

Elève-ingénieur :

Tugdual Le Pen
Imagerie Numérique
2^{ème} année du cursus ingénieur

IRISA

263 Avenue du Général Leclerc
35000 RENNES - France
contact@irisa.fr

Tuteur universitaire :

Pierre Maurel
Enseignant chercheur

Tuteur organisation :

Olivier Le Meur
Enseignant chercheur

ÉTUDE ET MODÈLE PRÉDICTIF DE LA SAILLANCE SUR DES ŒUVRES D'ART

Année universitaire 2019 - 2020

Remerciements

Je tiens dans un premier temps à exprimer ma gratitude à l'IRISA et plus particulièrement à l'équipe Percept pour m'avoir accueilli et considéré en tant que collaborateur durant ces six mois de stage.

Je remercie mon tuteur Olivier Le Meur pour sa pédagogie, sa confiance et son savoir-faire qui m'ont permis d'avancé sur mon projet serainement et efficacement.

Merci également aux doctorants et ingénieurs de l'équipe Percept avec qui j'ai pu échanger des bons moments et des conseils précieux pour le développement de mon projet.

Je désire aussi aussi remercier les professeurs de l'Ecole Supérieure d'ingénieurs de Rennes, qui m'ont fourni les outils nécessaires au bon déroulement de mon stage. Je tiens à remercier spécialement Pierre Maurel mon professeur référent universitaire.

Enfin, pour conclure, je souhaiterais remercier toutes les personnes qui ont participé de différentes façons à la réussite de mon stage.

Résumé

Pour valider ma 4^{ème} année de mon cycle ingénieur en Technologie de l'Information avec spécialité Imagerie Numérique, j'ai effectué un stage d'une durée de six mois dans l'Institut de Recherche en Informatique et Systèmes Aléatoires (IRISA). C'est un laboratoire de recherche impliqué dans le domaine de l'informatique et des technologies de l'information. Il couvre l'ensemble des thématiques de ces domaines, de l'architecture des ordinateurs et des réseaux à l'intelligence artificielle en passant par le génie logiciel, les systèmes distribués et la réalité virtuelle.

J'ai rejoint plus précisément l'équipe Percept (2018) qui est spécialisée dans le comportement visuel de différentes populations. L'un des projets de cette équipe est d'étudier la saillance dans les peintures. Notamment la capacité de déterminer cette saillance automatiquement au moyen de machine learning.

Mon objectif est de participer à ce projet et mettre en place des applications qui permettraient de montrer les possibilités d'utilisations de ce genre de programme.

To validate my 4th year of my engineer cycle specializing in Digital Imaging, I did a six-month internship in the Research Institute in Computer Science and Random Systems (IRISA). It is a research laboratory involved in the field of computer science and information technology. It covers all the themes of these fields, from the architecture of computers and networks to artificial intelligence, including software engineering, distributed systems and virtual reality.

I joined the team Percept (2018) which specializes in the visual behavior of different populations. One of the projects of this team is to study the saliency in paintings. In particular the ability to determine this salience automatically by means of machine learning.

My objective is to participate in this project and set up applications which allow us to show the possibilities of uses of this kind of program.

Sommaire

| | |
|----------------------------------------------------------------------|-----------|
| Remerciements | 2 |
| Résumé | 3 |
| I Introduction | 5 |
| II IRISA | 7 |
| III Contexte | 8 |
| III.1 Fixation et saccade | 8 |
| III.2 Saillance et carte de saillance | 9 |
| III.3 Base de données et oculométrie | 11 |
| IV Présentation et premiers programmes | 12 |
| IV.1 Présentation pour la journée du patrimoine de l'IRISA | 12 |
| IV.2 Vidéo fondu | 13 |
| IV.3 Vidéo chemins visuels | 13 |
| V Modèles de saillance | 14 |
| V.1 État de l'art et choix du modèle de saillance | 14 |
| V.2 Entrainement du modèle | 16 |
| VI Applications | 18 |
| VI.1 Recherche d'application | 18 |
| VI.2 Effet Ken burns | 19 |
| VI.3 Sous-titre descriptif | 21 |
| VII Conclusion | 23 |
| Bibliographie | 24 |

I. Introduction

La peinture et le regard de l'Homme ont toujours eu un lien étroit. En effet chaque spectateur regardera un tableau d'une manière différente de son voisin parce que chaque individu a sa propre culture, son propre point de vue... Pourtant la structure d'une peinture amènera le spectateur à suivre un sens de lecture. Celui-ci sera généralement commun à tous les spectateurs. Par exemple un individu qui découvre le tableau de La Joconde pour la première fois regardera presque systématiquement en premier lieu le visage de Mona Lisa et particulièrement ces yeux qui ont un effet particulier. Rare sont les personnes qui commenceront par identifier les éléments du décor en arrière-plan de la peinture.



Image I.1 – La Joconde de Leonard de Vinci

Ce sont l'ensemble de ces éléments qui attirent l'œil humain qui constituent la saillance. C'est un élément important pour de nombreux domaines. On pense notamment au domaine du marketing et de la publicité qui doivent créer des affiches ou des spots publicitaires avec pour objectif d'attirer le plus possible l'attention et le regard des consommateurs.

La saillance dans la peinture permet d'analyser et de comprendre le regard humain ainsi que toutes les particularités qui en découlent. L'équipe Percept, équipe de recherche du laboratoire de l'IRISA, se penche sur le sujet et notamment à l'automatisation pour déterminer la saillance dans les peintures à l'aide de modèles de réseaux de neurones basés sur le machine learning.

I. Introduction

C'est là que mon sujet de stage intervient. Cela consiste dans un premier temps à faire l'état de l'art des différents modèles de saillance qui existent sur des images naturelles. Dans un second temps le but est d'adapter le meilleur modèle pour qu'il s'adapte à des peintures. Et enfin à partir des résultats de ce modèle trouver des applications visuelles et ludiques pour montrer l'intérêt d'un tel modèle.

Ce stage qui m'as été proposé par Olivier Le Meur correspondait à ce que je recherchais. C'est-à-dire un stage basé sur le machine learning, qui fait suite à mon projet industriel à l'ESIR qui consistait à générer des visages au moyen de réseau de neurones antagoniste génératif (GAN). Mais aussi un stage varié qui puisse me permettre de me former sur plusieurs compétences différentes.

II. IRISA

L'IRISA - Institut de Recherche en Informatique et Systèmes Aléatoires - est aujourd'hui le plus grand laboratoire de recherche français (+ de 850 personnes) dans le domaine de l'informatique et des technologies de l'information. Il couvre l'ensemble des thématiques de ces domaines, de l'architecture des ordinateurs et des réseaux à l'intelligence artificielle en passant par le génie logiciel, les systèmes distribués et la réalité virtuelle.

