



# IClicker

PROJET NUMÉRO 73

---

Étudiant : Sergiu Iacob  
Encadrant : Marius Bilasco

mars, 2020

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Le contexte . . . . .	2
1.2	L'idée . . . . .	2
1.3	Motivation . . . . .	2
1.4	Le but . . . . .	3
1.4.1	Objectifs essentiels . . . . .	3
1.4.2	Objectifs préférables . . . . .	4
<b>2</b>	<b>Plan de développement</b>	<b>5</b>
2.1	Comment résoudre le problème . . . . .	5
2.1.1	Définir le problème . . . . .	5
2.1.2	Le neurone artificiel . . . . .	5
2.1.3	Les réseaux de neurones . . . . .	6
2.2	Strategie . . . . .	6
2.2.1	Obtenir des données . . . . .	6
2.2.2	Perceptrons multicouches . . . . .	7
2.2.3	Réseau neurones convolutifs . . . . .	7
2.2.4	Recherche d'autres méthodes . . . . .	7
<b>3</b>	<b>Informations préliminaires</b>	<b>8</b>
3.1	Manière de travailler . . . . .	8
3.2	Configuration utilisée . . . . .	8
3.3	Informations techniques . . . . .	8
3.4	Conditions requises . . . . .	8
3.5	Limites . . . . .	9

# CHAPITRE 1

## INTRODUCTION

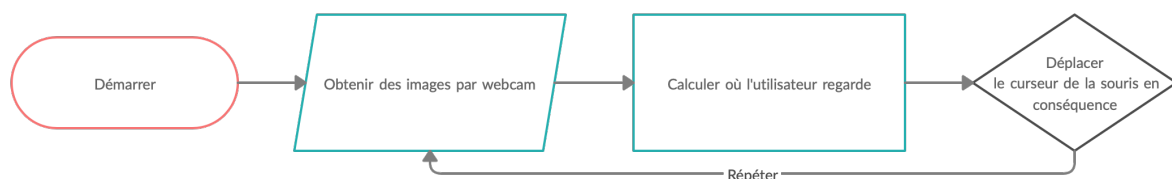
### 1.1 LE CONTEXTE

Au moment de travailler sur ce projet, j’ai passé 5 semestres comme étudiant à la Faculté d’Informatique, Iași, et le dernier (en cours) comme étudiant Erasmus, à l’Université de Lille. Comme le destin l’a voulu, l’épidémie de COVID-19 nous force actuellement à nous isoler pendant au moins deux semaines, alors quelle meilleure occasion de passer du temps de qualité à travailler sur ce projet ?

Il s’agit d’un projet à double objectif. En tant qu’étudiant Erasmus, j’aurai développé ce projet pour le cours “Projet Individuel”, ayant un soutenance finale à la fin du mois de Mai, 2020. Après avoir terminé le programme Erasmus, le projet aura servi de thèse pour l’obtention d’une Licence, à la Faculté d’Informatique, Iași.

### 1.2 L’IDÉE

L’idée de base est d’avoir une logiciel qui suivra le visage de l’utilisateur, avec le but de déplacer la position du curseur en temps réel. Pour cela, nous utiliserons une webcam pour obtenir des images. Avec ceux, nous essaierons de calculer l’endroit où l’utilisateur regarde et de déplacer le curseur de la souris en conséquence.



**FIGURE 1.1**  
Procédure générale

### 1.3 MOTIVATION

Pour décider le projet sur lequel je voulais travailler, j’ai pris en compte deux facteurs clés : ma future carrière professionnelle et la facilité d’utilisation du projet. Je voulais travailler sur un projet qui alimenterait mon intérêt pour l’Intelligence Artificielle et qui me donnerait la possibilité d’appliquer les recherches que je ferai dans ce domaine. Aussi, je voulais avoir une approche pratique de ce projet et faire quelque chose qui soit utile aux gens.

