****

Bessekri mohamed 200147098

TALBI Mohamed abdelmadjid 201400007584

DEtection des objets

colores

Contents

[**Introduction** 3](#_Toc532935389)

[**Problématique :** 3](#_Toc532935390)

[**Chapitre I : Etat de l'art** 4](#_Toc532935391)

[1. OpenCv : 4](#_Toc532935392)

[2. Filtres morphologiques : 4](#_Toc532935393)

[2.1 Erosion : 4](#_Toc532935394)

[2.2 Délitation : 4](#_Toc532935395)

[3. Thresholding : 5](#_Toc532935396)

[**Chapitre I I : L'expérimentale** 6](#_Toc532935397)

[1. Méthodologie : 6](#_Toc532935398)

[2. Interfaces : 6](#_Toc532935399)

[2.1 Détection : 6](#_Toc532935400)

[2.2 Entrainement : 7](#_Toc532935401)

[3. Application : 7](#_Toc532935402)

Introduction

Le tracking dans une vidéo consiste à suivre un objet en mouvement (ou plusieurs objets) au fil du temps. Les applications du tracking sont variées notamment l’interaction homme-machine, sécurité et surveillance, communication et compression vidéo, contrôle du trafic…etc.

Le suivi vidéo peut prendre beaucoup de temps étant donné la quantité de données contenues dans la vidéo. Ajoutant à cela la complexité est le besoin possible d'utiliser des techniques de reconnaissance d'objet pour le suivi, un problème difficile en soi.

Problématique :

Le but de ce projet est de suivre les mouvements des objets colorés en utilisant des techniques de reconnaissance d'objet pour le suivi.

Chapitre I : Etat de l'art

## 1. OpenCv :

OpenCV (Open Source Computer Vision) est une bibliothèque proposant un ensemble de plus de 2500 algorithmes de vision par ordinateur, accessibles au travers d'API pour les langages C, C++, et Python. Elle est distribuée sous une licence BSD (libre) pour les plate-formes Windows, GNU/Linux, Android et MacOS.

## 2. Filtres morphologiques :

Le traitement d'image morphologique comporte un ensemble d’opérations non linéaires liées à la forme ou à la morphologie des éléments d'une image. L’idée des transformations morphologiques et de traiter un ensemble à l'aide d'un autre ensemble, appelé élément structurant, qui sert de sonde. À chaque position de l'élément structurant, on regarde s'il touche ou s'il est inclus dans l'ensemble initial. En fonction de la réponse, on construit un ensemble de sortie. On obtient ainsi des opérateurs de base qui sont relativement intuitifs.

### 2.1 Erosion :

L'érosion est l'un des deux opérateurs de base dans le domaine de la morphologie mathématique, l'autre étant la dilatation.

L'effet de base de l'opérateur sur une image est d'éroder les limites des régions des pixels de premier plan (c'est-à-dire typiquement des pixels blancs). Ainsi, les zones des pixels de premier plan rétrécissent et les vides dans ces zones deviennent plus grands. L’érosion se base sur des techniques ensemblistes. Elle effectue un **ET** logique entre l’image et l’élément structurant.

### 2.2 Délitation :

Comme pour l’érosion, la dilation a été conçue à la base pour les images binaires.

L'effet de base de l'opérateur sur une image est d'agrandir progressivement les limites des régions des pixels de premier plan. Ainsi, les zones des pixels de premier plan augmentent en taille alors que les vides dans ces régions deviennent plus petits. Cet opérateur se base aussi sur la théorie des ensembles et effectue un **OU** logique entre l’image et l’élément structurant.



figure 1: exemple d'érosion et de délitation

## 3. Thresholding :

Le seuillage d'image est la méthode la plus simple de segmentation d'image. Pour simplifier les choses on explique le sur une image sur le niveau de gris.

L'idée du seuillage est de simplifier les données visuelles pour l'analyse. Tout d'abord, il faut convertir en l’image en niveaux de gris. On considère que l’image a au moins 255 valeurs. Ce que le seuillage fait, au niveau le plus basique, est de tout convertir en blanc ou noir, en se basant sur une valeur « seuil ». Si par exemple nous voulons que le seuil soit 125 (sur 255), alors tout ce qui était inférieur ou égal à 125 serait converti en 0, autrement dit : en noir, et tout ce qui est au-dessus de 125 sera converti en 255, donc le blanc. Si on converti l’image en niveaux de gris comme, vous obtient du blanc et du noir. Si vous on ne convertit pas, on obtient des images seuillées, mais avec des couleurs, autre que le noir et le blanc.



figure 2: exemple de Thresholding

Chapitre I I : L'expérimentale

## 1. Méthodologie :

Pour atteindre notre but on a utilisé les méthodes basics expliqué ci-dessus enfin de construire un algorithme :

- Load HSV image from the cam feed

- Convert Image to HSV

- Adjust thresholding range using track-bars

- Apply thresholding

- Apply erosion

- Apply dilation

RE-DO steps till the noise get eliminated and the colored objects maintain their shapes

## 2. Interfaces :

Notre application est construite de deux fenêtre (détection et entrainement ).

