

Prédiction de la saillance et modélisation des saccades oculaires

*Hakima Arroubat
Majda Ennaciri
Nezha Najih
Safae El-omari*

Encadré par :

M. Emmanuel Daucé

Remerciement :

Tout d'abord nous tenons à remercier notre encadrant M. Emmanuel Daucé pour avoir été disponible pour répondre à nos questions, et pour son accompagnement dans ce projet avec patience, en donnant des remarques et conseils judicieux qui ont contribué sûrement à alimenter notre réflexion.

Nous tenons à remercier également le corps professoral de l'Ecole Centrale Marseille pour tout ce qu'il nous a appris, son aide et son accompagnement durant toute l'année.

Table des matières

1. Introduction.....	3
2. Planification du projet	4
3. Théorie des modèles de prédiction de la saillance	
3.1. Génération de carte de saillance.....	5
3.2. Deep Object Features (DeepGaze II)	5
3.3. Fonction de contraste d'intensité.....	6
3.4. IttyKoch.....	7
3.5. Grad-Cam.....	8
3.6. Métriques	
d'évaluation.....	10
4. Etude de cas	
4.1. Implémentation du DeepGaze.....	12
4.2. La fixation Séquentielle.....	13
4.3. Comparaison et évaluation.....	14
5. Conclusion.....	19
Bibliographie.....	20

I. Introduction

Au cours de la dernière décennie, les modèles de détection de la saillance des images ont fait des progrès remarquables. Ces modèles ont connu un grand bond en termes de performances, grâce aux progrès de l'apprentissage en profondeur et données annotées à grande échelle, même s'ils ne parviennent toujours pas à atteindre le niveau humain.

En s'inspirant de système visuel humaine qui fonctionne comme un filtre pour allouer plus d'attention aux régions attrayantes et intéressantes pour un traitement ultérieur, les modèles de prédiction de la saillance essayent de parcourir l'image afin de concentrer l'attention visuelle sur certaines parties d'une image, réduisant ainsi la complexité de l'analyse de la scène. Les systèmes, alimentés par ces modèles, décomposent le problème complexe de la compréhension de la scène en sélectionnant rapidement, de manière efficace, des emplacements remarquables à analyser en détail. En outre, inspirés par le phénomène de perception visuelle, certains modèles de saillance visuelle se concentrent sur la prédiction des fixations humaines

Parmi les approches traditionnelles utilisées, on trouve des méthodes qui sont principalement basées sur des signaux de bas niveau et des fonctionnalités artisanales. D'autre part, certaines autres méthodes utilisent le *center prior* pour améliorer les performances de détection d'objets saillants. Néanmoins, ces méthodes ont une capacité limitée à détecter toute la structure des objets saillants dans des scènes complexes. A cet effet, des méthodes basées sur l'apprentissage en profondeur ont donné un résultat qualitatif saut dans les performances. Ainsi, ils ont été largement utilisés, dans ces dernières années, pour la détection de saillance grâce, notamment, à leur grande capacité à modéliser la sémantique de haut niveau. [1]

L'importance de ces modèles de prédiction de la saillance réside, d'abord, dans le fait qu'ils présentent des prédictions vérifiables qui peuvent être utilisées pour comprendre les mécanismes de l'attention humaine au niveau comportemental et les niveaux neuronaux. Ensuite, ils permettent de transmettre aux machines la capacité d'analyser rapidement des scènes complexes et consacrer leur perception limitée et ressources cognitives aux sous-ensembles les plus pertinents. Le traitement des images par ces modèles, présente une étape de prétraitement efficace pour de nombreuses tâches de vision par ordinateur telles que la classification d'images, la segmentation d'image, segmentation vidéo, retouche d'image et suivi d'objet. De surcroît, ces modèles de saillance peuvent estimer l'attention de l'utilisateur avec des applications au marketing, compression d'images, études sociales, le diagnostic du patient et la surveillance. [2,3]

Dans cet article, nous passons en revue quatre modèles de détection de saillance (Grad-Cam, Ittychokh, Deep gaze II, ICF), résumer les enjeux importants des méthodes existantes, et discuter des problèmes existants. Avant de passer aux cas d'étude pour tester les modèles sur des images de la data CAT2000, et les comparer grâce à des coefficients statistiques (NSS, AUC, MSE), en essayant de répondre essentiellement sur la question : Comment les modèles peuvent-ils être évalués de manière plus fine ?

II. Planification du projet

Tout d'abord, nous avons choisi le sujet de la prédiction de la saillance, nous avons contacté notre tuteur. Il nous a suggéré quelques articles pertinents. Ensuite, nous avons fait des recherches sur le sujet. De plus, nous avons essayé d'analyser les articles relatifs au modèle DeepGaze II. Ensuite, nous nous sommes concentrés davantage sur les codes de ce modèle. Au début, nous avons rencontré des problèmes au niveau de l'exécution. Mais nous avons finalement réussi à les faire fonctionner et à les tester.

Par la suite, nous avons décidé de chercher d'autres méthodes de prédiction de la saillance. Finalement, nous avons choisi de travailler sur Grad-Cam, Ittykoch, et ICF. Après avoir exécuté les modèles, nous avons cherché des données pour les comparer. Nous avons trouvé la base de données CAT2000. Nous les avons d'abord comparés visuellement, puis nous avons utilisé des mesures d'évaluation.

Pour la dernière partie, nous avons commencé les fixations séquentielles, puis nous avons analysé leur performance sur les différents modèles.

Diagramme de GANTT : Annexe 1

III. Théorie des modèles de prédiction de la saillance

1. Génération de carte de saillance

Pour générer les cartes de saillance, la première chose à faire est de détecter les points de fixation, et puis en se basant sur cet ensemble de points, nous pouvons construire une carte de saillance en rendant ces points comme des pixels blanc sur une image noir et après on brouille avec des propriétés particulières conçues en utilisant un filtre gaussien et finalement il faut normaliser pour remettre les à l'échelle entre 0 et 1.

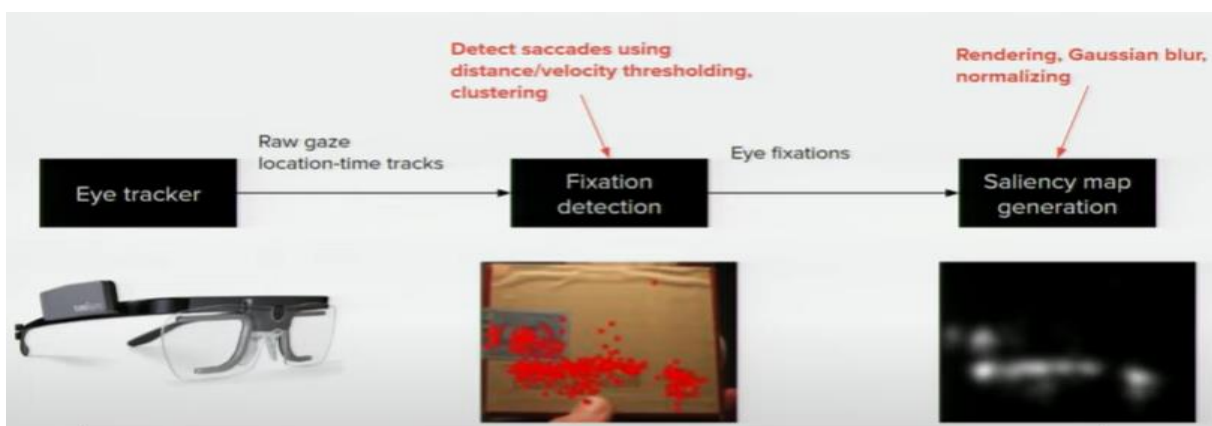


Fig. 1 carte des fixations et de la saillance [source : <https://www.youtube.com/watch?v=L0-Sm8ERvVU&t=253s>]