Si l'on regarde en arrière, on constate que les gens ont toujours travaillé à rendre la technologie accessible au plus grand nombre. Qu'il s'agisse du système braille, de l'aide aux aveugles pour lire et écrire, des lecteurs d'écran ou des assistants intelligents (Siri, Bixby, etc.), l'accessibilité est un facteur clé pour faire en sorte que les gens bénéficient des avantages de la technologie. Par conséquent, je serais rassuré de savoir que j'ai moi aussi travaillé sur un projet qui pourrait potentiellement aider les personnes handicapées – et pas seulement elles – à utiliser l'ordinateur sans l'aide des mains.

Quant à l'Intelligence Artificielle, il est inutile de souligner son importance contemporaine. De l'applicabilité médicale, de la conduite et du pilotage autonomes et l'agriculture intelligente aux réfrigérateurs intelligents qui vous disent quand vous êtes à court de lait, l'Intelligence Artificielle est très répandue et sa croissance ne va pas s'arrêter de sitôt. Pour moi, c'est une raison supplémentaire de l'étudier et de mieux la comprendre, car elle est intégrée dans notre vie quotidienne.

## 1.4 LE BUT

### 1.4.1 OBJECTIFS ESSENTIELS

L'un des premiers objectifs que j'essaierai d'atteindre est de savoir approximativement où l'utilisateur regarde en utilisant uniquement ses yeux. Par exemple, un bon début serait de savoir sur quelle partie de l'écran l'utilisateur se concentre. En se basant sur la grille ci-dessous, si l'utilisateur regarde le carré numéro *i*, alors nous déplacerons le curseur de la souris au centre de ce carré.



**FIGURE 1.2**  
3x3 grille

Deuxièmement, je vais essayer de simuler les fonctions de clic comme suit : si l'utilisateur ferme son œil gauche pendant un certain temps, nous interpréterons cela comme un clic gauche. La même chose sera faite pour le clic droit, en utilisant l'œil droit.

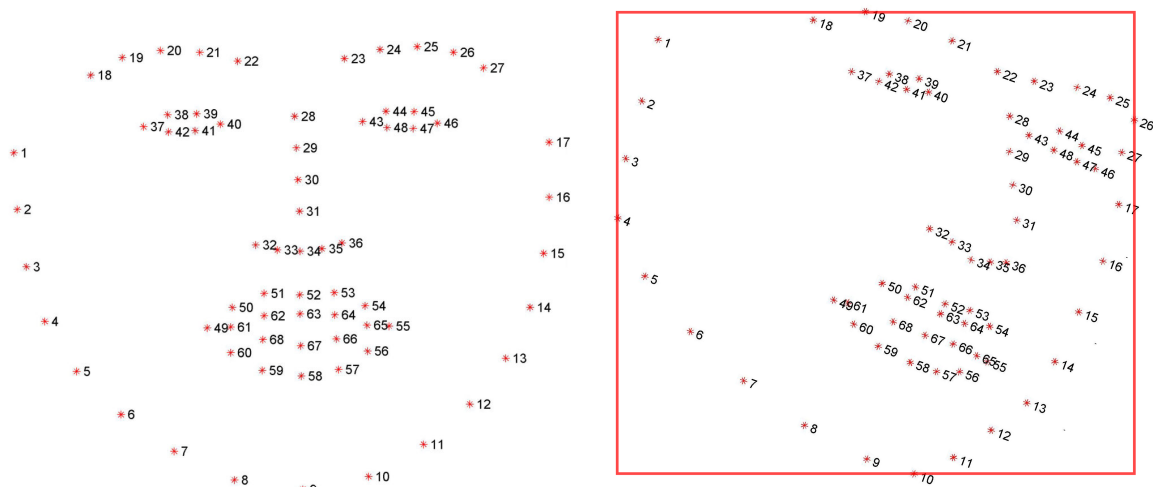
A titre d'orientation générale, je vais essayer de développer l'application afin qu'elle soit facilement maintenable. Je me concentrerai également sur les concepts que j'ai appris tout au long de mes études

universitaires : “clean code”, principes OOP, principes SOLID, cohésion, etc. Une bonne condition serait également de rendre l’application multiplateforme.