L'IRISA, créé en 1975, est issu d'une volonté de collaboration entre huit établissements tutelles pluridisciplinaires : CentraleSupélec, CNRS, ENS Rennes, IMT Atlantique, Inria, INSA Rennes, Université Bretagne Sud, Université de Rennes 1. Il est aujourd'hui dirigé par Jean Marc Jézéquel.



Image II.1 – Logo de l'IRISA

l'IRISA est présent sur 3 sites géographiques au sein du territoire breton (Rennes, Lannion et Vannes). Mon stage s'est déroulé dans les locaux de Rennes. Le laboratoire est structuré en sept départements scientifiques :

- D1 - Systèmes Large Échelle
- D2 - Réseaux, Télécommunication et Services
- D3 - Architecture
- D4 - Langage et génie logiciel
- D5 - Signaux et Images numériques, Robotique
- D6 - Média et interactions
- D7 - Gestion des données et de la connaissance

L'équipe PERCEPT du département Média et interactions est spécialisé dans le comportement visuel de différentes populations.

III. Contexte

Le stage a commencé par de la documentation en rapport avec le sujet de stage. N'ayant pas d'accès à un poste la première semaine de stage, Olivier m'a donné quelques documents introduisant des notions importantes sur le regard humain et la saillance. Ce sont des domaines plutôt liés à l'anatomie et la psychologie mais qu'il est important de comprendre si l'on veut pouvoir interpréter les résultats obtenus en sortie des scripts.

Je vais vous expliquer ici les notions fondamentales qui permettent de comprendre le regard humain et comment il est possible de le mesurer pour l'analyser.

III.1. Fixation et saccade

Le regard est une alternance entre des périodes où l'œil reste relativement stationnaire, que l'on appelle "**fixations**", et de courtes périodes de plus grande mobilité, que l'on appelle "**saccades**"[3]. Ce sont des notions qui ont été décrites pour la première fois en 1879 par Javal et Lamare. Il a été possible d'établir des mesures sur le mouvement des yeux deux décades plus tard (Erdmann et Dodge 1898). Ces mesures ont ouvert le champs aux possibilités d'expérimentation sur la psychologie liée au mouvement des yeux. Cela a permis de mieux comprendre le processus d'analyse quand quelqu'un lit, résoud un problème, regarde un film ou quand quelqu'un regarde une peinture.

Chaque fixation est reliée à une autre fixation par une saccade. Ainsi le regard d'une personne dans le temps est donc constitué d'une succession de fixations et de saccades qui forment une chaîne. On appelle cela le **chemin visuel**. C'est en analysant le chemin visuel que l'on est capable de comprendre comment un individu regarde une peinture ou tout autre élément visuel.

Sur l'image III.1 (*Avenue of trees in a small town*, Alfred Sisley, 1866) on peut voir l'exemple d'une représentation de chemins visuels de trois observateurs différents distingués chacun par une couleur différente. Sur cette image chaque fixation est représentée par un cercle numéroté qui correspond à son positionnement dans le parcours du regard. La taille des cercles dépend de la durée de la fixation en question. Ici chaque fixation est reliée à une autre fixation par un trait qui représente une saccade.



Image III.1 – Exemple de chemins visuels de 3 observateurs différents

III.2. Saillance et carte de saillance

Sur l'exemple précédent il est facile de remarquer que chaque individu aborde la peinture avec un regard différent d'un autre. Cependant il est très important de noter qu'il y a des similarités dans les zones regardées. Il y a des endroits de la peinture que les trois observateurs ont regardé, comme le bout du chemin ou l'arbre de gauche au premier plan, tandis que d'autres endroits sont ignorés, les bordures de la peinture entre autre. Ce sont ces zones plus souvent regardées que l'on appellera des zones saillantes.

La **saillance** est donc la notion qui définit qu'un élément est facilement remarqué. Elle existe aussi dans le domaine sonore ou linguistique. Ici c'est évidemment la saillance visuelle qui nous intéresse. Un élément dit saillant est donc un élément qui dénote du reste de l'œuvre et qui attirera le regard de l'observateur. Les critères qui définissent la saillance d'un objet sont régies par deux facteurs importants. Le facteur "**Bottom-up**" basé sur les informations simples comme les couleurs, le contraste ou la luminosité. Le facteur "**Top-down**" lui est basé sur des informations propres à l'observateur. Il peut être influencé par une tâche à accomplir, par sa culture, ses connaissances, son âge... C'est à partir de tous ces éléments que l'on peut déduire une forte connexion entre le mouvement des yeux, ou le chemin visuel, et la saillance d'une peinture.

De cette relation on va pouvoir, à partir du chemin visuel de plusieurs observateurs, générer une **carte de saillance** (voir image III.2). Celle-ci va nous permettre de mettre en évidence les éléments saillants d'une image. Elle est obtenue en moyennant les points de fixations de différents observateurs. Plus une zone est blanche, plus elle est saillante.

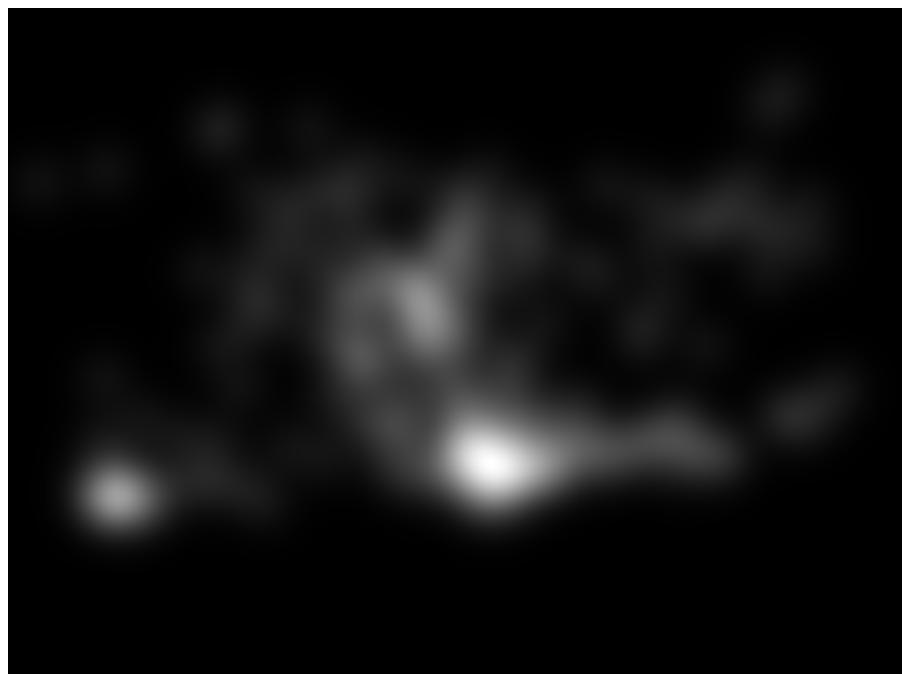


Image III.2 – Exemple de carte de saillance

Si on superpose la carte de saillance et la peinture associée on peut facilement voir quels sont les éléments saillants de l'œuvre (voir image III.3). Ici la carte de saillance nous révèle que le village au bout du chemin, les deux personnages en bas à gauche et la végétation au premier plan sont les éléments saillants de l'œuvre.



