deux images données en entrée au modèle. L'approche retenue exploite des techniques d'apprentissage supervisé par réseau de neurones profonds c'est pourquoi nous présenterons les notions générales liées à l'apprentissage supervisé, puis les architectures des réseaux de neurones utilisées ici.

1 Apprentissage supervisé

1.1 Objectif d'un modèle supervisé

Posons $\mathcal{X}$ l'ensemble des entrées possibles et $\mathcal{Y}$ l'ensemble des sorties possibles, comme des vérités terrain produites par un humain, il existe une distribution de probabilité inconnue sur l'ensemble $\mathcal{X} \times \mathcal{Y}$, que nous noterons $p(x, y)$ avec $(x, y) \in \mathcal{X} \times \mathcal{Y}$. L'apprentissage consiste à la recherche d'une fonction $f : \mathcal{X} \to \mathcal{Y}$ telle que $f(W, x) = Wx = \hat{y}$ où $\hat{y}$ est une approximation de y et où W est l'ensemble des poids que l'on cherche pour faire l'inférence. L'écart entre y et $\hat{y}$ est mesurée par une fonction de coût notée $L(y, \hat{y})$ qui est différentiable. De ces éléments en découle la notion de risque attendu R :

$$R(f) = \int_{\mathcal{X} \times \mathcal{Y}} p(x, y) \dot{L}(y, \hat{y})\, dx, dz \tag{1}$$

L'objectif est alors de trouver f^* tel que $R(f^*)$ est minimal. Mais comme $p(x, y)$ est inconnue il est nécessaire d'approximer R en définissant une sous-espace $\mathcal{D} \in \mathcal{X} \times \mathcal{Y}$ composé de n couples (x_i, y_i). Cette approximation s'appelle le risque empirique $R_{\mathcal{D}}$:

$$R_{\mathcal{D}}(f = \frac{1}{n} \sum_{i}^{n} L(y_i, \hat{y}_i) \tag{2}$$

Le but de l'apprentissage revient maintenant à trouver f^* qui minimise $R_{\mathcal{D}}(f^*)$. Nous verrons plus tard comment cette optimisation est réalisée.

1.2 Régression et classification

L'apprentissage supervisé permet de résoudre les problèmes de régression et de classification. L'objectif de la régression est de prédire et/ou expliquer les valeurs

d'une variable quantitative Y, la variable à expliquer, à partir des valeurs de p variables $X_1 ... X_p$, les variables explicatives. L'objectif de la classification est de déterminer au quel des q groupes, représentés par une variable qualitative, appartient un nouvel individu, décrit par p variables$X_1 ... X_p$, à le plus de chance d'appartenir.

1.3 Apprentissage d'un modèle supervisé

Avant de présenter les différentes architectures de réseaux de neurones, il est intéressant de savoir comment un modèle supervisé est capable d'apprendre.

Nous venons de voir que nous pouvons résumer un modèle par une fonction :

$$\hat{y} = f(W, x) \tag{3}$$

qui utilise une ensemble de paramètres W pour estimer le résultat $\hat{y}$ à partir de l'entrée x. Afin d'avoir un modèle capable de résoudre la tâche donnée il faut ajuster W à l'aide d'une fonction de perte L qui nous calcul l'erreur du modèle pour les prédictions faîtes sur un ensemble de données d'entraînement $\mathcal{D} = \{(x_i, y_i), ..., (x_n, y_n)\}$ de taille n dont nous connaissons les vérités terrain. L'objectif est de minimiser cette fonction de coût $L(f(W, x), y)$ pour l'ensemble D, ce qui revient à minimiser le risque empirique $R_{\mathcal{D}}$ (*cf.* équation 2).

Afin de minimiser L nous allons utiliser la rétro-propagation du gradient [RHW86] qui permet d'ajuster les poids W. Cet algorithme est en deux étapes :

— La propagation avant qui permet de calculer les sorties du réseau à partie des entrées

— La propagation arrière durant laquelle les dérivées partielles de la fonction de coût L par rapport aux paramètres W du modèle sont rétro-propagés pour mettre à jour chaque poids w_i du réseau en fonction de sa dérivée partielle et du taux d'apprentissage α :

$$\Delta w_i = -\alpha \frac{\partial L}{\partial w_i} \tag{4}$$

Il est important de noter que L n'est pas obligatoirement une fonction convexe ce qui implique que le modèle peut converger vers un minimum local qui n'est pas le minimum global de la fonction. L'algorithme de descente de gradient standard présenté ci-dessus risque de converger vers un minima local. Pour éviter cela il est possible d'utiliser des méthodes de descente stochastique par lot qui consiste à faire la mise à jour des poids à partir du gradient moyen calculé sur des petits sous ensemble de D. Cela permet de changer la direction du gradient pour chacun de ces lots et potentiellement de sortir d'un minimum local tout en disposant d'une convergence

rapide. De plus, il existe différents améliorations de cette approche avec notamment Adagrad [DHS11], Adam [KB17], Momentum [Qia99].