### 2.1 Détection :

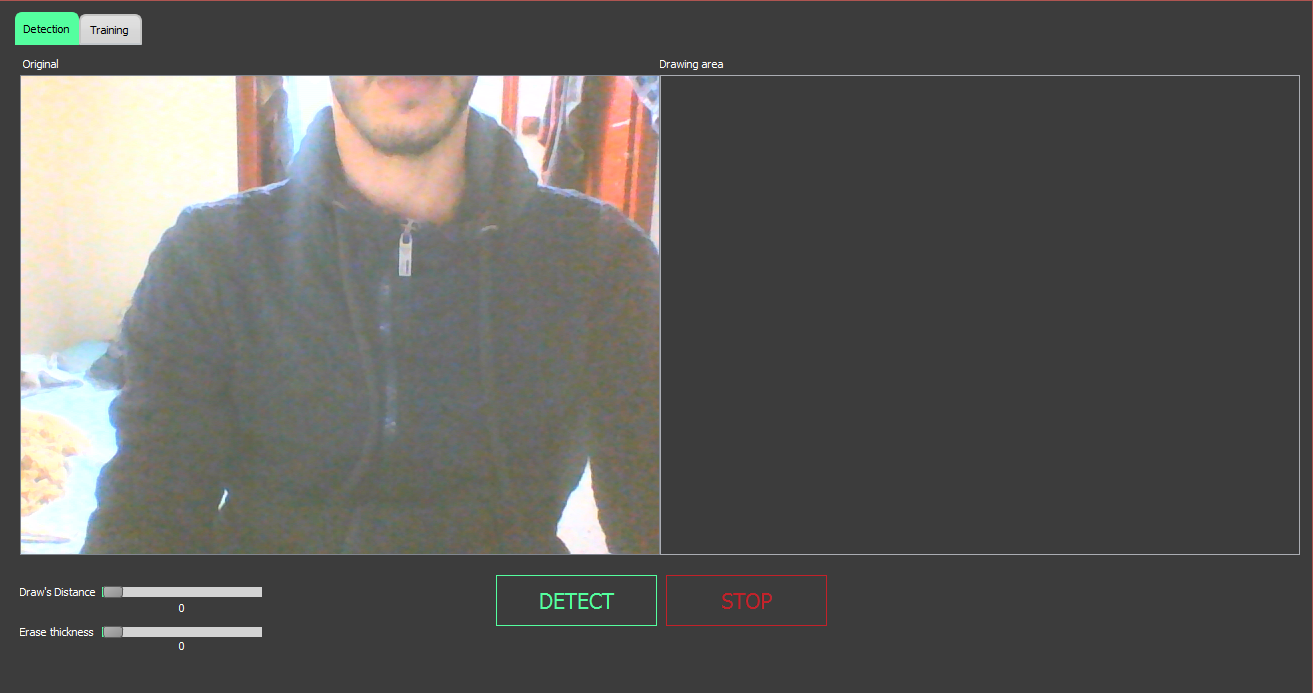
****

figure 3: Fenêtre de detection

**Zone1 :** Camera feed

**Zone2 :** Où on va dessiner.

**Zone3 :** Btn pour démarrer la détection.

**Zone4 :** Btn pour arrêter la détection.

**Zone5 :** Contrôle de distance entre les deux objects détecté.

**Zone6 :** Contrôle de l'épaisseur de la gomme.

### 2.2 Entrainement :

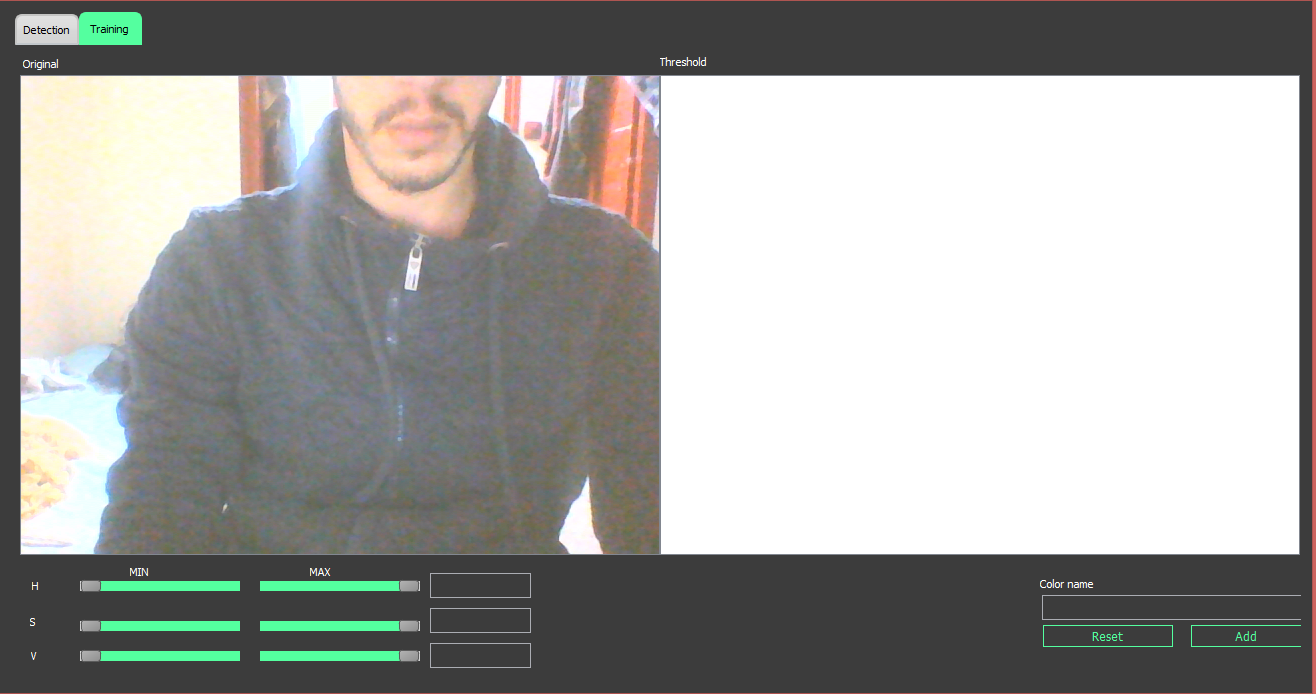
****

figure 4: Fenêtre d'entrainement

**Zone1 :** Camera feed.

**Zone2 :** Thresholding.

**Zone3 :** Track-bars pour l'ajustement des intervalles.

**Zone4 :** Permet d'ajouter et réinitialiser les couleurs.

## 3. Application :

On va utiliser dans notre expérimentale deux couleurs comme des couleurs a détectés et une autre pour la détection de la gomme.

On va ajuster les intervalles de chaque couleur ensuite on lance la détection.