Image III.3 – Carte de saillance et peinture superposées

Mon but lors de ce stage sera d'entrainer un réseau de neurones capable de générer une carte de saillance avec comme seule entrée une peinture.

III.3. Base de données et oculométrie

Afin de pouvoir entraîner un réseau de neurones il nous faut une base de données avec des données oculométriques sur des peintures. C'est-à-dire une base de données avec l'enregistrement du chemin visuel de différents observateurs qui regardent une peinture.

La mesure du mouvement des yeux d'une personne est possible grâce à un **oculomètre** (voir image III.4). Cet appareil retranscrit le regard d'un observateur avec une grande précision et peut nous donner des informations comme la position et la durée d'une fixation. Inventé vers la fin du 19^{ème} siècle, il en existe aujourd'hui des versions électroniques très efficace.



Image III.4 – Exemple d'oculomètre

Des étudiants de l'ISTIC ont effectué un stage sous la direction d'Olivier Le Meur pour constituer une base de données oculométrique de 21 observateurs sur un total de **150 peintures**. Ce sont des peintures de 5 grands mouvements artistiques (Fauvisme, Impressionnisme, Réalisme, Romantisme et Pointillisme) avec chacun 3 genres (Nature morte, Nu, Paysage). Chaque participant avait 5 secondes pour regarder une peinture ce qui nous donne des données oculométrique étaler de 5 secondes dans le temps.

IV. Présentation et premiers programmes

IV.1. Présentation pour la journée du patrimoine de l'IRISA

Olivier m'a proposé de réaliser **un diaporama** pour le stand de l'équipe Percept lors de la journée du patrimoine à l'IRISA programmé le 24 Mars 2020 à l'origine mais qui a été reportée à cause du contexte actuel de situation sanitaire d'urgence. J'ai tout de même réalisé la présentation et devrait être diffusée à la prochaine journée du patrimoine. Olivier m'as conseillé de réaliser le diaporama en **LATEX** qui est un langage de programmation qui permet de réaliser des documents équivalents à ceux édités sous Word ou Open Office. L'avantage de ce langage est qu'il permet d'automatiser beaucoup d'élément, notamment la mise en page. Dans ma présentation je devais mettre une dizaine d'œuvres avec pour chacune d'elle une bonne quantité d'information (description, carte de saillance...). **LATEX** m'a permis d'automatiser la mise en diapositive des peintures pour un rendu de qualité.

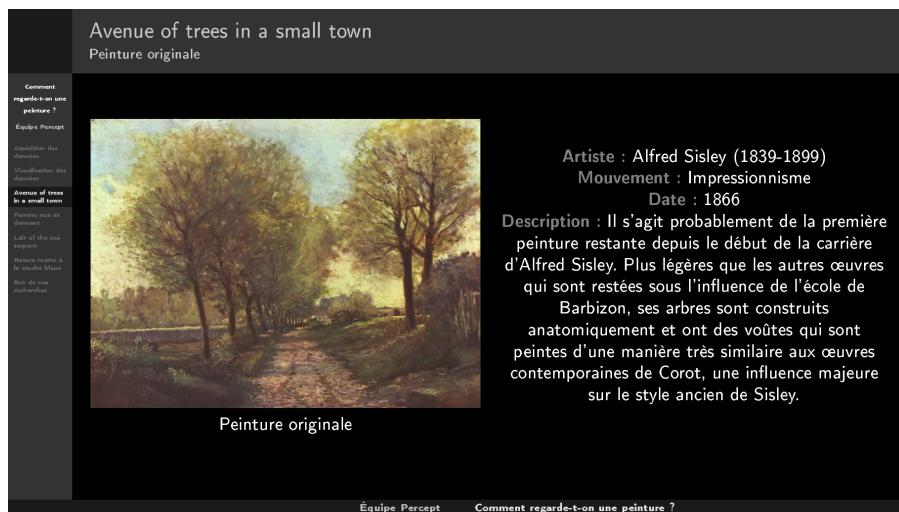


Image IV.1 – Extrait de la présentation

Afin de rajouter des animations dans mon diaporama, notamment des vidéos pour faciliter la compréhension, je me suis lancé dans la programmation de petits programmes en langage Python. Même si il y a eu de nombreux TPs en Python lors de cette année à l'ESIR j'ai remarqué que j'avais encore beaucoup à apprendre en regardant ce qu'il se faisait déjà sur le gitlab de l'équipe Percept (site web qui permet d'échanger et de sauvegarder facilement des documents). Cela m'a aussi permis de me familiariser avec les différentes données présentes dans la base de données oculométrique.

IV.2. Vidéo fondu

Le premier script permet de créer une vidéo avec une **transition en fondu** entre chaque image donnée en entrée du programme. L'intérêt d'un tel script est de le combiner au résultat d'un autre programme du gitlab qui donnait des cartes de chaleur de saillance. Ce sont des cartes de saillance en couleur où les zones saillantes sont représentées par des couleurs chaudes. Cela permet de faire évoluer la carte de chaleur de saillance en fonction du temps d'observation (voir Image IV.2). Pour ma présentation j'ai généré des cartes de chaleur au bout de 1 seconde d'observation, puis au bout de 2, etc. jusqu'à 5 secondes d'observation. La vidéo permettait donc d'enchaîner les différentes cartes de chaleur avec un rendu propre avec pour but final d'analyser l'évolution de la saillance au cours du temps.