2. Deep Object Features (DeepGaze II)

DeepGaze II est un modèle de prédiction de la saillance qui est basé sur des caractéristiques profondes qui sont formées sur la reconnaissance d'objet. Dont l'architecture est la suivante :

Tout d'abord, Une image d'entrée donnée est sous-échantillonnée par un facteur 2 et après sera l'entrée du réseau VGG 19 normalisé pour lequel tous les filtres ont été redimensionnés pour produire des cartes d'entités avec une variance unitaire sur le jeu de données ImageNet. Ensuite, les cartes des caractéristiques d'une sélection de haut niveau couches convolutives (conv5_1, relu5_1, relu5_2, conv5_3, relu5_4, sélectionnés par recherche aléatoire) sont sur-échantillonnées par un facteur de 8 de sorte que la résolution spatiale est suffisante pour une prédiction précise. Ces cartes de caractéristiques ont la même taille et peuvent être combinées en un seul tenseur tridimensionnel avec (5×512) canaux, qui est utilisé comme entrée pour un deuxième réseau de neurones ce qu'on appelle le réseau de lecture. Ce réseau a une architecture simple. Il se compose de trois couches de convolutions 1×1 utilisent 16, 32, 2 caractéristiques, ces couches sont suivies de non-linéarités ReLU. Le dernier couche qu'un seul canal de sortie ce qu'on appelle O(x,y). Ce réseau de lecture ne peut représenter qu'une non-linéarité ponctuelle dans les caractéristiques VGG. Ce qui montre qu'il est uniquement capable d'apprendre les interactions sur les canaux et pas sur les pixels. Autrement dit, il ne peut pas apprendre de nouvelles caractéristiques spatiales. Le modèle est pré-entraîné à l'aide de la Dataset SALICON et, comme critère d'arrêt on utilise la Dataset MIT1003.

La sortie de réseau de lecture est convolée avec une gaussienne afin de régulariser les prédictions.

$$\mathbf{S}(\mathbf{x},\mathbf{y})=\mathbf{O}(\mathbf{x},\mathbf{y})*\mathbf{G}_{\sigma}$$

Puisque Les fixations ont tendance à être proches du centre de l'image le fait qui dépend de la tâche et de l'ensemble de données. Nous modélisons explicitement le biais central comme une distribution a priori ajoutée à S :

$$\mathbf{S}'(\mathbf{x},\mathbf{y}) = \mathbf{S}(\mathbf{x},\mathbf{y}) + \log \mathbf{p}_{baseline}(\mathbf{x},\mathbf{y})$$

De plus, on utilise une estimation de noyau gaussien sur toutes les fixations à partir de jeu de données d'entraînement.

Enfin, $\mathbf{S}'(\mathbf{x},\mathbf{y})$ est converti en une distribution de probabilité sur l'image au moyen d'un Softmax.[\[4\]](#)

$$\mathbf{p}(\mathbf{x},\mathbf{y})=\frac{\exp(\mathbf{S}'(\mathbf{x},\mathbf{y}))}{\sum_{\mathbf{x},\mathbf{y}} \exp(\mathbf{S}'(\mathbf{x},\mathbf{y}))}$$

3. Fonction de contraste d'intensité (ICF) :

L'architecture du modèle ICF de bas niveau est la même que DeepGaze II (Fig. 1), sauf qu'on remplace les caractéristiques VGG qui ont été formées sur reconnaissance d'objets par un espace de caractéristiques qui ne peut extraire d'image que des informations purement de bas niveau (intensité et intensité contraste).

Tout d'abord on sous-échantillonne l'image par un facteur de 2 et projeter les canaux de couleur RVB sur leurs principales composantes pour les images naturelles (calculés sur le jeu de données MIT100), qui donne le canal de luminance et deux canaux de couleur.

Pour chacun de ces canaux, on calcule indépendamment l'intensité et le contraste locaux à différentes échelles spatiales. Pour l'intensité locale, on calcule simplement une pyramide gaussienne avec 5 échelles différentes. Les écarts-types des noyaux gaussiens sont de 5,10,20,40,80 px et la taille de la fenêtre est de 171 px. On utilise le rembourrage le plus proche pour que la carte d'entités en sortie a les mêmes dimensions spatiales que la carte d'entités en entrée.

Pour le contraste local, on calcule d'abord 5 résidus passe-haut en soustrayant chaque niveau de la pyramide gaussienne du canal d'entrée. Puis, on met au carré ces résidus pour calculer le contraste pixel par pixel, et enfin on brouille les carrés des résidus avec le même noyau gaussien qui a servi à calculer le résidu (Fig. 2). Cette procédure donne 5 cartes de caractéristiques d'intensité et 5 de contraste pour chaque canal d'entrée, et donne ainsi 30 cartes de caractéristiques qui constituent l'entrée du réseau de lecture. Le réseau de lecture et les étapes qui suivent sont pareil à DeepGaze II. [\[4\]](#)

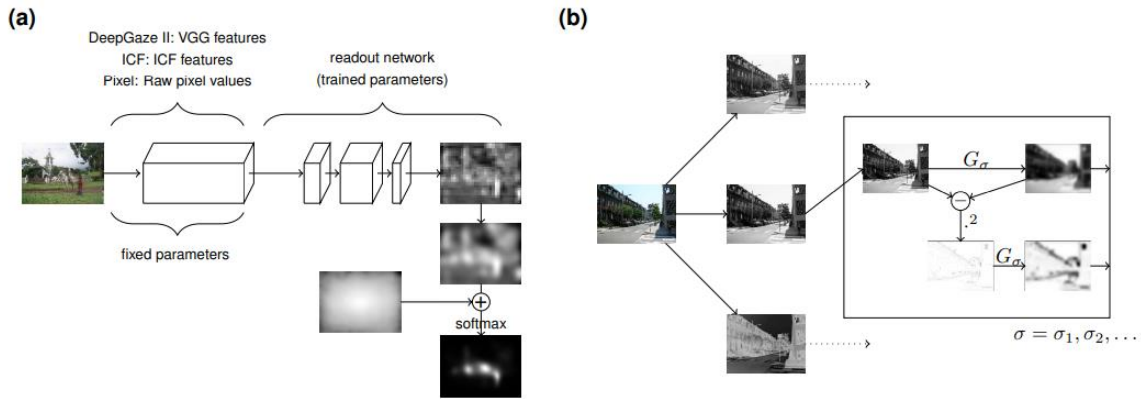


Fig.2 :(a) l'architecture des modèles : Chaque modèle dispose d'un espace de fonctions fixe qui alimente le réseau de lecture : DeepGaze II utilise caractéristiques VGG-19 et le ICF utilise un contraste local simple à différentes échelles. Les activations de ces caractéristiques sont transmises au deuxième réseau de neurones qui s'appelle réseau de lecture. Il en résulte une carte de saillance. (b) l'espace de caractéristique de ICF. [4]

4. Itty Koch

Le modèle de saillance Itty Koch utilise d'abord un extracteur de feature pour convertir une image en cartes de caractéristiques. Ensuite les features des images multi-échelles sont combinées en une seule carte de saillance topographique.