2 Architectures des réseaux de neurones

Les modèles d'apprentissage neuronaux sont destinés à approximer des fonctions complexes qui dépendent d'un grand nombre d'entrées et dont les poids sont organisés selon le type d'architecture sélectionnées. Cette famille de méthode permet de traiter des signaux bruts complexes, tel que des images. Nous commencerons par une description du perceptron et du perceptron multi-couches qui sont les éléments à la base d'un réseau de neurones, puis nous poursuivrons avec des architectures plus récentes dédiées au traitement d'image telles que les réseaux convolutifs, auto-encodeurs et génératifs adverse.

2.1 Perceptron

Le perceptron [Ros58] est l'élément de base d'un réseau de neurones. Proposé pour le problème de classification ou de régression linéaire. Il est définit par :

$$\hat{y} = \sigma(Wx) \tag{5}$$

où :

— x : le vecteur d'entrée du modèle de taille n ;

— W : le vecteur de poids de taille n à ajuster ;

— σ : la fonction d'activation tq. $\sigma(Wx) = \begin{cases} 1 & \text{si} \quad \sum_{i=1}^{n} w_i x_i > \theta \\ 0 & \text{sinon} \end{cases}$ permet de déterminer la classe de x (0 ou 1).

— $\hat{z}$: la classe attribuée à x.

2.2 Perceptron multi-couches (MLP)

Comme vu ci-dessus, le perceptron classique ne permet pas de traiter des problèmes non-linéaires. Une manière résoudre ce problème est de projeter de manière non linéaire les données dans un espace de redescription où elles seront linéairement séparables. Le perceptron multi-couches [RHW86] utilise ce principe en empilant plusieurs couches de perceptrons, appelées les couches cachées, intercalés par des fonctions d'activations, ce qui permet de créer un espace de projection des données linéairement séparable. La figure 3 présente la structure d'un MLP à deux couches cachées où x_i représente les différentes entrées du modèle, h_i^l les neurones de la $l^{\text{ème}}$ souche

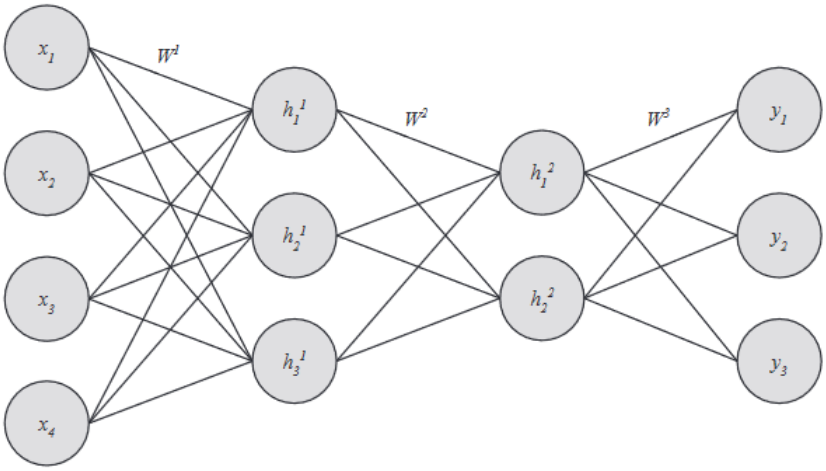


FIGURE 3 – Exemple de MLP à deux couches cachées

cachée et y_i les sorties, chacune des couches sont inter-connectées par les matrices de poids W^l

2.3 Réseau de neurones convolutif (CNN)

2.3.1 Structure générale

Les CNN sont des réseaux multi-couches particulièrement adaptés au traitement des signaux en deux dimensions comme les images. Popularisés avec les travaux de Y. Le Cun et *al.* dans les années 1990 avec les réseaux *LeNet* [LeC+89] pour la reconnaisse de caractères manuscrits. Cette famille de réseau tire sa force de l'utilisation de filtres permettant d'exploiter la topologie des images et de réduire le nombre de paramètres à estimer, ce qui permet d'avoir de meilleurs résultats et une complexité moins importantes qu'avec les MLP.

