***RECONNAISSANCE FACIALE POUR GESTION DE PRESENCE***

# https://www.kongakura.fr/images/opencv-5de11b6a69d84.png***OpenCv***

*OpenCV (Open Computer Vision) est une bibliothèque graphique open source qui a été initialement développée par Intel. Elle est largement utilisée dans le domaine du traitement d'images et de la*

*vision par ordinateur.*

*Sa première version a été publiée en juin 2000 et elle continue d'être développée, maintenue et documentée par une communauté active de plus de 40 000 membres.*

*OpenCV propose une vaste gamme de fonctionnalités pour le traitement d'images en temps réel. Elle comprend plus de 2 500 algorithmes, une documentation complète, du code source et des exemples de code. Parmi les fonctionnalités offertes par OpenCV, on trouve la détection d'objets, la lecture et l'écriture d'images, la manipulation et la transformation d'images, la reconnaissance faciale, le suivi de mouvement, la segmentation d'image, et bien d'autres.*

*En ce qui concerne la reconnaissance faciale, OpenCV joue un rôle important. La bibliothèque offre des algorithmes et des modèles pré-entraînés spécialement conçus pour la reconnaissance faciale. Ces fonctionnalités permettent d'identifier et de reconnaître des visages dans des images ou des vidéos.*

* ***Pourquoi OpenCv ?***

*Bien que d'autres bibliothèques telles que scikit-learn, Keras et TensorFlow offrent également des fonctionnalités de reconnaissance faciale, OpenCV se démarque par ses performances optimisées et sa vaste gamme de fonctionnalités spécifiquement conçues pour le traitement d'images et la vision par ordinateur. Ces caractéristiques font d'OpenCV un choix solide et couramment utilisé pour les projets de gestion d'absence par reconnaissance faciale*

* *La Performance : OpenCV est une bibliothèque très efficace et optimisée pour le traitement d'images et de vidéos en temps réel. Il peut gérer de grands volumes de données d'images et de vidéos, ce qui est essentiel pour les systèmes de reconnaissance faciale.*
* *Les Fonctionnalités : OpenCV offre une large gamme de fonctionnalités de traitement d'images et de vidéos, notamment la détection de visage, la reconnaissance faciale, le suivi d'objets, la détection de mouvement, etc. Ces fonctionnalités sont toutes importantes pour la construction d'un système de gestion d'absence par reconnaissance faciale.*

***Installation d'OpenCV avec Python***

*Une fois Python installé, le gestionnaire de paquets pip est également installé automatiquement.*

*Pour installer le package OpenCV de Python, ouvrez votre terminal de commande et tapez la ligne suivante :*

*pip install opencv-python ou python -m pip install opencv-python*

*Pour installer la version comprenant les fonctionnalités supplémentaires (OpenCV Contrib), utilisez la commande suivante :*

*pip install opencv-contrib-python*

*Pour utiliser OpenCV dans votre script, importez-le en mémoire avec le mot-clé import :*

**

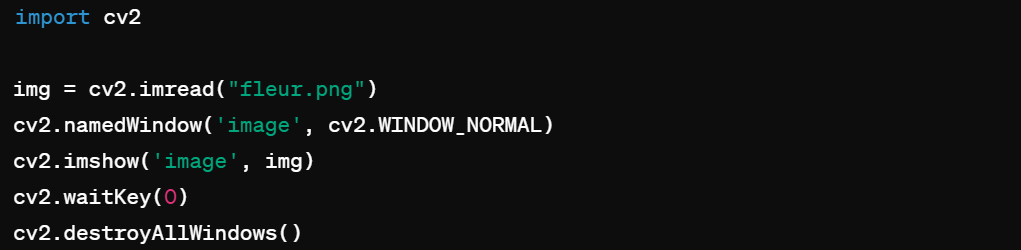
*Chargement et sauvegarde d'images avec OpenCV Pour charger une image, utilisez la fonction cv2.imread(nom\_image) :*

*Pour sauvegarder une image, utilisez la fonction cv2.imwrite(nom\_image, image) :*