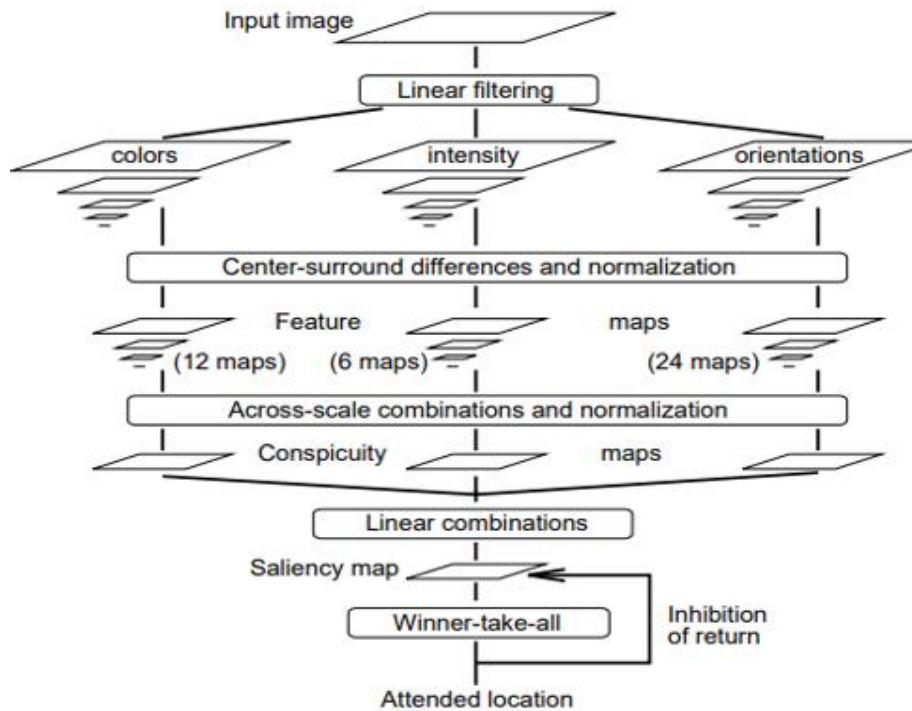


Fig. 3 L'architecture générale de modèle ITTYCOCH

En premier temps on applique des filtres linéaires afin d'extraire les caractéristiques d'intensité, de couleur et d'orientation de l'image. Le modèle extrait une carte d'intensité comme une carte caractéristique des points brillants dans l'image. Car ils sont plus susceptibles de déclencher une réponse de saillance. Ensuite, pour la paramétrisation de l'orientation, Elle comprend l'angle (filtre rotation) θ , le décalage de phase ψ , la longueur d'onde λ , l'écart-type σ et l'obliquité γ . Et il utilise des pyramides de Gabor orientées, qui sont une composition d'un noyau gaussien et sinusoïdal.

$$g(x,y|\theta, \psi, \lambda, \sigma, \gamma) = \exp\left(\frac{x'^2 + (\gamma y')^2}{-2\sigma^2}\right) \cos\left(\frac{2\pi}{\lambda} + \psi\right)$$

Avec :

$$x' = x \cos\theta + y \sin\theta$$

$$y' = -x \sin\theta + y \cos\theta$$

Pour la dernière caractéristique, c'est celle du couleur. Puisque les humains semblent être attirés par un certain ensemble de couleurs, c'est pourquoi un biais de couleur est mis en œuvre. Le filtre est utilisé pour extraire les canaux rouge, vert, jaune et bleu de l'image d'entrée. Donc, quatre pyramides gaussiennes R () ; G () ; B() et Y () sont créées à partir de ces canaux de couleur.

Dans l'ensemble, le modèle calcul 42 cartes de caractéristiques, six pour l'intensité, 12 pour la couleur et 24 pour l'orientation.

Ensuite, il survient l'étape de la combinaison de différentes cartes de caractéristiques. Le problème qu'il peut arriver dans cette partie, est que les objets saillants apparaissant fortement dans quelques cartes et donc ils peuvent être masqués par du bruit ou des objets moins saillants présents dans un plus grand nombre de cartes. Comme solution le modèle utilise un opérateur de normalisation des cartes, N (:). Son rôle est de normaliser les valeurs de la carte à un intervalle fixé, ensuite trouver l'emplacement du maximum global M de la carte et calculer la moyenne \overline{m} de toutes ses autres localisations. Après multiplier la carte par (M- \overline{m}).

Seuls les maxima locaux d'activité sont considérés de telle sorte que N (:) compare les réponses associées à des "points" d'activation significatifs sur la carte et ignore les zones homogènes. Pour la comparaison de l'activité maximale de la carte entière à la moyenne de tous les points d'activation, on mesure la différence entre l'emplacement le plus actif et la moyenne. Lorsque cette différence est importante, l'endroit le plus actif est le point qu'on cherche. Lorsque la différence est faible, la carte ne contient rien d'unique et est supprimée. [5]

5. Grad-Cam :

C'est une méthode basée sur la rétropropagation, elle utilise une architecture de réseau convolutif CNN. Ce genre de méthode est rapide en termes de calcul, et elle produit des cartes de saillance mettant en évidence les zones de l'image qui ont plus d'influence sur la prédiction. Ces cartes aident à visualiser si le modèle se concentre sur les mêmes régions sur lesquelles les humains se concentraient.

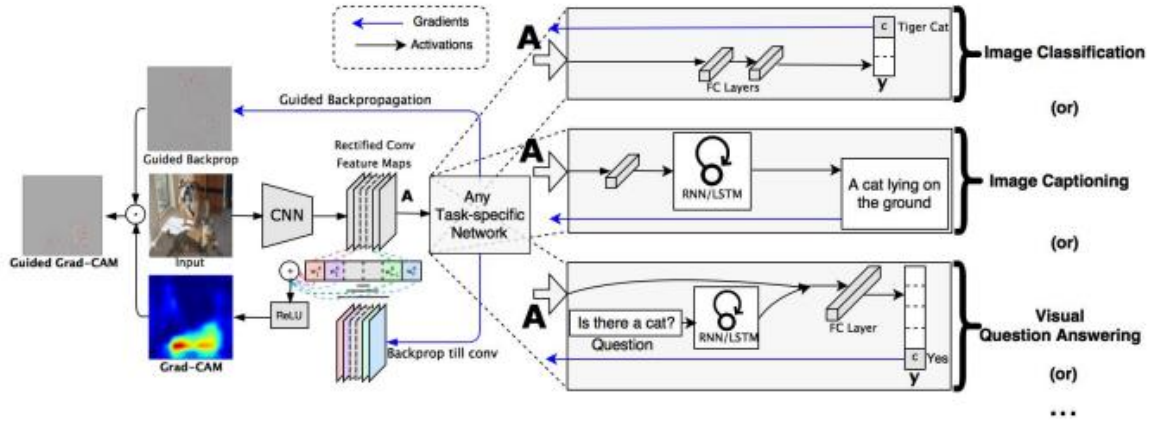


Fig. 4 Grad-CAM : Explications visuelles des réseaux profonds via la localisation basée sur le gradient.

Afin d'obtenir le Grad-CAM d'une image donnée à l'entrée, on calcule le score d'importance basé sur les gradients. Ce score est donné par la formule :

$$\alpha_k^c = \frac{1}{Z} \sum_i \sum_j \frac{\partial y^c}{\partial A_{ij}^k}$$

Où k est l'indice de la carte d'activation dans la dernière couche convolutive, y^c est le score de classe ciblée c , c'est-à-dire la valeur que nous obtenons dans la couche de sortie du réseau convolutif juste avant le softmax et A_{ij} est la carte d'entité.

On considère que les pixels ayant une forte influence positive sur le score sont les pixels correspondant aux régions d'intérêt, et pour les négatifs ou ceux proches de zéro, ils correspondent aux éléments non importants comme l'arrière-plan. Ensuite, nous multiplions chaque carte d'activation par son score d'importance alpha et additionnons les valeurs.

Pour l'obtention de ces régions importantes, une non-linéarité ReLU est appliquée à la sommation :

$$L_{Grad-CAM}^c = ReLU \left(\sum_k \alpha_k^c A^k \right)$$

La raison de l'utilisation d'une fonction d'activation RELU est d'éliminer les pixels qui ont une influence négative et de ne garder que les pixels ayant une densité augmentée. [6,7]

Les Saccades et les points de fixations :

Les saccades sont des mouvements balistiques rapides des yeux qui changent brusquement le point de fixation. Ils nous aident à comprendre pourquoi et comment il est possible de regarder dans n'importe quelle direction et d'orienter notre regard exactement vers l'endroit où un objet visuel est apparu. [8] L'idée, maintenant, est d'essayer de transmettre cette capacité de parcourir l'image avec des saccades à une machine, comment ça se passe-t-il donc ?