L'architecture d'un CNN est basée sur trois type de couches :

1. une **couche de convolution** qui consiste à appliquer un noyau de taille $n \times m$ sur l'image afin d'extraire des cartes de caractéristiques. Ces cartes contiennent les motifs élémentaires de l'image. De plus, les poids de la convolution, *i.e.* le noyau, est partagé entre toutes les positions de l'image ce qui réduit le nombre de paramètres à apprendre et le rend indépendant de la taille de l'entrée.
2. une **couche d'agrégation** qui permet de sous-échantillonner les cartes obtenus par la convolution. Cette agrégation peut-être une moyenne, un maximum ou un minimum d'un ensemble de valeurs contiguës.
3. une **couche totalement connectée** avec une fonction d'activation non linéaire. Cette couche permet de faire les prédictions pour un problème de classification ou de régression.

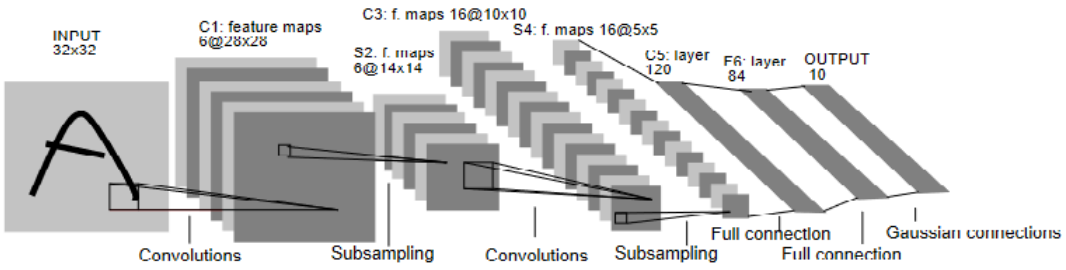


FIGURE 4 – Schéma du réseau *LeNet-5* [LeC+89]. Les termes *Convolutions*, *Subsampling*, et *Full connection* correspondent aux couches de convolutions, d'agrégation et totalement connectés.

L'agencement de ces couches est illustré par la figure 4.

Au cours des dix dernières années, la construction de bases de données plus grande et variées, comme ImageNet [Den+09] et MS-COCO [Lin+15], a permis de développer des réseaux avec plus de couches comme *AlexNet* [KSH12] et le *ResNet* [KSH12] que nous avons sélectionné comme extracteur de cartes de caractéristiques pour notre modèle qui sera présenté plus tard.

2.4 Auto-encodeur (AE)

Contrairement aux tâches de classification et de régression citée plus-tôt, l'objectif d'un auto-encodeur est d'apprendre une représentation compressée de la donnée et de savoir la reconstruire le mieux possible. Cette architecture, illustrée en figure 5, est divisée en deux partie :

— l'**encodeur**, qui est un réseau de neurone classique, un CNN pour le traitement d'images, qui permet de compresser l'information vers un espace latent z de dimension beaucoup plus faible que celle de l'image d'origine, ce qui permet de ne conserver que les informations les plus significatives.

— le **décodeur** quand à lui cherche à reconstruire l'image d'origine à partir de la représentation condensée z.

Dans le cadre de ce projet, utiliser un AE est un moyen de construire une représentation condensée, manipulable via des opérations spécifiques afin de supprimer ou d'ajouter des éléments sémantiques lors de la génération de nouveaux contenus contrastés.

2.5 Réseau génératif adverse (GAN)

Les modèles de type GAN [Goo+14] permettent de générer des contenus. L'idée principale est d'entraîner simultanément deux réseaux : un générateur et un discriminateur. L'objectif du générateur est de produire des contenus synthétiques réalistes

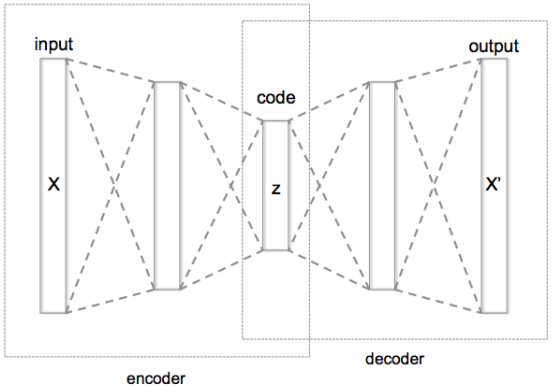


FIGURE 5 – Schéma d'un auto-encodeur. Le signal d'entrée x est condensé dans une représentation latente z. Puis le décodeur reconstruit une approximation du signal d'entrée d'origine x'

.