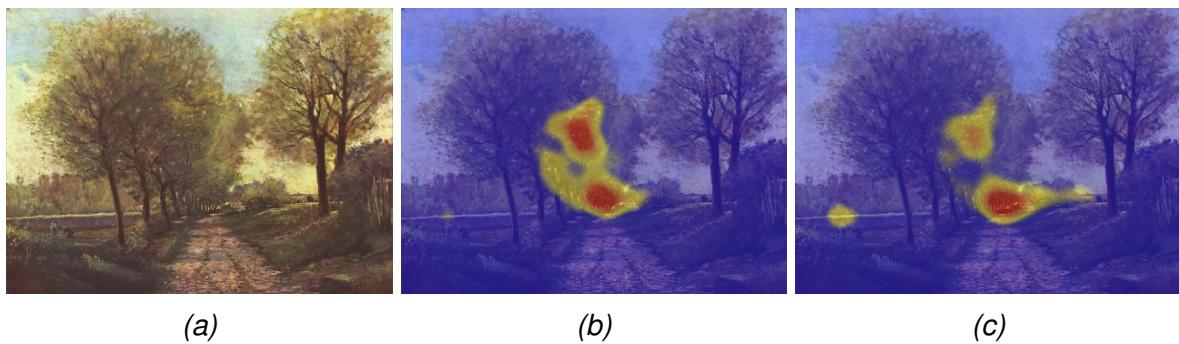


Image IV.2 – (a) Peinture originale, (b) Carte de chaleur de saillance après 2s d'observation, (c) Carte de chaleur de saillance après 5s d'observation

Ce programme n'est pas très compliqué en soit mais il m'a permis de découvrir des librairies sur Python très utiles que je n'avais jamais essayé auparavant.

IV.3. Vidéo chemins visuels

Mon deuxième script consistait à afficher les chemins visuels représentés par des cercles et des lignes comme vu précédemment avec l'image III.1. Cela permet de visualiser le mouvement des yeux de l'observateur et de pouvoir suivre son regard. Les défauts du résultat obtenu c'est qu'avec plusieurs observateurs l'image était rapidement surchargée par les chemins visuels qui se superposaient et les couleurs choisies aléatoirement pour différencier les observateurs pouvaient être très similaires.

J'ai donc fait évolué ce script en y ajoutant de l'animation. J'ai rajouté une option qui permettait de générer une vidéo où chaque cercle (donc chaque fixation) s'affichait les uns après les autres. Cela permet donc d'avoir en fin de vidéo une image qui reste surchargée mais comme le spectateur a suivi le déroulement de l'animation, celui-ci est beaucoup moins confus. Ce point est aussi vrai pour l'inconvénient des couleurs trop similaires.

V. Modèles de saillance

V.1. État de l'art et choix du modèle de saillance

Il existe aujourd’hui de nombreux modèles de saillance qui permettent de **générer des cartes de saillance** à partir de scènes naturelles. Le gros avantage de ces modèles est qu’ils permettent d’éviter la fastidieuse tâche de récupération des données oculométriques sur des humains tout en ayant des résultats proches d’une carte de saillance humaine.

Il y a aujourd’hui deux types de modèles de saillance : les modèles "**fait-mains**", qui appliquent des traitements sur l’image suivant des fonctions mathématiques, et les modèles basés sur **l’apprentissage profond**, qui s’entraînent sur des bases de données pour s’améliorer. Les modèles fait-mains sont en général plus anciens et précèdent l’avènement du machine learning et de l’apprentissage profond. Ils sont donc généralement moins puissants que les modèles profonds.

Ici notre objectif est de déterminer parmi les modèles qui existent quel est le modèle qui obtient les meilleurs résultats quand on lui donne des peintures en entrée. Pour pouvoir comparer le plus objectivement possible les résultats, il est nécessaire d’utiliser des **métriques de qualité**. De la même manière que le mètre est utilisé pour mesurer une distance, les métriques sont des outils de mesure avec des échelles variées qui permettent d’associer un score jugeant la qualité de nos cartes de saillances. Il est préférable d’utiliser plusieurs métriques différentes puisque chacune d’entre elles a ses qualités et ses défauts. Ici les métriques utilisées sont celles du benchmark du MIT [4].

Mon rôle ici a été de tester tous les modèles profonds pour vérifier dans un premier temps s’il était possible de les faire tourner et dans un second temps de donner des peintures en entrées pour pouvoir y appliquer les métriques de qualité. On peut voir dans l’image V.1 les résultats des différents modèles comparés à la carte de saillance originale.

Visuellement il est assez évident de dire que les modèles profonds sont plus proches de la carte de saillance humaine que les autres. Cela se confirme dans l’analyse des métriques. Dans le tableau V.1 on voit que les scores moyens des modèles fait-mains sont tout le temps moins bons que les modèles profonds. Les scores en gras sont les meilleurs scores entre tous les modèles. Ici SAM-ResNet est clairement le plus performant avec le meilleur score

dans cinq des sept métriques utilisées. C'est logiquement qu'avec Olivier on a choisi de continuer nos recherches avec SAM-ResNet.