### 1.4.2 OBJECTIFS PRÉFÉRABLES

Pour aller un peu plus loin, si la prédiction des carrés se passe bien, on peut essayer de prédire l’emplacement exact du curseur. Cela signifie que nous déplacerons le curseur de la souris exactement là où nous pensons que l’utilisateur regardera.

Avoir plus d’informations sur l’utilisateur pourrait nous aider dans nos calculs. Pour cela, je vais aussi essayer de voir comment je peux utiliser d’autres parties du visage pour améliorer les calculs : par exemple, la position du nez. L’image ci-dessous représente quelques points de repère du visage que nous utiliserons pour nos calculs : les points [37, 48] sont pour les yeux et [28, 36] pour le nez.



**FIGURE 1.3**  
Repères faciaux  
Source : ibug

## CHAPITRE 2

# PLAN DE DÉVELOPPEMENT

### 2.1 COMMENT RÉSOUDRE LE PROBLÈME

#### 2.1.1 DÉFINIR LE PROBLÈME

Le problème que nous essayons de résoudre est de suivre précisément les yeux pour pouvoir déplacer le curseur de la souris. Cela fait partie d'une gamme plus large de problèmes de *Computer Vision*, et plus précisément de *eye tracking*. Selon WIKIPEDIA 2020, le suivi des yeux est "le processus de mesure du point de vue (où l'on regarde) ou du mouvement d'un œil par rapport à la tête".

Sachant cela, nous allons nous attaquer à ce problème en utilisant *l'Intelligence Artificielle*, car c'est devenu aujourd'hui un moyen viable de résoudre ces problèmes. Nous allons donc traiter un problème d'*Apprentissage Automatique*, plus précisément un problème d'*Apprentissage profond*.

Les tâches d'Apprentissage Profond peuvent être supervisées, semi-supervisées ou non supervisées. Nous nous concentrerons sur les tâches *supervisées* et nous traiterons notre problème comme l'une d'entre elles. Cela signifie que nous allons d'abord acquérir quelques données d'apprentissage, qui consisteront en des images de webcam marquées par la position du curseur de la souris. De cette façon, notre algorithme apprendra où le curseur de la souris doit se trouver, en fonction de l'endroit où l'utilisateur regardait lorsque l'image a été prise.