à partir d'un vecteur aléatoire. En d'autres termes, le générateur apprend à faire correspondre des points d'un espace latent qui, avant l'entraînement, na pas de réelle signification, avec des image de sortie spécifiques. Le discriminateur quand à lui apprend à distinguer les contenus générés des contenus réels (présents dans la base d'entraînement). L'entraînement commun des deux modèles les met en compétition : le générateur doit produire des contenus de plus en plus réalistes afin de tromper le discriminateur qui lui doit affiner sa capacité à déceler les faux contenus. Le modèle a convergé quand la probabilité du discriminateur de se tromper est de $\frac{1}{2}$.

3 Modèles de manipulation sémantique

Nous pouvons trouver dans la littérature des modèles de GAN ou de AE capables de transformer la sémantique d'une image. Nous rappelons ici quelques propositions existantes.

DCGAN - Deep Convolutional Adversarial Network

A. Radford et *al.* dans [RMC16] propose le modèle *DCGAN* qui introduit une architecture convolutive de GAN dont l'apprentissage est stable. Au cours de leurs expérimentations les auteurs ont explorés l'espace latent de leurs GANs entraînés sur plusieurs jeux d'entraînement différents, et notamment sur des données regroupant les visages de célébrités. Ceci leur a permis de démontrer deux points intéressants. Le premier est l'application d'opérateurs arithmétiques, comme [Mik+13] dans l'évaluation de GANs en traitement automatique des langues, sur les vecteurs de l'espace latent afin de générer de nouveaux visages dont les expressions sont différentes. La

figure 6 illustre bien cet aspect, le visage d'une femme souriante moins le visage d'une femme neutre plus le visage d'un homme neutre a donné le visage d'un homme souriant.

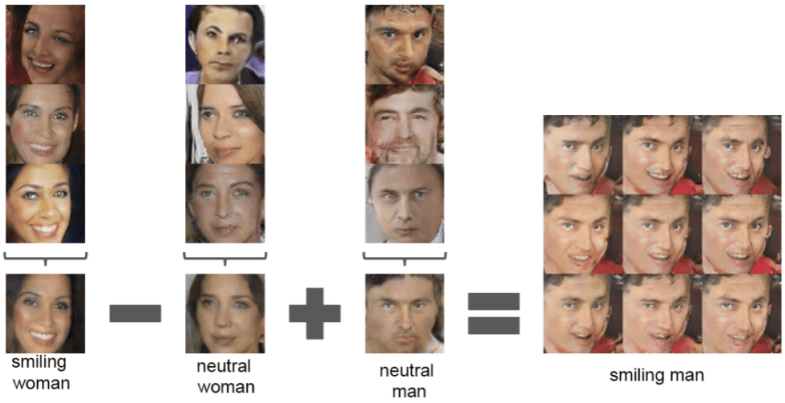


FIGURE 6 – Exemple d'arithmétique vectorielle sur des points de l'espace latent pour générer des visages avec le DCGAN [RMC16].

La deuxième démonstration concerne la transition entre deux visages générés, notamment en créant un chemin linéaire entre deux points de la dimension latente, *i.e.* deux visages, puis en générant tous les visages à intervalle régulier entre ces deux points. La figure 7 montre que la variation linéaire de la position dans l'espace latent engendre la rotation de la gauche vers la droite du visage généré.

FIGURE 7 – Exemple de chemin entre deux visages générés [RMC16]

Comme énoncée plus tôt notre problématique est de générer le contraste entre deux images, *i.e.* générer une image qui serait la soustraction des éléments de celles-ci. La capacité à réaliser une opération arithmétiques sur l'espace latent de ce GAN est en lien avec ce que nous cherchons mais il ne permet pas de requêter le modèle avec nos deux images que nous voulons contraster afin de connaître leur position dans

l'espace latent pour ensuite réaliser notre opération de soustraction pour en générer une nouvelle.

The Many Moods of Emotion

Dans [Vie+18] Vielzeuf et *al.* propose deux modèles afin de construire une représentation des émotions continues à partir d'étiquettes discrètes et d'appliquer à un visage une expression désirée.

Le premier un permet de construire une projection en trois dimensions des expressions faciales en utilisant l'espace latent en trois dimensions d'un réseau de convolution capable de classifier les émotions. La figure 8 illustre la création d'un espace continu permettant d'avoir des coordonnées pour chaque type d'émotions.

FIGURE 8 – Illustration de la représentation des émotions dans un espace 3-d [Vie+18]. Chaque plan est coloré en fonction de la classe discrète d'émotion. Pour chaque zone plusieurs expressions sont générées.