Pour détecter les points de fixation : un mécanisme d'attention séquentielle est démontré et est mis en œuvre, ainsi on calcule les points de fixations f en se basant sur la formule suivante :

$$f^{(t+1)} = \arg \max_{i,j} S^{(t)}$$

Avec

$$S^{(t+1)} = (1 - G_\sigma(\delta_{i,j}))$$

Et : $S(t)$ la carte de saillance actuelle, où $\delta_{i,j}$ est une image dont le pixel (i, j) est le seul pixel non nul, et G est le regard.

Pour la Fixation séquentielle utilisant l'attention séquentielle. Partant de la carte de saillance, le point de la saillance maximale est échantillonné. Ensuite, une région autour de ce point est atténuée avant le point suivant est échantillonné.

6. Métriques d'évaluation

Pour évaluer les performances du modèle, nous utilisons des métriques d'évaluation. Il s'agit des scores permettant d'évaluer les performances des modèles de saillance et de les comparer les uns aux autres. Ces métriques évaluent une carte de saillance en leur donnant, en entrée, la carte de saillance et la densité de fixation pour calculer la probabilité de distribution. Parmi les métriques les plus utilisées, on trouve AUC, le gain d'information, NSS et SIM.

○ La métrique AUC

En générale, AUC est la mesure de séparabilité. Et dans le cas de saillance, pour un seuil donné, les pixels de saillance peuvent être traités comme un classificateur, avec tous les points au-dessus du seuil sont indiqués comme "fixation" et tous les points en dessous du seuil comme "arrière-plan".

La métrique AUC ajoute un bruit aléatoire sur la carte de saillance pour s'assurer qu'il n'y avait pas de pixels avec la même valeur de saillance. Ce type de métrique mesure la performance en choisissant la distribution qui la plus grande valeur. [9]

○ La métrique de similarité SIM

Cette métrique normalise la carte de saillance du modèle et la carte empirique comme des vecteurs de probabilité et il additionne le minimum par pixel de deux cartes de saillance. Cette méthode mesure la distance entre les cartes de saillance. [10]

$$\sum_i \min(p_i, q_i) = \sum_i \frac{1}{2} (p_i + q_i - |p_i - q_i|) = 1 - \frac{1}{2} \|p - q\|_1$$

○ Gain d'information

Le critère d'évaluation Gain d'information se base sur la probabilité, ce critère considère le placement de la fixation comme une probabilité. Ce qui est évident, puisque les êtres humains placent leurs fixations à des endroits différents, les uns par rapport aux autres. Le gain d'information est considéré comme un critère d'évaluation représente une fonction de perte utile et intuitive pour l'évaluation du modèle. En utilisant le cadre de log-vraisemblance rapporté spécifiquement comme gain d'information expliqué, il indique ce que le modèle sait des données au-delà du modèle de base, qui est ici le biais central indépendant de l'image, exprimé en bits/fixation :

$$IG(\hat{p} \| p_{baseline}) = \frac{1}{N} \sum_i \log \hat{p}(x_i, y_i | I_i) - \log p_{baseline}(x_i, y_i)$$

Où $p_{\mathcal{X}}(x, y|I)$ est la densité du modèle à l'emplacement (x, y) lors de la visualisation de l'image I. et ***Pbaseline*** est la densité du modèle de base.

- NSS (normalized scanpath saliency)

Etant donné une carte binaire des emplacements de fixations F (données humaines) et la carte de saillance S (réponse du modèle) on calcule le NSS comme le suivant :

$$NSS(S, F) = \frac{1}{N} \sum_{i=1}^N \bar{S}_i \odot F_i$$

Avec $N = \sum F_i$ le nombre total de pixels fixés et $\bar{S} = \frac{S - \mu(S)}{\sigma(S)}$ est la carte de saillance normalisée. Les valeurs positives montrent la correspondance et les valeurs négatives l'anti-correspondance. [\[11\]](#)

IV. Étude de cas :

Dans notre étude, nous nous intéressons, principalement, par le modèle de Deep gaze étant donné qu'il s'agit d'une modèle tendance utilisée pour la prédiction de la saillance, vu son efficacité, et de fait qu'il donne un bond significatif dans les performances. Nous avons fixé un objectif de montrer d'abord la performance de ce modèle par rapport à d'autres modèles tels que (ITTYKOCH, ICF, Grad-Cam...), en les comparant par des métriques tels que : NSS, AUC, Ensuite, nous essayons d'extraire, des cartes de saillance obtenu de chaque modèles, des points de fixations, et modélisons, après, le fonctionnement de l'œil humain, autrement dit, parcourir les images de manière séquentielle à l'aide des saccades oculaires.

1. Implémentation du DeepGaze :

La mise en place de modèle deep gaze n'était pas évident. En effet, nous avons tester différents codes de ce modèles conçus par différentes équipes, et finalement nous avons réussi à trouver un qui marche bien.

Tout d'abord, nous avons essayé de mettre en œuvre un code élaboré par matthias-k [<https://github.com/matthias-k/DeepGaze>], qui a été développé à l'aide des fonctions de bibliothèque pytorch. Le premier problème que nous avons rencontré est lie à la performance de la machine que nous utilisons. En effet, le code ne fonctionne que sur des ordinateurs puissants avec une carte graphique de type GPU. Nous avons résolu ce problème, en travaillant sur le google colab avec la modification de type d'exécution. Donc, nous avons réglé ce problème lié à la carte graphique.

Ensuite, une erreur s'est aussi intervient lors de l'exécution du code, qui est lie à la version de pythorch utilisée, à découvrir finalement que c'est une erreur associée à la conception du code, en regardant un forum ou le développeur de code s'excuse et il promet de fixer le problème bientôt.

Après, nous avons trouvé un autre programme développé par Steffen Schneider [[12](#)]. Nous avons essayé de le faire fonctionner mais nous avons été contraintes par une data que nous n'avons pas trouvé et qu'est été utilisés pour tester le modèle. Nous avons essayé de la substituer par une autre data mais ce n'était pas marché dû au format de données compliqué, et de fait que nous n'avons pas trouvé aucunes autres informations utiles sur la data qui peut faire marcher le programme. Nous avons décidé d'aller chercher un autre programme.

Finalement, nous sommes arrivés à faire fonctionner le modèle proposé par Matthias Kümmerer [[13](#)]. D'abord, le modèle nous ne donne pas de bons résultats que sur une image de bibliothèque Scipy.

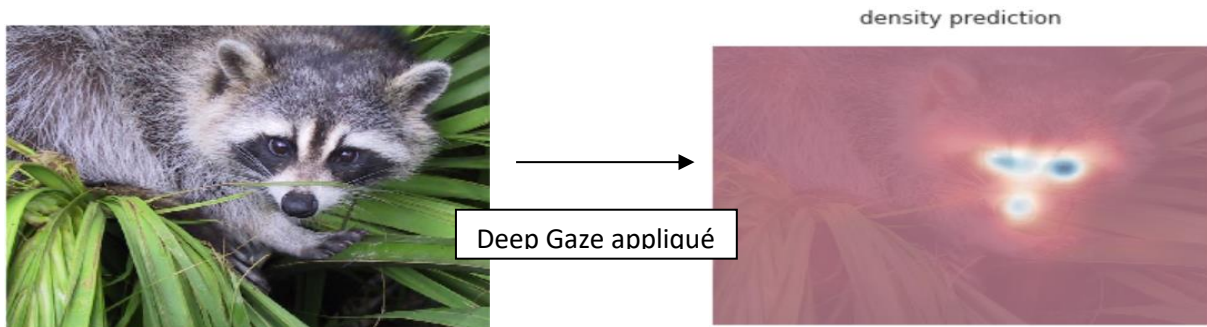


Fig. 5 résultat d'application de deepgaze sur l'image face de bibliothèque scipy

Nous avons testé le modèle sur d'autres images, mais le modèle ne marche pas



Fig. 6 : deepgaze donne mauvais résultats

A découvrir par la suite que plus qu'on recompile le modèle sur la même image, plus que la performance s'améliore :