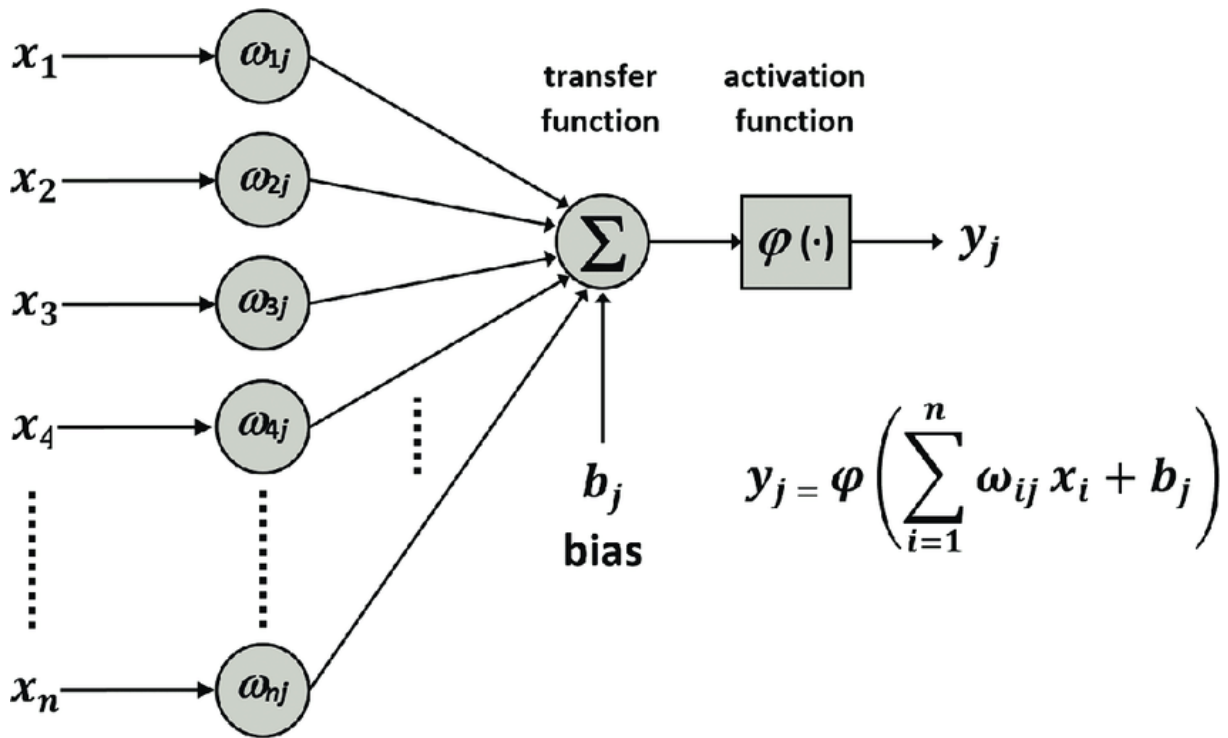
#### 2.1.2 LE NEURONE ARTIFICIEL

Avant d'entrer dans des architectures d'Apprentissage Profond plus sophistiquées, nous devons d'abord examiner le neurone artificiel. En bref, il s'agit d'un modèle formel et simplifié de neurone biologique. Il peut nous aider à faire la *classification binaire* à l'aide de la formule suivante :

$$y = \varphi(w * x + b)$$

Dans la formule ci-dessus,  $x$  est un vecteur de valeurs réelles, représentant notre entrée,  $w$  est un vecteur de valeurs réelles, appelé *poids* et  $b$  est le *biais*. Le produit  $w * x$  représente le produit scalaire et est égal à  $w * x = \sum_{i=1}^n w_i x_i$ ,  $n$  étant la longueur de notre entrée.

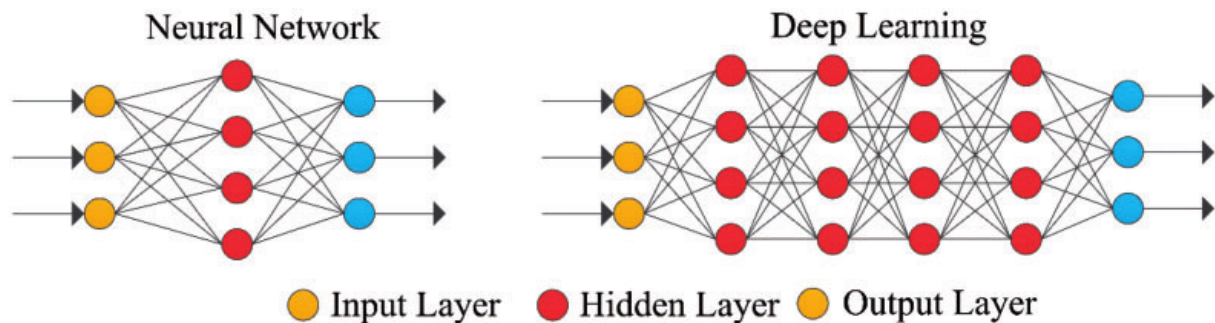
Le résultat de la classification est donné par  $\varphi$ , qui est appelé une *fonction d'activation*. Si cette fonction sert de seuil, elle effectuera une classification binaire, donnant soit 0 soit 1. Nous pourrions même utiliser une fonction d'activation telle que  $\varphi(x) = x$  qui ne nous donnera pas une classification binaire, mais pourrait nous aider à résoudre des problèmes de régression linéaire.



**FIGURE 2.1**  
Schéma d'un neurone artificiel  
Source : ResearchGate

### 2.1.3 LES RÉSEAUX DE NEURONES

Le “cheval de bataille” des problèmes de vision par ordinateur est le réseau de neurones artificiels. Celui-ci est, comme son nom l’indique, composé de plusieurs neurones artificiels, répartis sur plusieurs couches.