Le second réseau est un GAN qui exploite cette représentation en trois dimensions afin de modifier l'expression d'un visage donné en paramètre. L'apprentissage du modèle est contrôlée par quatre fonction de perte :

— *Adversarial loss* (L_{adv}) qui permet de générer de faux visages qui ne sont pas distinguable d'un vrai par de discriminateur ;

— *Real regression loss* (L_{reg}^{real}) qui force le discriminateur à prédire les coordonnées de l'émotion de l'image d'origine ;

— *Fake regression loss* (L_{reg}^{fake}) qui force le générateur à construire un visage avec l'expression souhaitée ;

— *Reconstruction loss* (L_{rec}) qui permet que la visage généré converse le contenu qui n'est pas en lien avec l'expression du visage.

La figure 9 illustre les résultats obtenus par ce modèle, Nous pouvons observer que les résultats pour ce visage correspondent aux émotions demandées.

Pour conclure, nous retenons l'idée de créer un espace de projection des images interrogeable et exploitable par un GAN pour la génération ainsi que la fonction de perte de reconstruction permettant de conserver le contenu nécessaire - celui qui est

FIGURE 9 – Exemple de résultats obtenus par l'approche [Vie+18] pour la transformation du visage dans sept classes d'émotions.

unique à notre image sujet dans notre cas. Mais, les images que nous voulons traiter sont très variées et comportent différents éléments à segmenter.

Quatrième partie

Contributions

1 Création d'un jeu de données pour la génération de contraste

A l'heure actuelle, cette tache de génération de contraste comme nous l'avons défini semble nouvelle. De ce fait, il n'existe pas de jeu de données où chaque entrée est un couple d'image répondant à des contraintes liées à leurs contenus.

1.1 Choix de la base initiale

Afin de pouvoir construire un premier jeu de données dans un temps restreint nous avons fait le choix d'exploiter la base d'images *MS-COCO* [Lin+15] initialement dédié à la segmentation d'objets hétérogènes et de classification de scène. Cette base regroupe 328k images contenant 91 classes d'objets différentes réparties dans 12 méta-classes : *personnes*, *vehicules*, *intérieur*, *extérieur*, *animaux*, *accessoires*, *sports*, *cuisine*, *électronique*, *appareils*, et *meubles*. La figure 10 présente un échantillon de la base indexé par les classes.

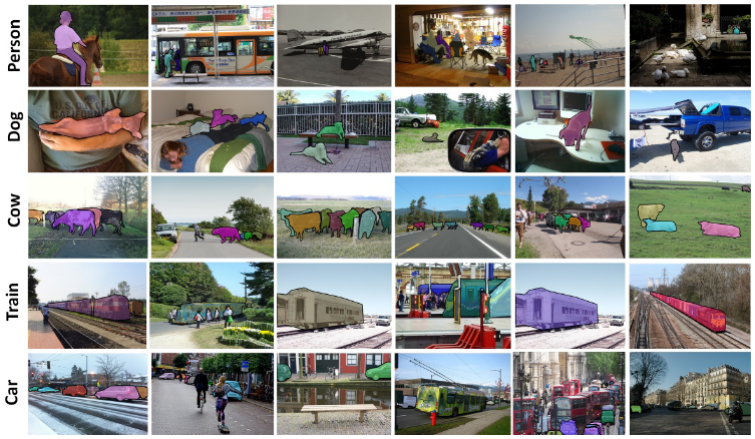


FIGURE 10 – Extrait de la base *MS-COCO* [Lin+15]. Nous pouvons observer que les images et les scènes sont très diversifiées.

Cette base correspond à nos attente car elle contient une diversité d'objets et de scènes qui n'est pas présente dans *ImageNet* [Den+09] qui elle contient majoritairement des images qui ne possède qu'une seule image et qu'une seule instance. Ce qui nous permettra de construire plus de 100k paires d'images.

1.2 Construction du jeu de contraste

Bien que le nombre potentiel de paires est important nous avons fait des choix concernant la sélection des couples.

Le premier est de ne conserver que des images ne contenant que des éléments appartenant à exactement deux méta-classes ce qui, selon mon hypothèse, permet de simplifier l'apprentissage et l'étude des résultats. Il est bon de préciser qu'une image peut posséder plusieurs instances (objets) d'une même méta-classe.

Sur le diagramme de la figure 11 nous pouvons observer que la méta-classe *personne* est sur-majoritaire. De ce fait le premier choix est affiné en ne sélectionnant que les images avec au moins une personne. Ceci implique que chaque image de chaque couple ne sera sélectionnée qu'en fonction d'une seule méta-classe.