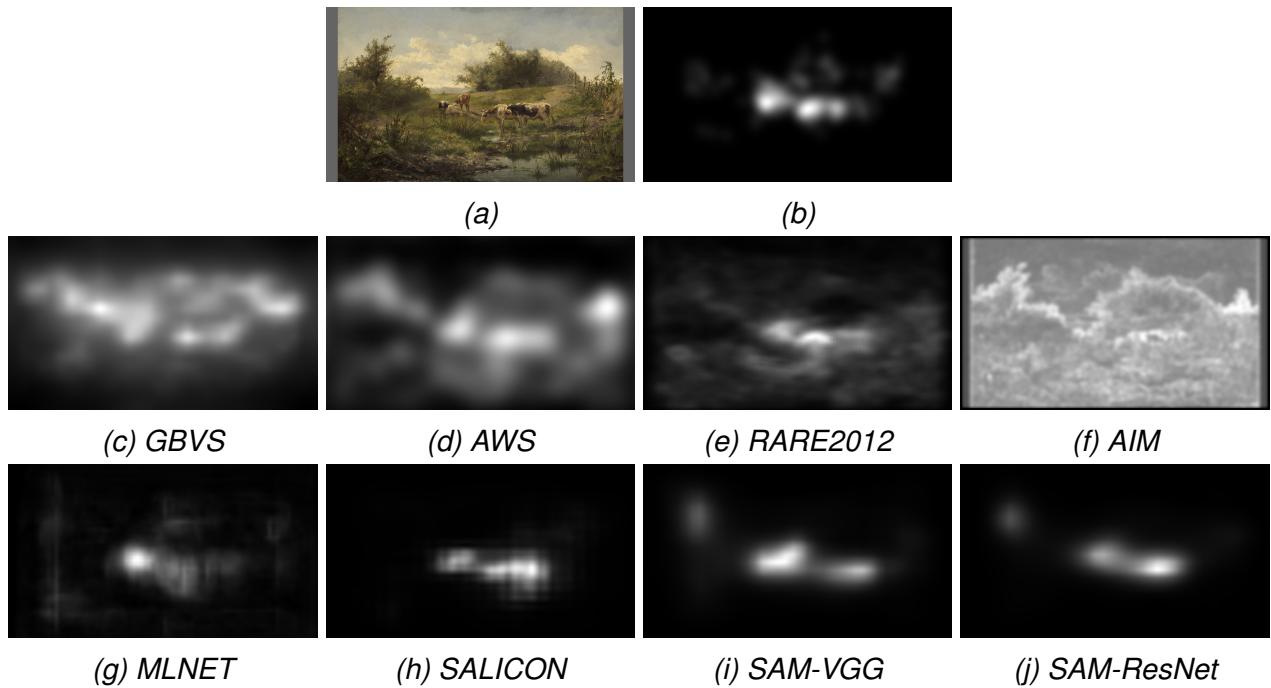


Image V.1 – Cartes de saillance de modèles fait-mains (2^{ème} ligne) et profonds (3^{ème} ligne). La 1^{ère} ligne illustre le stimuli originale et sa carte de saillance humaine(Cows at a pond, Bilders, 1856)

| Modèle | CC ↑ | KL ↓ | SIM ↑ | NSS ↑ | AUC-B ↑ | AUC-J ↑ |
|------------|--------------|--------------|--------------|--------------|--------------|--------------|
| GBVS | 0.506 | 0.962 | 0.446 | 1.256 | 0.809 | 0.817 |
| RARE2012 | 0.443 | 1.020 | 0.438 | 1.103 | 0.777 | 0.786 |
| AIM | 0.315 | 1.245 | 0.371 | 0.772 | 0.723 | 0.735 |
| AWS | 0.427 | 1.045 | 0.430 | 1.083 | 0.762 | 0.769 |
| Moyenne | 0.422 | 1.068 | 0.421 | 1.053 | 0.774 | 0.776 |
| MLNET | 0.576 | 0.832 | 0.513 | 1.524 | 0.770 | 0.818 |
| DeepGazell | 0.485 | 0.896 | 0.488 | 1.394 | 0.679 | 0.804 |
| SALICON | 0.538 | 0.880 | 0.517 | 1.445 | 0.708 | 0.827 |
| SAM ResNet | 0.700 | 0.984 | 0.613 | 1.834 | 0.782 | 0.862 |
| SAM VGG | 0.617 | 0.970 | 0.561 | 1.603 | 0.752 | 0.846 |
| Moyenne | 0.583 | 0.912 | 0.551 | 1.560 | 0.738 | 0.831 |

Tableau V.1 – Performances des modèles de saillance sur les peintures de la base de données

V.2. Entrainement du modèle

On a donc un modèle profond pré-entraîné pour déterminer la saillance dans les scènes naturelles. Pour adapter ce modèle aux peintures afin d'obtenir de meilleurs scores et donc une meilleure qualité dans les cartes de saillance générées, on va faire du "**fine tuning**", que l'on peut traduire par "ajustement". C'est-à-dire qu'au lieu d'entraîner le modèle en partant de zéro, on va prendre les poids du modèle actuel et entraîner le modèle à partir de ceux-ci. Si on compare ça à une course, au lieu de commencer la course au départ, on part d'un checkpoint au milieu de la course. Cela permet de gagner du temps lors de l'entraînement sans perdre en qualité.

La base de données utilisée pour l'entraînement est celle décrite dans la partie III.3. Des 150 peintures disponibles, 90 sont utilisées pour l'entraînement, 20 pour la validation et les 40 dernières sont pour le test du modèle. Les peintures étaient dans un premier temps choisies par ordre alphabétique, ce qui était supposément un ordre aléatoire. Il s'est avéré après coup que la plupart des natures mortes ou des peintures nues avait le genre dans leur titre ce qui faussait l'aléatoire. J'ai donc créé un petit script qui m'a permis de faire la séparation de ma base de données de manière complètement aléatoire. Cela a permis d'améliorer la qualité des cartes de saillance générées.

| Modèle | CC ↑ | KL ↓ | SIM ↑ | NSS ↑ | AUC-B ↑ | AUC-J ↑ |
|------------------------|-------|-------|--------|-------|---------|---------|
| SAM ResNet | 0.691 | 1.088 | 0.609 | 1.792 | 0.786 | 0.857 |
| SAM ResNet (fine-tuné) | 0.758 | 0.836 | 0.681 | 1.922 | 0.845 | 0.882 |
| Gain (%) | +9.7% | -23% | +11.8% | +7.2% | +7.5% | +2.9% |

Tableau V.2 – Performances de SAM-ResNet après fine-tuning

Les résultats obtenus après l'entraînement sont plutôt bons. Les scores avant et après des métriques sont visibles dans le tableau V.2. Toutes les métriques montrent un **score supérieur** pour le modèle ré-entraîné. Un exemple des résultats sur les images V.2 montre qu'effectivement sans être une copie conforme de la carte de saillance humaine, la version ré-entraînée de Sam-ResNet donne des résultats corrects.

Cependant il est important de noté que les mouvements artistiques des peintures influencent le résultat du modèle de saillance. En effet les résultats sont bien meilleurs sur des peintures appartenant au mouvement du Réalisme ou du Romantisme que celles du Pointillisme ou de l'Impressionisme. Cela est dû au fait que le modèle étant construit et entraîné originellement pour des scènes naturelles est ce qui se rapproche le plus des peintures réalistes. Pour améliorer les résultats je pense qu'il faudrait ré-entraîner entièrement le modèle avec une plus grande base de données et avoir plus de diversité avec plus de styles et de mouvements différents. C'est un travail qui demande beaucoup de temps et d'organisation notamment pour la collecte de données.