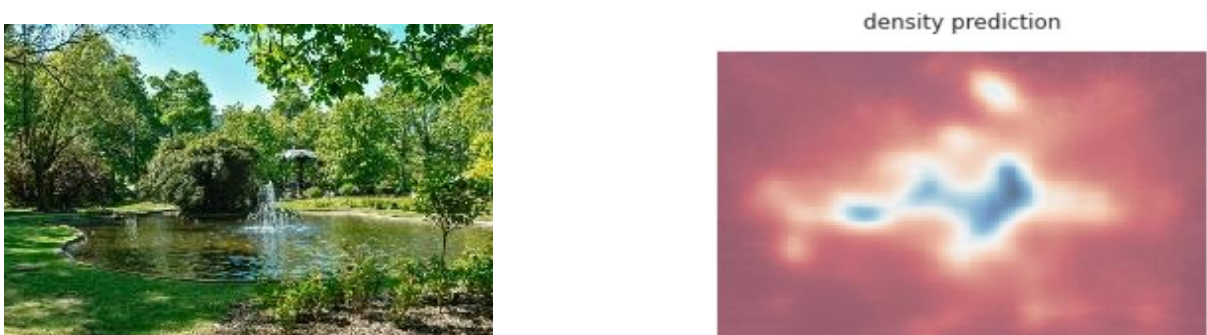


Fig. 7 : deepgaze donne de bons résultats

Donc, nous compilons le modèle sur même image tant de fois jusqu'à ce qu'il nous donne des meilleurs résultats.

2. La Fixation séquentielle

Après avoir obtenu les cartes de saillance de chaque modèle, nous essayons de tracer la trajectoire des saccades sur l'image en se basant sur les points de fixations extraites des cartes de saillance selon ce que nous avons expliqué dans la section 2.

Nous avons utilisé, pour ce faire, un algorithme (en annexe). Au début, il nous donne des mauvais résultats pour les cartes de saillances obtenues par des modèles autres que le Ittykoch :

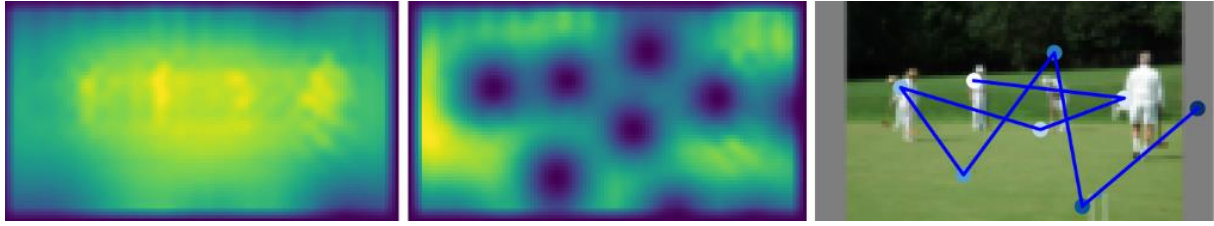


Fig. 8 La Fixation séquentielle obtenu par le ITTYCOCH

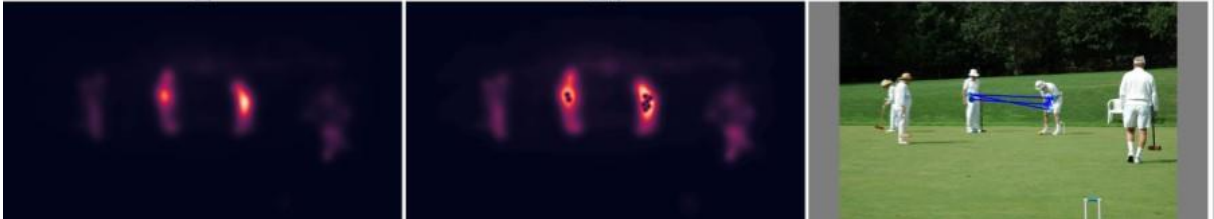


Fig. 8 La Fixation séquentielle obtenu par le Deepgaze

Finalement, nous avons fixé le problème en madéfiant les paramètres de l'algorithme pour le rendre adapté à chaque modèle et pour chaque image. En effet, pour chaque modèle nous spécifions le nombre de points à tracer, le nombre d'arrêt ainsi que la taille de l'image à la sortie du modèle.

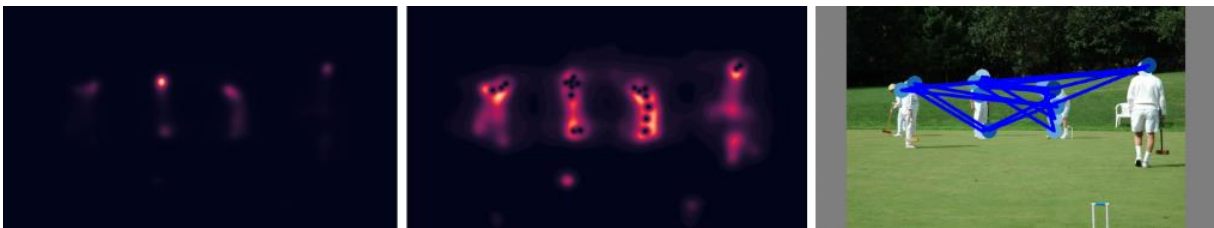


Fig. 9 La Fixation séquentielle obtenu par le Deepgaze après avoir madéfier les paramètres

3. Comparaison et évaluation

Le jeu de données utilisé pour comparer nos modèles est CAT2000, il s'agit d'une DATA contient deux ensembles d'images : training et test. Les images des trains (100 de chaque catégorie) contiennent les fixations de 18 observateurs avec seulement 6 observateurs sont retenus. Nous avons utilisé que les images de training pour avoir les grounds Truth des images, autrement dit les fixations réelles obtenu en montrant les images à des personnes de tranches d'âges entre 18 ans et 27 ans.

○ Comparaison des modèles visuellement


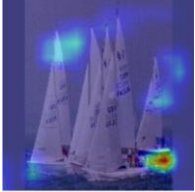
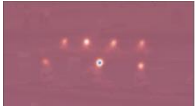
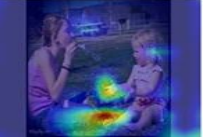
Image	ICF	DeepGazell	IttyKoch	Grad-Cam
				
				
				
				
				
				

Tableau 1: comparaison entre IttyKoch, deep gaze, ICF et Grad-Cam pour obtenir les cartes de saillances sur le jeu de données CAT2000.

- Comparaison des modèles avec des métriques d'évaluation

Nous avons calculé la moyenne des scores NNS, AUC, gain d'informations et la similarité de 50 images de cette Data pour chaque modèle et nous avons trouvé les résultats suivants :

MODELE	AUC_judd_score	Infogain_score	NSS_score	SIM_score
ittykoch	0,27	-1.33	0.86	0,64
ICF	0,41	-0.96	1.59	0,61
deepgaze	0,48	-0.85	1.97	0,67

Tableau2 Résultats détaillés des métriques. Infogain = gain d'information (bits / fixation), AUC_judd_score= aire sous la courbe ROC (%), NSS = saillance normalisée du chemin de balayage. DeepGaze II atteint des performances optimales en NSS, AUC, gain d'informations et la similarité.