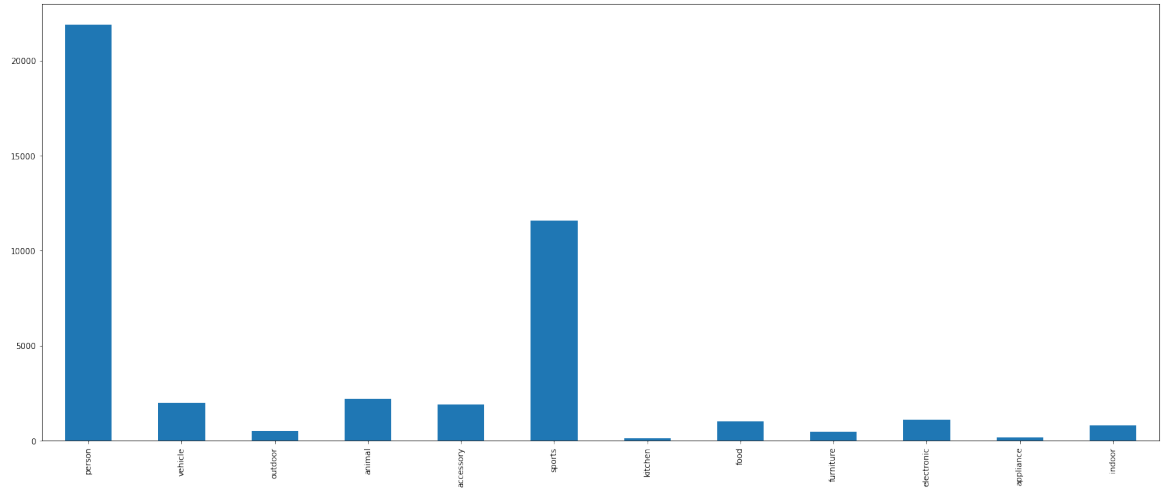


FIGURE 11 – Graphique indiquant le nombre d'images dans les quelles apparaissent chaque méta-classes

Le deuxième et dernier choix est de construire les couples de manière "intra méta-classe" afin de conserver une certaine proximité sémantique entre les images.

Une fois ces opérations de filtrage et de regroupement réalisées le jeu est construit par tirage aléatoire. Le jeu obtenu comporte 11k paires. La répartition des classes parmi les paires est indiqué en table 1 ; nous pouvons observer que la catégorie sport est la plus représentée ce qui concorde avec le graphique 11 où cette méta-classe est la deuxième la plus représentée. Nous pouvons voir en figure 12 des exemples des paires construites.

Méta-classe	Nombre de paires
sports	5782
animal	1114
accessory	1056
electronic	524
food	517
indoor	411
outdoor	271
furniture	254
appliance	87
kitchen	58

TABLE 1 – Répartition des paires d'images selon les méta-classes

FIGURE 12 – Exemples de paires construites

2 Modèle de génération de contraste

Pour rappel l'objectif de notre modèle est de générer une image marquant la différence entre deux images données en entrée et cela de manière auto-supervisée. Cette section décrit son architecture. Nous verrons dans un premier temps la méthode de

représentation des images. Puis, le GAN permettant d'exploiter cette représentation.

2.1 Représentation des images

A l'instar de ce qui est fait dans [Vie+18], présenté plus tôt, nous voulons créer une représentation latente des images. Pour cela nous avons optés pour un modèle auto-encodeur siamois, illustré en vert en 13, dont les objectifs sont d'apprendre à séparer la représentation des images sujet x_f et de contraste x_c dans :

— z_f^c la représentation des éléments propres à x_f ;

— z_c^c la représentation des éléments propres à x_c ;

— z_u les éléments de communs aux deux images.

et à générer une nouvelle image réaliste à partir de z_f^c et qui regroupe tous les éléments uniquement présents dans x_f.

Pour cette partie du modèle nous utilisons une architecture *U-Net* [RFB15] modifiée.