**FIGURE 2.2**  
Schéma d'un neurone artificiel  
Source : ResearchGate

## 2.2 STRATEGIE

### 2.2.1 OBTENIR DES DONNÉES

Il est bien connu qu’un algorithme d’Apprentissage Automatique n’est bon que si les données que nous lui fournissons le sont aussi. C’est incroyablement important, c’est pourquoi je vais m’efforcer de rassembler autant de données que possible, ainsi que d’autres informations que je pourrais trouver importantes.

Un autre point essentiel est *traitement des données* et *analyse des données*. Le premier concerne le nettoyage de l'ensemble des données des instances inutiles et la mise en forme des données recueillies dans un format utilisable pour nos réseaux neuronaux. Nous examinerons également comment nous pouvons obtenir des informations supplémentaires qui pourraient être utiles à partir de nos images.

### **2.2.2 PERCEPTRONS MULTICOUCHES**

Un type particulier de réseaux neuronaux est le réseau multicouche Perceptron. Ce sera ma première expérience et le point de départ pour tenter de résoudre le problème.

En utilisant ceci, je vais essayer de résoudre le premier objectif 1.4.1, et c'est approximativement de prédire où l'utilisateur regarde. Je vais d'abord essayer avec une grille plus petite de 2x2, puis je passerai à une grille de 3x3.

### **2.2.3 RÉSEAU NEURONES CONVOLUTIFS**

Une étape supplémentaire consistera à utiliser les réseaux neuronaux convolutifs, une architecture de pointe pour travailler avec des images. Cela pourrait permettre d'atteindre un objectif plus préférable, à savoir prévoir exactement l'emplacement du curseur de la souris en fonction de l'endroit où l'utilisateur regarde.

### **2.2.4 RECHERCHE D'AUTRES MÉTHODES**

Après avoir expérimenté les méthodes ci-dessus, je vais rechercher de meilleures façons de résoudre ce problème. Cela pourrait signifier un meilleur traitement des données, un meilleur réglage des réseaux de neurones, etc.



## CHAPITRE 3

# INFORMATIONS PRÉLIMINAIRES

### 3.1 MANIÈRE DE TRAVAILLER

Pour développer le projet, j'utiliserai tout d'abord GitHub pour structurer et sauvegarder le code. Vous pouvez trouver le dépôt du projet ici : <https://github.com/sergiuiacob1/iClicker/>

### 3.2 CONFIGURATION UTILISÉE

For any TODO : label this section and reference it wherever I mention “training took  $x$  seconds”. Mention laptop model.

Emphasize on webcam resolution & quality.

Emphasize on no GPU & cpu speed.

### 3.3 INFORMATIONS TECHNIQUES

TODO : Section title – maybe find a better one ?

Technology name	Version	Useful links
Python3	3.6	<a href="https://www.python.org">https://www.python.org</a>
Conda	4.8.0	<a href="https://conda.io/">https://conda.io/</a>
...	...	...

Maybe mention important Python libraries used :

Library name	Version	Useful links
Keras	...	...
PyTorch	...	...
OpenCV	...	...

### 3.4 CONDITIONS REQUISES

Technical requirements : Every technology previously mentioned 3.3 should be installed

Currently only tested on MacOS, possible that Windows will give errors. Linux should work

Link github repository here and mention install instructions.

### 3.5 LIMITES

Some of the *CURRENT* limits are (current because more will appear, probably) : App only tested with 1 monitor.

If setup has multiple cameras, it chooses the first one it finds.

Mention training data collection : only me. I wear no glasses.

Emphasize on wearing glasses : might not work. To test this.

Mention necessity of good lighting.

Might only work on MacOS and Linux right now (will test to be sure).

# BIBLIOGRAPHIE

WIKIPEDIA (2020). *Eye Tracking*. Wikimedia Foundation. URL : [https://en.wikipedia.org/wiki/Eye\\_tracking](https://en.wikipedia.org/wiki/Eye_tracking) (visité le 15 mar. 2020).