- La comparaison des fixations séquentielles des modèles :

ITTYKOCH

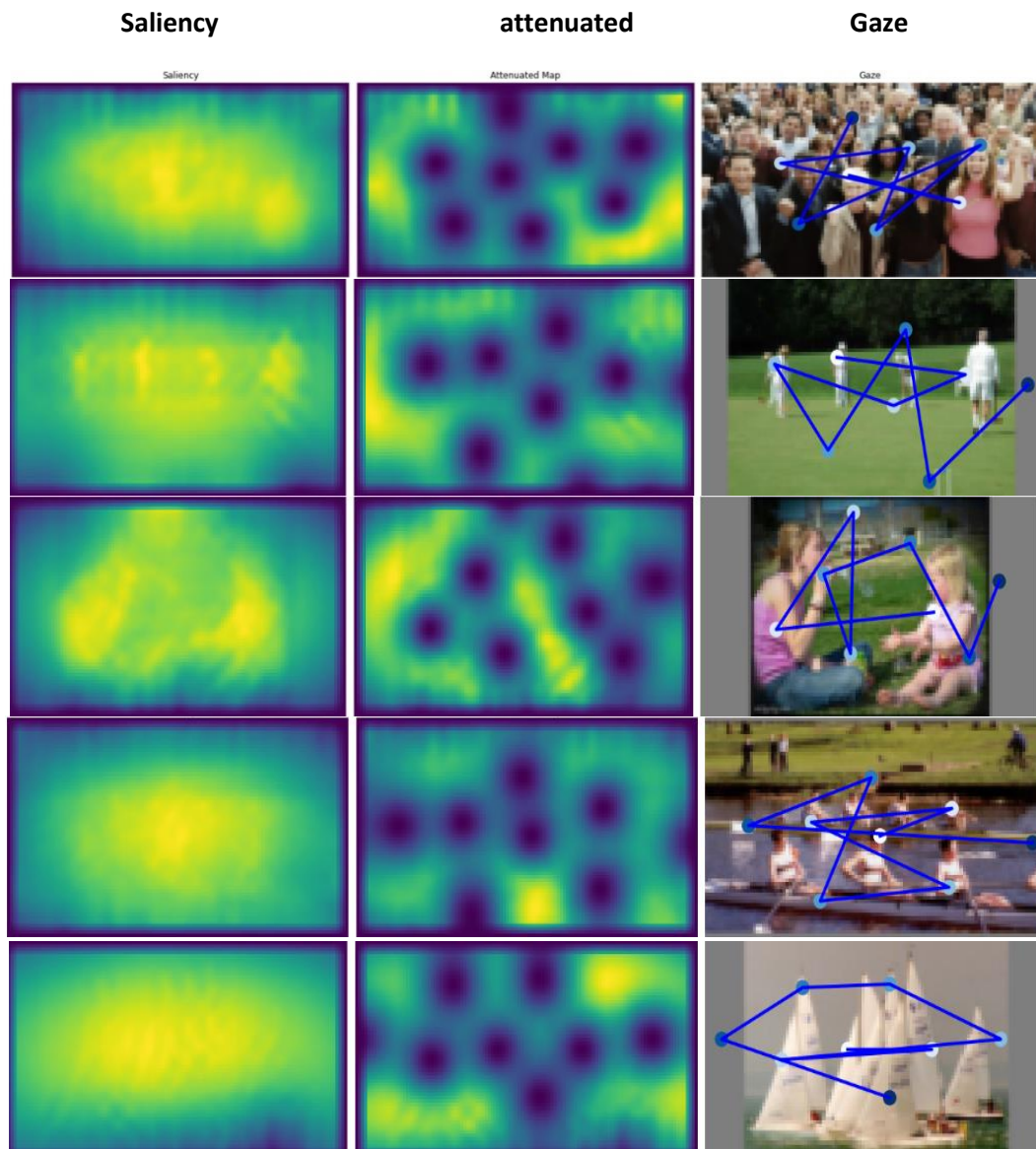


Fig.10 les résultats des fixations séquentielles de model IttyKoch pour toutes les images, y compris les stimuli de test

ICF :

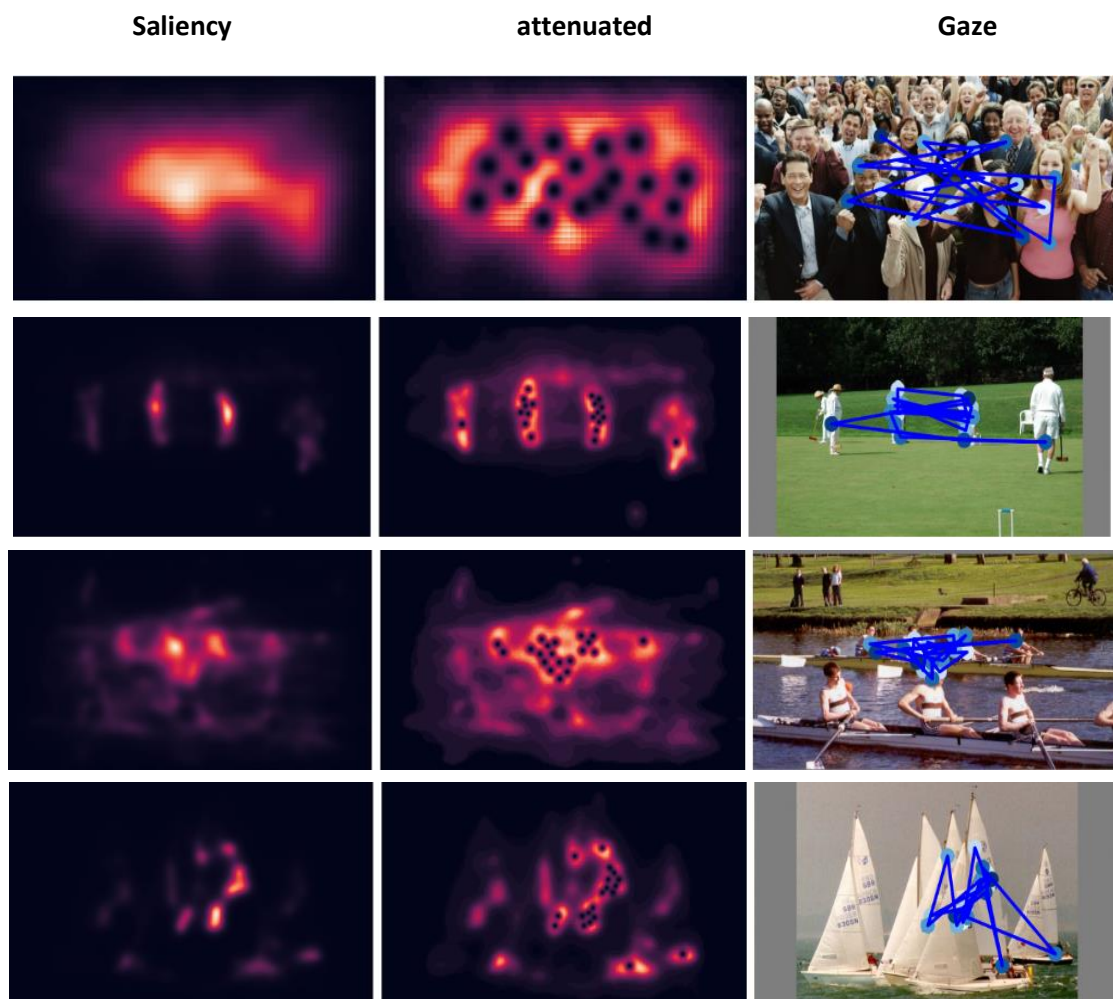


Fig.11 : les résultats des fixations séquentielles de model ICF pour toutes les images, y compris les stimuli de test.