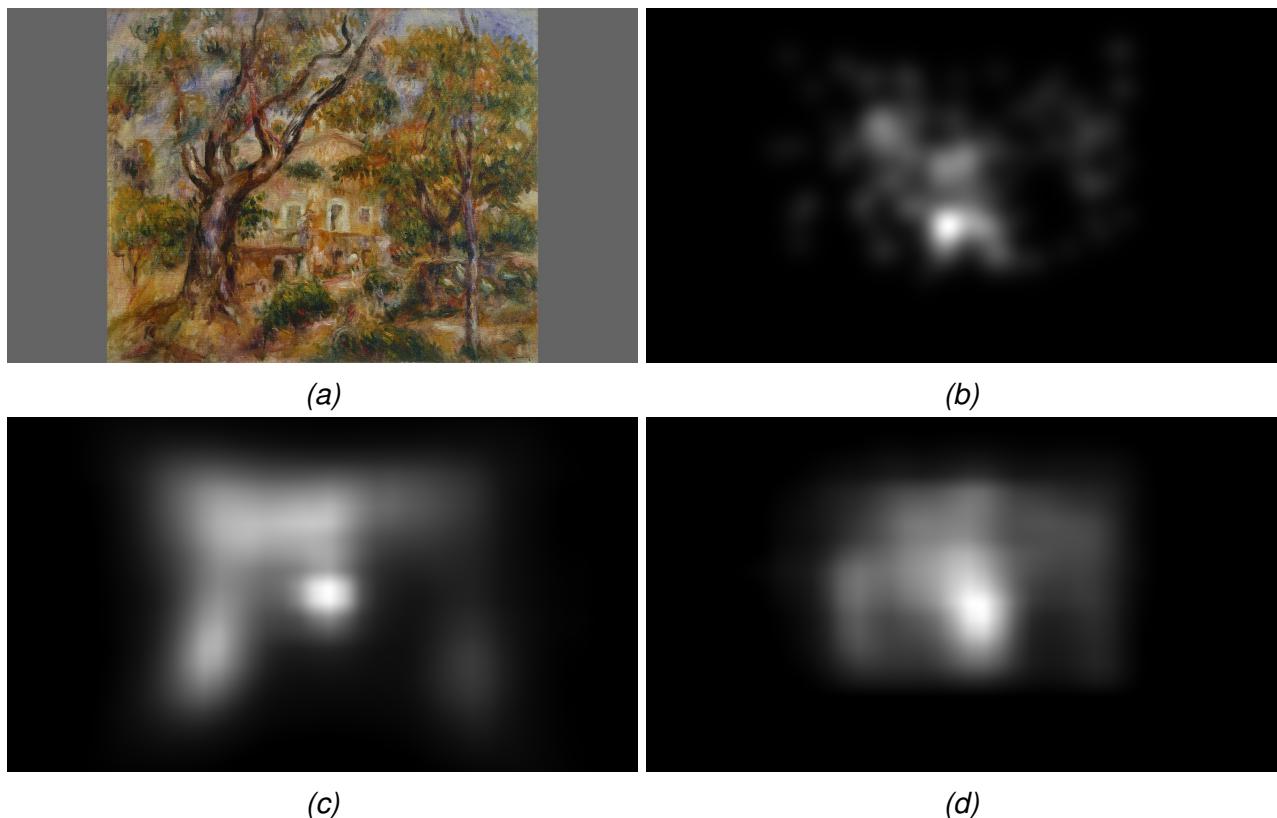


Image V.2 – (a) Peinture originale, (b) carte de saillance humaine, (c) prédiction SAM-ResNet et (d) prédiction de SAM-ResNet ré-entraîné (La ferme des Collettes, Renoir, 1908)

VI. Applications

Les applications possibles à partir des cartes de saillance sont illimitées. Notre problématique ici était de réussir à **animer** nos peintures à partir de la carte de saillance générée par le modèle de saillance. Eakta Jain, professeur assistante à l'université de Floride, a déjà travaillé avec Olivier et a pu nous apporter son expérience grâce aux différents projets d'applications basés sur le mouvement du regard qu'elle a réalisé [5]. Ses projets liés au regard humain sont souvent appliqués aux comics qui a des similarités avec la peinture.

VI.1. Recherche d'application

Afin de déterminer quelle application il serait intéressant de développer, il faut faire une recherche de ce qui existe déjà ou inventer autre chose. Je vais présenter ici les différentes pistes qui ont été envisagées et expliquer notre choix final.

Pour ajouter du mouvement, il existe des logiciels qui permettent de générer des **plotagraphes**. Basé sur les cinéagraphes, qui sont des mélanges d'images et de vidéos, les plotagraphes permettent d'ajouter du mouvement à partir de l'image seule. En revanche il faut ajuster à la main les zones à mouvoir donc difficilement automatisable et le résultat n'est impressionnant que sur les fluides (eau, nuages, fumée...).

Eakta a un projet de **segmentation** des différentes parties d'un comic pour animer les personnes qui parle et ajouter du son [9]. La segmentation est obtenue à partir du chemin visuel aquis par oculométrie. Une idée intéressante mais difficilement adaptable à toutes les peintures car trop détaillée.

Un autre projet d'Eakta appelé "Predicting Moves-on-Stills for Comic Art using Viewer Gaze Data" [6] ajoute un effet de **Ken Burns** sur des pages de comics en fonction des prédictions du mouvement du regard du spectateur. Cet effet intéressant et facile à mettre en place. C'est la piste vers laquelle on se dirigera et que l'on détaillera dans la partie suivante. Il existe aussi une version 3D de cet effet [7] qui est très impressionante mais est compliquée à mettre en place et difficile à mettre en relation avec la carte de saillance.

VI.2. Effet Ken burns

L'effet **Ken Burns** est représenté par un mouvement de caméra (panoramique, zoom ou rotation) sur une image fixe. L'ajout de mouvement et d'animation permet de garder l'attention du spectateur. C'est un effet qui est très régulièrement utilisé dans les documentaires, dans les journaux télévisés ou tout simplement dans les diaporamas photos.

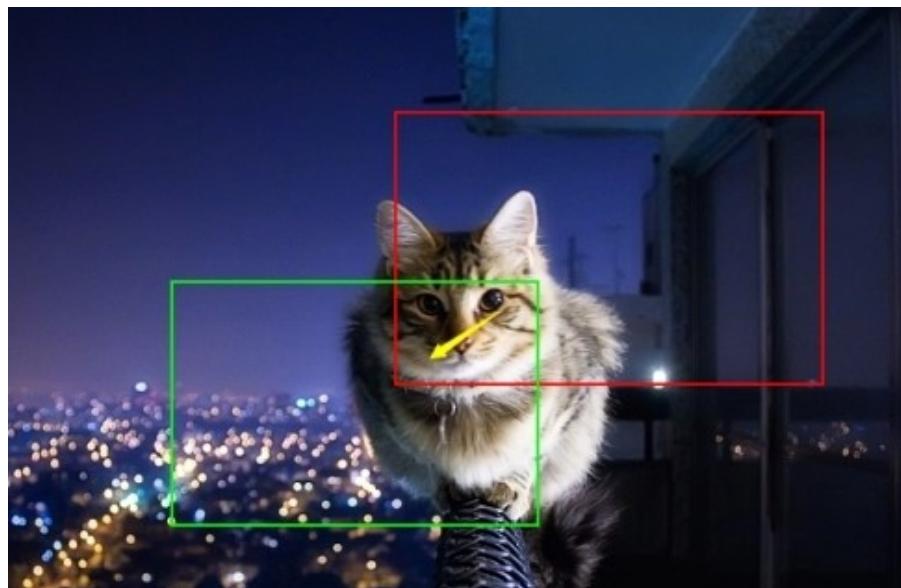


Image VI.1 – Exemple d'effet Ken Burns avec panoramique du cadre rouge vers le cadre vert

J'ai donc décidé de partir sur un effet Ken Burns comme projet d'application. Mon but ici est que le mouvement de la caméra suive celui d'un œil humain. On est capable de générer un chemin visuel à partir d'une carte de saillance grâce à un **modèle saccadique**. J'ai pu en utiliser un créé par Olivier Le Meur [8] disponible sur le gitlab de l'équipe Percept. Ainsi à partir d'une peinture on obtient une carte de saillance grâce au modèle de saillance, puis le chemin visuel grâce au modèle saccadique et enfin un effet Ken Burns qui suit le chemin visuel.