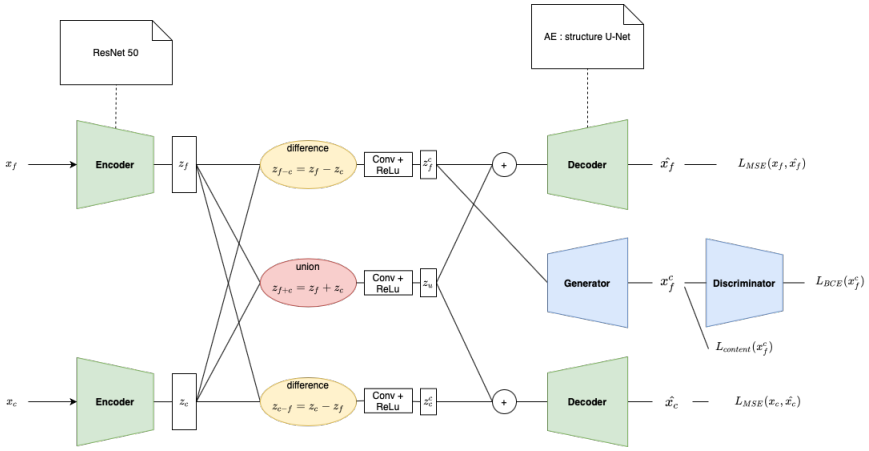


FIGURE 13 – Schéma de notre modèle de génération de contenu

U-Net

L'architecture *U-Net* [RFB15] est initialement dédiée à la segmentation de données. Ce qui la distingue des AE classique est que les couches cachées de l'encodeur et du décodeur sont connectées, *cf.* 14 ; lors de la reconstruction de l'image les cartes de caractéristiques produites lors de la phase descendante sont concaténée avec celles de même dimension produite lors de la phase ascendante. Cette particularité permet de mieux reconstruire les cartes de caractéristiques de plus grande résolution. La partie convolutionnelle, située au milieu du réseau, de dimension $512 \times 28 \times 28$ correspond à la représentation latente sur laquelle nous réalisons nos opérations de *différence* et

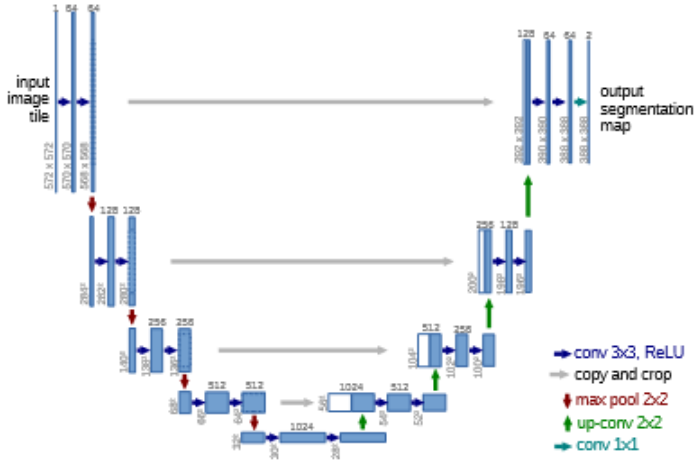


FIGURE 14 – Schéma du modèle U-Net [RFB15]. Le réseau est symétrique, le décodeur de droite exploite les cartes de caractéristiques extraites par l'encodeur pour mieux reconstruire l'image.

d'*union* pour produire z_f^c, z_c^c et z_u. Ces opérations correspondent respectivement à une soustraction et une concaténation.

Afin de contrôler de mesurer la qualité de la reconstruction nous utilisons une fonction erreur quadratique moyenne :

$$L_{MSE} = \mathbb{E}_{z_i^c}\left[(x_i - D_{AE}(z_i^c \oplus z_u)^2\right] \tag{6}$$

où :

— x_i est l'image d'origine ;

— D_{AE} le décodeur de l'AE ;

— z_i^c la représentation du contenu propore à l'image x_i ;

— z_u la représentation du contenu comment à x_i et x_j.

2.2 Génération de contenu

Pour générer notre nouvelle image contrastée nous avons utilisé un *DCGAN* (*cf.* 3) qui prend en paramètres la représentation séparée de l'image sujet (z_f^c). Afin de contrôler la génération d'images réalistes nous utilisons la fonction de perte d'entropie croisée :

$$L_{BCE} = \mathbb{E}_{x_f,x_c}\left[logD(x_f) + logD(x_c)\right] + \mathbb{E}_{z_f^c}\left[log1 - D(G(x_f^c))\right] \tag{7}$$

où

— D et G le discriminateur et le générateur ;

— x_f et x_c sont les images sujet et de contraste initiales ;

— x_f^c l'image généré à partir de z_f^c sensé représenter ce qui est propre à x_f ;

Cinquième partie

Conclusion

Le travail réalisé lors de ce stage et notamment ma formation d'auditeur RGAA m'a permis de cerner les différentes problématiques de l'accessibilité et de l'audit.