Deepgaze :

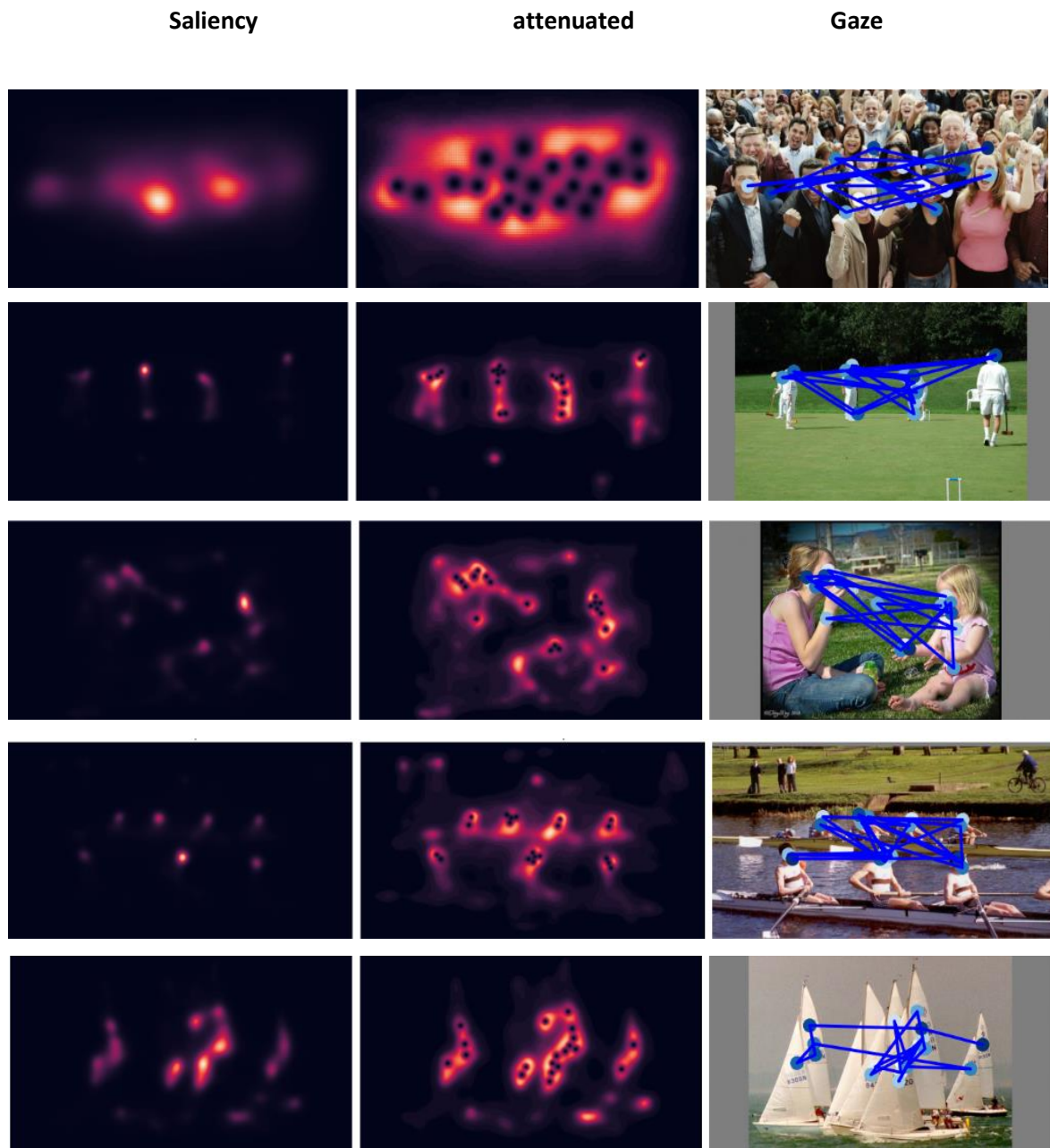


Fig.12 : les résultats des fixations séquentielles de model deepgaze pour toutes les images, y compris les stimuli de test.

Après avoir réalisé une analyse des résultats des métriques (NSS, AUC, gain d'information, SIM) et de fixation séquentielle, nous avons constaté que le meilleur modèle de la prédiction de saillance est celui de DeepGaze.

V. Conclusion

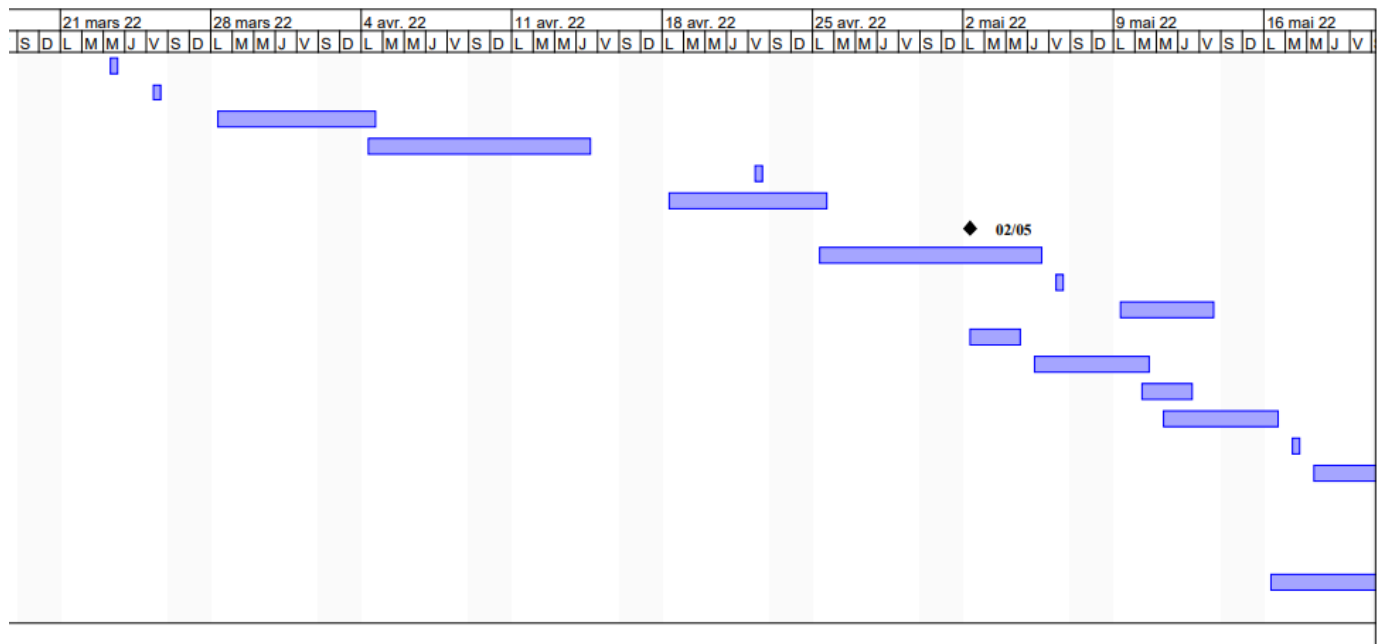
Au cours de ce projet, nous avons simulé les saccades par l'utilisation des modèles de réseau de neurones notamment DeepGaze II, ICF et Grad-CAM, ainsi que des modèles non neuronaux tel que ItyKorch. Nous avons également évalué ces modèles à travers des métriques. Après l'évaluation de ces modèles nous avons constaté que DeepGazeII est le plus performant. Pour conclure, malgré toutes les difficultés rencontrer durant ce projet, nous avons bien saisi l'objectif de ce projet théoriquement et techniquement.