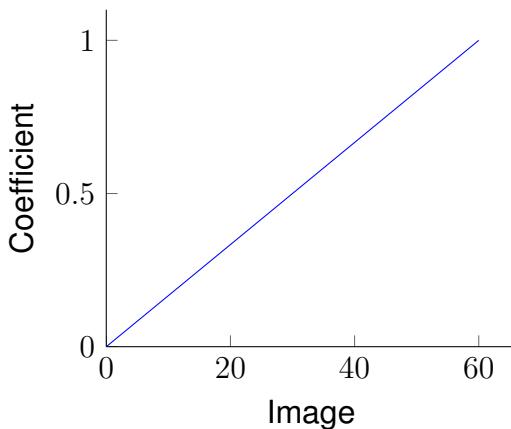
Je me suis donc lancé dans la réalisation d'un programme en Python qui en entrée prend une peinture et le chemin visuel associé pour en sortie obtenir une vidéo avec l'effet Ken Burns. Le principe de base est d'avoir une caméra qui se déplace dans l'image et qui enregistre ses déplacements à la fréquence de la vidéo de sortie.

J'ai décidé ici de faire le même schéma de déplacement pour chaque vidéo. Au début de la vidéo on a une pause sur la peinture entière pour que l'observateur puisse la voir dans son intégralité. Ensuite on fait un zoom x2 sur la première fixation de mon chemin visuel. Ensuite on fait des panoramiques pour passer d'une fixation à une autre. Pour chaque fixation je fais une pause de quelques secondes pour que l'observateur puisse prendre le temps de

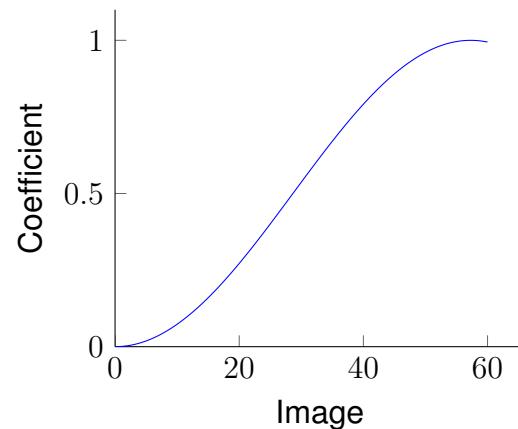
regarder l'œuvre et d'identifier les petits détails. Enfin je fais un dézoom pour retrouver une vue d'ensemble de la peinture.

Je me suis attaqué à la programmation des différents mouvements de caméra, le zoom et le panoramique. J'ai d'abord commencé par créer l'effet de zoom qui consiste à redimensionner l'image tout en gardant à l'identique la résolution de la caméra. Donc par exemple pour un zoom x2 je double la taille de l'image mais je garde celui de la caméra sur les dimensions d'origine. Une fois zoomé, le but est de se déplacer de fixation en fixation en suivant le chemin visuel. Afin d'obtenir des mouvements directs et de pouvoir gérer les bords proprement je précalcule la position finale de ma caméra puis je fais bouger la caméra de mon point d'origine vers la position finale. En fusionnant les deux mouvements on peut créer un zoom sur un point précis de l'image.

Dans un premier temps le mouvement était trop linéaire ce qui a pour effet de donner un rendu final assez pauvre et saccadé. Pour lisser les mouvements j'ai appliqué un effet de ease-in-out. Cela permet d'obtenir une accélération progressive au début du mouvement et une décélération progressive à la fin. Au lieu d'avoir une interpolation linéaire (graphe VI.2a), c'est-à-dire que la caméra se déplace de la même distance entre chaque image d'un plan à un autre, ici on a une interpolation avec de petites distances en début et en fin de parcours mais de grandes distances en milieu de parcours (graphe VI.2b).



(a) Linéaire



(b) Ease-in-out

Image VI.2 – Graphes représentant les coefficients d'interpolations pour 60 images

Et enfin comme les chemins visuels sont souvent composés d'une vingtaine de fixations souvent proches les unes des autres, j'ai décidé de ne pas faire un déplacement de caméra pour chacune d'entre elles. Cela ferait trop de répétition et perdrat le spectateur dans la peinture. A la place je fais en sorte que les fixations présentes sur un certain pourcentage du cadre de la caméra soit ignorées pour le reste du parcours de la caméra.

VI.3. Sous-titre descriptif

Pour améliorer l'application, Olivier, Eakta et moi avons pensé à un concept d'intelligence artificielle qui décrit ce qu'elle voit. L'idée ici serait d'ajouter aux vidéos des **sous-titres automatiques** générés par un réseau de neurones qui décrivent ce qui est représenté. Il en existe qui fonctionne pour des scènes naturelles avec par exemple le modèle NeuralTalk2 de Karpathy [10] ([lien démo](#)).

J'ai testé ce modèle sur les peintures mais le résultats est ridiculement mauvais. Quasiment aucune description ne correspond à l'image comme on peut le voir en exemple de l'image VI.3. Il faudrait donc ré-entrainer le modèle pour pouvoir espérer obtenir de meilleurs résultats. Cependant plusieurs problèmes nous empêche de travailler sur cette solution. Premièrement le programme en lui-même est assez complexe et je n'ai pas réussi à faire tourner la partie entraînement du modèle. Deuxièmement il est très dur de modifier le dataset de mots puisque celui-ci est composé de mots associés à des éléments d'images que seul le modèle peut interpréter. On a donc quasiment aucune maléabilité du dataset. Et troisièmement la fin du stage arrive trop vite pour que je prenne le temps de m'attarder dessus.