*Affichage d'une image avec OpenCV Pour afficher une image à l'écran, commencez par créer une fenêtre graphique à l'aide de la fonction namedWindow, puis utilisez imshow pour afficher l'image.*

*La fonction waitKey permet d'attendre un événement, et destroyAllWindows libère la mémoire une fois la fenêtre fermée.*

*Voici un exemple complet :*

**

***Les fonctions de OpenCv***

***Lien vers une documentation de OpenCv :* https://docs.opencv.org/4.x/d6/d00/tutorial\_py\_root.html**

***Documentation des fonctions utilisées pour l'affichage d'image***

* *cv2.namedWindow('image', cv2.WINDOW\_NORMAL): Cette fonction crée une fenêtre avec le nom spécifié. Le deuxième argument indique si la fenêtre peut être redimensionnée (cv2.WINDOW\_NORMAL) ou non (cv2.WINDOW\_AUTOSIZE).*
* *cv2.imshow('image', img): Affiche l'image spécifiée dans la fenêtre nommée.*
* *cv2.waitKey(0): Attend une touche pressée pendant la durée spécifiée en millisecondes. Si 0 est passé en argument, il attend indéfiniment jusqu'à ce qu'une touche soit pressée.*
* *cv2.destroyAllWindows(): Détruit toutes les fenêtres créées par OpenCV et libère la mémoire associée.*

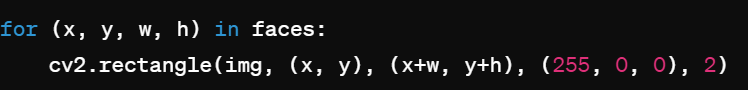
***Detection de visage avec OpenCv :***

*cv2.CascadeClassifier() Cette fonction initialise un détecteur de visage en utilisant un fichier XML pré-entrainé pour la détection de visage. Par exemple :*

**

*detectMultiScale() Cette fonction est utilisée pour détecter les visages dans une image en utilisant le détecteur de visage précédemment initialisé. Elle renvoie une liste de rectangles délimitant les visages détectés. Exemple :*

## *faces = face\_cascade.detectMultiScale(gray, scaleFactor=1.1, minNeighbors=5, minSize=(30, 30))*

* Dessin des rectangles autour des visages détectés Une fois les visages détectés, vous pouvez dessiner des rectangles autour d'eux en utilisant cv2.rectangle() :*

## ***Les fonctions de manipulation :***

1. *Cv2.cvtcolor() : Convertit une image d'un espace colorimétrique à un autre. Il prend trois arguments : l'image source, la conversion de l'espace colorimétrique et la destination de l'espace colorimétrique.*
2. *Cv2.resize() :   Redimensionne une image. Il prend trois arguments : l'image source, la largeur et la hauteur de destination.*
3. *Cv2.blur() : Applique un flou gaussien à une image. Il prend trois arguments : l'image source, le noyau de flou et la taille du noyau.*

***Les fonctions d’ecriture sur image :***

1. *Cv2.line() : Dessine une ligne sur une image. Il prend six arguments : l'image source, le point de départ, le point de fin, la couleur, l'épaisseur et le type de ligne.*
2. *Cv2.rectangle() : : Dessine un rectangle sur une image. Il prend six arguments : l'image source, le point supérieur gauche, le point inférieur droit, la couleur, l'épaisseur et le type de ligne.*
3. *Cv2.circle() : Dessine un cercle sur une image. Il prend cinq arguments : l'image source, le centre du cercle, le rayon, la couleur et l'épaisseur.*
4. *Cv2.putText() : Écrit du texte sur une image. Il prend six arguments : l'image source, le texte, la position, la police, l'échelle, la couleur et le type de police.*

***Les fonctions de capture de video :***

1. *Cv2.videoCapture() :  Crée un objet OpenCV pour capturer une vidéo à partir d'une source. Il prend un argument : l'index de la source vidéo (par exemple, 0 pour la webcam).*

## 

***Réalisé par : Sous l’encadrement de :***

*Mr.prof Baidada Chafik*

* *Miskar Amina*
* *Sabrou Hafssa*
* *Zahir Oumaima*