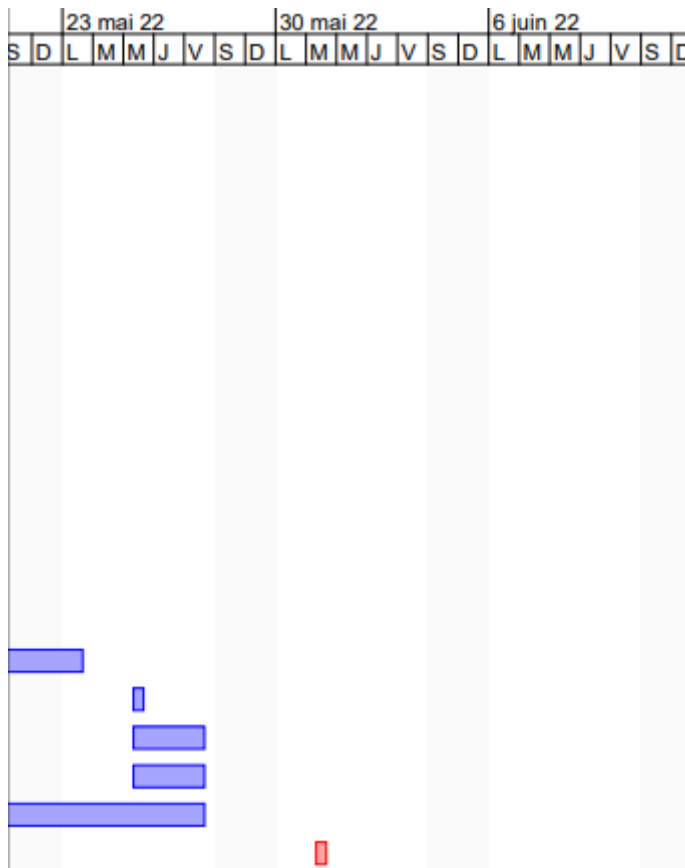
Bibliographie :

- [1] Sina Mohammadi, Mehrdad Nooria, Ali Bahria , Sina Ghofrani Majelana , Mohammad Havaeib. CAGNet: Content-Aware Guidance for Salient Object Detection. arxiv.org/pdf/1911.13168v2.pdf.
- [2] Runmin Cong, Jianjun Lei, Senior Member, IEEE, Huazhu Fu, Senior Member, IEEE, Ming-Ming Cheng, Weisi Lin, Fellow, IEEE, and Qingming Huang, Fellow, IEEE : Review of Visual Saliency Detection with Comprehensive Information. arxiv.org/pdf/1803.03391.pdf.
- [3] Ali Borji, Member, IEEE : Saliency Prediction in the Deep Learning Era: Successes, Limitations, and Future Challenges. arxiv.org/pdf/1810.03716.pdf.
- [4] Matthias Kummerer, Thomas S.A. Wallis, Leon A. Gatys, Matthias Bethge : Understanding Low- and High-Level Contributions to Fixation Prediction.
openaccess.thecvf.com/content_ICCV_2017/papers/Kummerer_Understanding_Low- and_ICCV_2017_paper.pdf.
- [5] Christof Koch, Ernst Niebur : A Model of Saliency-based Visual Attention for Rapid Scene Analysis. www.researchgate.net/publication/3192913_A_Model_of_Saliencybased_Visual_Attention_for_Rapid_Scene_Analysis.
- [6] Ramprasaath R. Selvaraju, Michael Cogswell, Abhishek Das, Ramakrishna Vedantam, Devi Parikh, Dhruv Batra : Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization. [\[1610.02391v4\] Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization \(arxiv.org\)](https://arxiv.org/abs/1610.02391v4).
- [7] Math and AI : Visual Saliency: From simple gradient based approach to GradCAM. [\(69\) Visual Saliency: From simple gradient based approach to GradCAM - YouTube](https://www.youtube.com/watch?v=69).
- [8] L. Goffart : Saccadic Eye Movements(2009). <https://www.sciencedirect.com/topics/medicine-and-dentistry/saccadic-eye-movement>.
- [9] Nicolas Riche, Matthieu Duvinage, Matei Mancas, Bernard Gosselin, Thierry Dutoit : Saliency and Human Fixations : State-of-the-art and Study of Comparison Metrics.
https://openaccess.thecvf.com/content_iccv_2013/papers/Riche_Saliency_and_Human_2013_ICCV_paper.pdf.
- [10] Matthias Kummerer, Thomas S.A. Wallis, Matthias Bethge : Saliency Benchmarking Made Easy: Separating Models, Maps and Metrics. [Saliency Benchmarking Made Easy: Separating Models, Maps and Metrics \(thecvf.com\)](https://arxiv.org/abs/1610.02391v4).
- [11] Steffen Schneider: Visual Attention: From Classical to Modern Approaches. <https://github.com/stes/saliency/blob/master/report/main.pdf>.
- [12] Steffen Schneider: Implementing Visual Saliency Models. <https://github.com/stes/saliency>.
- [13] Matthias K  mmerer : DeepGaze II. <https://deepgaze.bethgelab.org/>.

Annexe1 : Gantt

	🕒	Nom	Durée	Début	Fin	Prédécesseurs	Noms des ressources	
1	🕒	Choisir le sujet	1 jour?	23/03/22 08:00	23/03/22 17:00			
2	🕒	1ere réunion avec le tuteur	1 jour?	25/03/22 08:00	25/03/22 17:00			
3	🕒	Rechercher sur le modèle ...	6 jours?	28/03/22 07:00	04/04/22 17:00			
4	🕒	Analyser les articles et le ...	9 jours?	04/04/22 07:00	14/04/22 17:00			
5	🕒	2e réunion avec le tuteur	1 jour?	22/04/22 07:00	22/04/22 17:00			
6	🕒	Déterminer le contexte, l'o...	6 jours?	18/04/22 07:00	25/04/22 17:00			
7	🕒	1e rendu	0 jours?	01/05/22 07:00	02/05/22 17:00			
8	🕒	Se focaliser sur DeepGaze ...	9 jours?	25/04/22 07:00	05/05/22 17:00			
9	🕒	3e reunion avec le tuteur	1 jour?	06/05/22 07:00	06/05/22 17:00			
10	🕒	Reussir à faire fonctionner...	5 jours?	09/05/22 07:00	13/05/22 17:00			
11	🕒	Chercher d'autre méthode...	3 jours?	02/05/22 07:00	04/05/22 17:00			
12	🕒	Travailler sur les 3 autres ...	4 jours?	05/05/22 07:00	10/05/22 17:00			
13	🕒	Chercher les données pou...	3 jours?	10/05/22 07:00	12/05/22 17:00			
14	🕒	Comparer la performance ...	4 jours?	11/05/22 07:00	16/05/22 17:00			
15	🕒	4e réunion avec le tuteur	1 jour?	17/05/22 07:00	17/05/22 17:00			
16	🕒	Rechercher sur les saccades	4 jours?	18/05/22 07:00	23/05/22 17:00			
17	🕒	5e reunion avec le tuteur	1 jour?	25/05/22 07:00	25/05/22 17:00			
18	🕒	Analyser la performance d...	3 jours?	25/05/22 07:00	27/05/22 17:00			
19	🕒	Comparer les modèles en s...	3 jours?	25/05/22 07:00	27/05/22 17:00			
20	🕒	Rédaction du CR finale	10 jours?	16/05/22 07:00	27/05/22 17:00			
21	🕒	Soutenance Finale	1 jour?	31/05/22 07:00	31/05/22 17:00			





Annexe2 : liens des codes utilisées

Lien Google colab de Deepgaze II et ICF :

<https://colab.research.google.com/drive/17Fgl7zAuDN40ARaNQ1M8zBPggjbi7UQT?usp=sharing>

Lien Google colab de ItyKoch :

<https://colab.research.google.com/drive/1F0iSbWn7ST62QYUYfdQtGEI6WTslSDBz?usp=sharing>

Lien Google colab de Grad_Cam:

<https://colab.research.google.com/drive/1pOc1j3ASuDaKX4dde6DsIYj69YNYoLjR?usp=sharing>

lien du jeu de données CAT2000 :

http://saliency.mit.edu/results_cat2000.htm

lien Google colab pour évaluer les modèles :

https://colab.research.google.com/drive/1Ap_4ptK_YZnhOwj-iB-XzyoIM6YnxStB?usp=sharing