Avec l'augmentation du numérique dans notre société, l'accès au web est un droit. De nombreux contenus ne sont pas adaptés pour être appréciés par tous. Les outils de génération par apprentissage profond semblent être une voie pour proposer de nouvelles images plus adaptées car plus simple à comprendre et plus en lien avec le contenu de la page.

Cette proposition de modèle de représentation de génération de contraste sémantique propose une nouvelle tâche à la communauté. De ce fait, le travail réalisé concernant la création d'un jeu de données de référence et son utilisation dans le modèle proposé ici sera prolongé au cours de ma thèse.

Références

[Ros58] F. Rosenblatt. « The perceptron : A probabilistic model for information storage and organization in the brain. » In : *Psychological Review* 65.6 (1958), p. 386-408. ISSN : 0033-295X. DOI : 10.1037/h0042519. URL : http://dx.doi.org/10.1037/h0042519.

[RHW86] David E. Rumelhart, Geoffrey E. Hinton et Ronald J. Williams. « Learning Representations by Back-propagating Errors ». In : *Nature* 323.6088 (1986), p. 533-536. DOI : 10.1038/323533a0. URL : http://www.nature.com/articles/323533a0.

[LeC+89] Yann LeCun et al. « Handwritten digit recognition with a back-propagation network ». In : *Advances in neural information processing systems* 2 (1989).

[Qia99] Ning Qian. « On the momentum term in gradient descent learning algorithms ». In : *Neural Networks* 12.1 (1999), p. 145-151. ISSN : 0893-6080. DOI : https://doi.org/10.1016/S0893-6080(98)00116-6. URL : https://www.sciencedirect.com/science/article/pii/S0893608098001166.

[Mau04] Fabrice Maurel. « Transmodalité et multimodalité écrit/oral : modélisation, traitement automatique et évaluation de stratégies de présentation des structures “visuo-architecturale” des textes ». Thèse de doct. Université de Toulouse, 2004.

[Den+09] Jia Deng et al. « ImageNet : A large-scale hierarchical image database ». In : *2009 IEEE Conference on Computer Vision and Pattern Recognition*. 2009, p. 248-255. DOI : 10.1109/CVPR.2009.5206848.

[DHS11] John Duchi, Elad Hazan et Yoram Singer. « Adaptive Subgradient Methods for Online Learning and Stochastic Optimization ». In : *Journal of Machine Learning Research* 12.61 (2011), p. 2121-2159. URL : http://jmlr.org/papers/v12/duchi11a.html.

[KSH12] Alex Krizhevsky, Ilya Sutskever et Geoffrey E. Hinton. « ImageNet Classification with Deep Convolutional Neural Networks ». In : *Advances in Neural Information Processing Systems 25*. Sous la dir. de F. Pereira et al. Curran Associates, Inc., 2012, p. 1097-1105. URL : http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf.

[Mik+13] Tomas Mikolov et al. *Distributed Representations of Words and Phrases and their Compositionality*. 2013. arXiv : 1310.4546 [cs.CL].

[Goo+14] Ian J. Goodfellow et al. « Generative Adversarial Nets ». In : *27th International Conference on Neural Information Processing Systems*. Montreal, Canada : MIT Press, 2014, p. 2672-2680.

[Lin+15] Tsung-Yi Lin et al. *Microsoft COCO : Common Objects in Context*. 2015. arXiv : 1405.0312 [cs.CV].

[RFB15] Olaf Ronneberger, Philipp Fischer et Thomas Brox. *U-Net : Convolutional Networks for Biomedical Image Segmentation*. 2015. arXiv : 1505.04597 [cs.CV].

[RMC16] Alec Radford, Luke Metz et Soumith Chintala. « Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks ». In : *4th International Conference on Learning Representations, ICLR 2016, San Juan, Puerto Rico, May 2-4, 2016, Conference Track Proceedings*. Sous la dir. d'Yoshua Bengio et Yann LeCun. 2016. url : http://arxiv.org/abs/1511.06434.

[KB17] Diederik P. Kingma et Jimmy Ba. *Adam : A Method for Stochastic Optimization*. 2017. arXiv : 1412.6980 [cs.LG].

[Fra18] Fédération des aveugles de France. « Etude sur l'usage des lecteurs d'écran et des outils et logiciels "basse vision" ». In : (2018). https://www.avh.asso.fr/sites/default/files/rapport_usage_technologies_assistance_mars2018.pdf.

[Vie+18] Valentin Vielzeuf et al. *The Many Moods of Emotion*. 2018. arXiv : 1810.13197 [cs.NE].