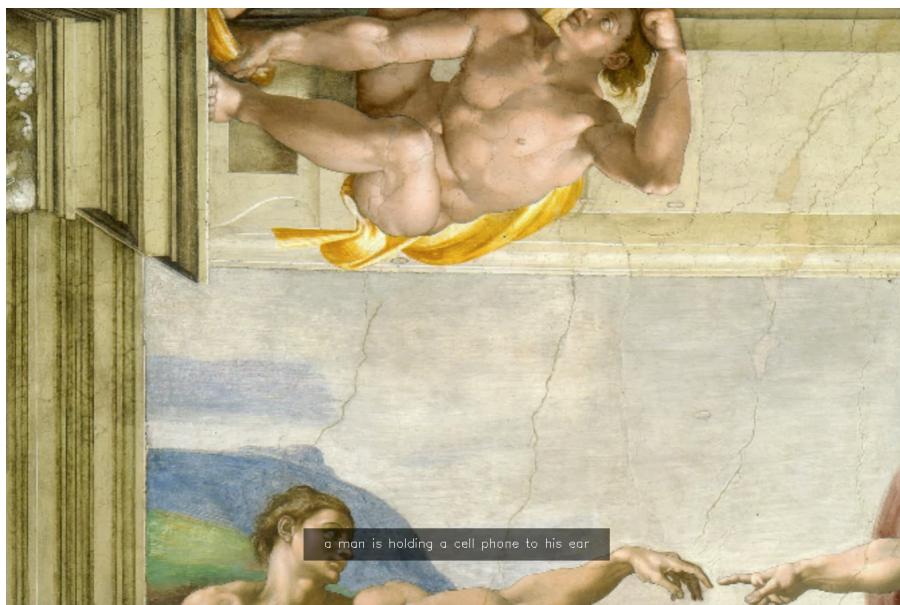


Image VI.3 – Exemple de sous-titre descriptif généré par NeuralTalk2 (Création d'Adam, Michel-Ange, 1512)

On a donc décidé avec Olivier de faire une **preuve de concept**. C'est-à-dire un script prototype qui permettrait de prouver qu'il est possible d'utiliser des sous-titres générés automatiquement sur ces peintures et les intégrer à notre vidéo. J'ai donc modifié mon programme qui génère des vidéos pour pouvoir insérer du texte en bas de l'image. Cela m'a permis d'obtenir le résultat visible sur l'image VI.3. Le programme de la preuve de concept se déroule en trois étapes. En premier je pré-génère les images lorsque la vidéo est en pause. Ensuite je rentre ces images dans le modèle de sous-titrage pour obtenir des

images avec des descriptions. Enfin je génère la vidéo en ajoutant mes images sous-titrées lors des pauses de ma vidéo.

Afin de pouvoir diffuser ces travaux sur les réseaux sociaux (LinkedIn ou Medium), j'ai ajouté à la pause du début de vidéo un sous-titre qui dit "What does an AI see in this paintings?", traduit par "Que voit une IA dans cette peinture?", ainsi que le nom de la peinture et de l'artiste en haut de la vidéo (voir image VI.4). Une question qui intrigue le spectateur et qui lui donne envie de regarder le reste de la vidéo.



Image VI.4 – Début de la vidéo (Venice from the porch of Madonna, Turner, 1835)

VII. Conclusion

Au cours de ces 6 mois de stage au sein de l'IRISA et plus particulièrement l'équipe Percept, j'ai pu découvrir le monde de la recherche. Une expérience unique qui m'as permis de me faire une idée de ce qui m'attend si je me lance dans le développement d'une thèse.

J'ai aussi pu découvrir le domaine de la saillance. Je pense que pour un étudiant qui a pour but de travailler dans l'imagerie numérique il est important de savoir comment fonctionne le regard humain. On a bien sûr déjà eu des cours sur ce sujet à l'ESIR mais ici j'ai pu approfondir mes connaissances sur ce sujet.

J'ai beaucoup aimé l'échange qu'il y a entre les différentes personnes au sein de l'équipe. Dès mon arrivée je me suis senti intégré à ce groupe de travail. Je l'ai aussi ressenti lors des réunions où tous les membres de l'équipe devaient présenter leur projet moi y compris. C'est un très bon moyen de comprendre ce que font les collègues et à l'inverse expliquer mon sujet. Cela permet donc par la suite que chacun puisse proposer des idées ou des solutions aux problèmes d'autres personnes qui bloquent dans leur projet.

Au niveau technique, j'ai beaucoup appris en programmation et particulièrement en Python et en L^AT_EX. Pour Python je n'avais jamais réalisé de projet aussi concret et qui soit ensuite mis à la disposition de l'équipe. L^AT_EX est un langage que j'ai eu la chance d'avoir le temps de maîtriser suffisamment pour obtenir des documents sobres, professionnels et automatiques.

Ce fut aussi intéressant de travailler avec l'apprentissage profond qui est aujourd'hui présent partout. Même si j'avais déjà travaillé sur un projet de machine learning à l'ESIR, j'ai pu renforcer mes connaissances sur le sujet.

Bibliographie

- [1] Site de l'IRISA - Présentation du laboratoire
<https://www.irisa.fr/fr/page/recherche-innovation-sciences-technologies-du-numerique>
- [2] Site de l'IRISA - Présentation de l'équipe PERCEPT
<https://www.irisa.fr/fr/equipes/percept>
- [3] *Art, Aesthetics, and the brain*, 2015
par J.P. Huston, M. Nadal, F. Mora, L.F. Agnati et C.J. Cela-Conde
- [4] *MIT saliency benchmark*, 2015
par Bylinskii Z, Judd T, Borji A, Itti L, Durand F, Oliva A, et al.
<http://saliency.mit.edu/>
- [5] Projet "Eye-tracking and Comics" de Eakta Jain
<https://jainlab.cise.ufl.edu/comics.html>
- [6] *Predicting Moves-on-Stills for Comic Artusing Viewer Gaze Data*, 2015
E. Jain, Y. Sheikh, J. Hodgins
<https://jainlab.cise.ufl.edu/documents/motion-comics-cga2015.pdf>
- [7] *3D Ken Burns Effect from a Single Image*, 2019
S. Niklaus, L. Mai, J. Yang, F. Liu
<https://arxiv.org/pdf/1909.05483.pdf>
- [8] *Saccadic model of eye movements for free-viewing condition*, 2015
O. Le Meur, Z. Liu
<https://www.sciencedirect.com/science/article/pii/S0042698915000504>
- [9] *Creating Segments and Effects on Comics by Clustering Gaze Data*, 2017
I. Thirunarayanan, K. Khetarpal, S. Koppal, O. Le Meur, J. Shea, E. Jain
https://jainlab.cise.ufl.edu/documents/REQGazeComics_preprint.pdf
- [10] Neuraltalk2 code
A. Karpathy
<https://github.com/karpathy/neuraltalk2>