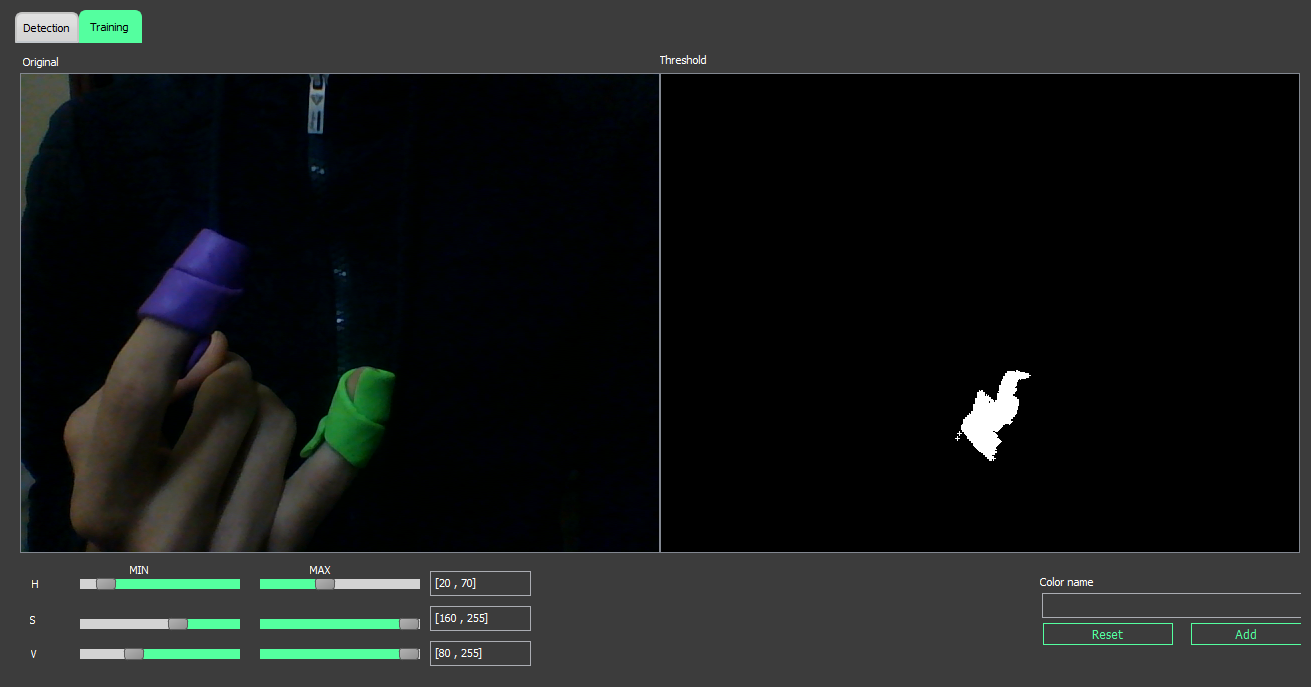


figure 5: Illustration sur Thesholding

On Lance la détection avec une distance fixé (Draw's Distance).

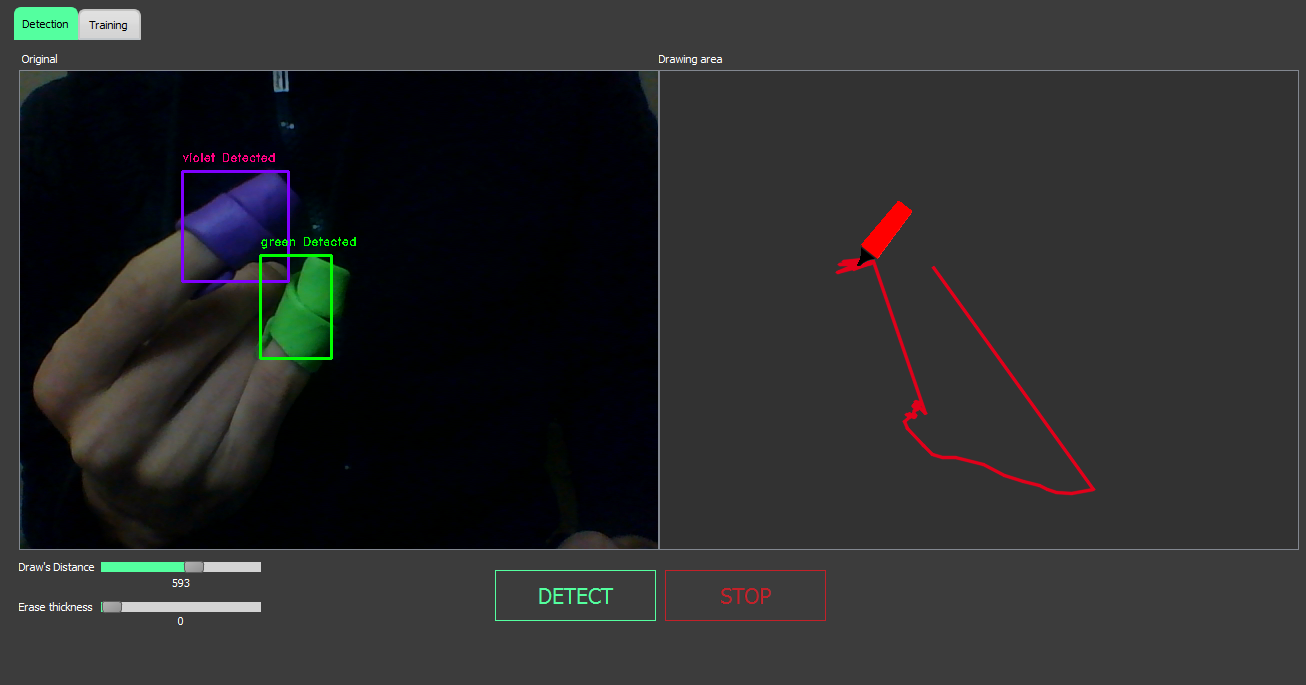
****

figure 6: Illustration sur la distance entre deux couleurs

Finalement, On ajoute la gomme pour ......

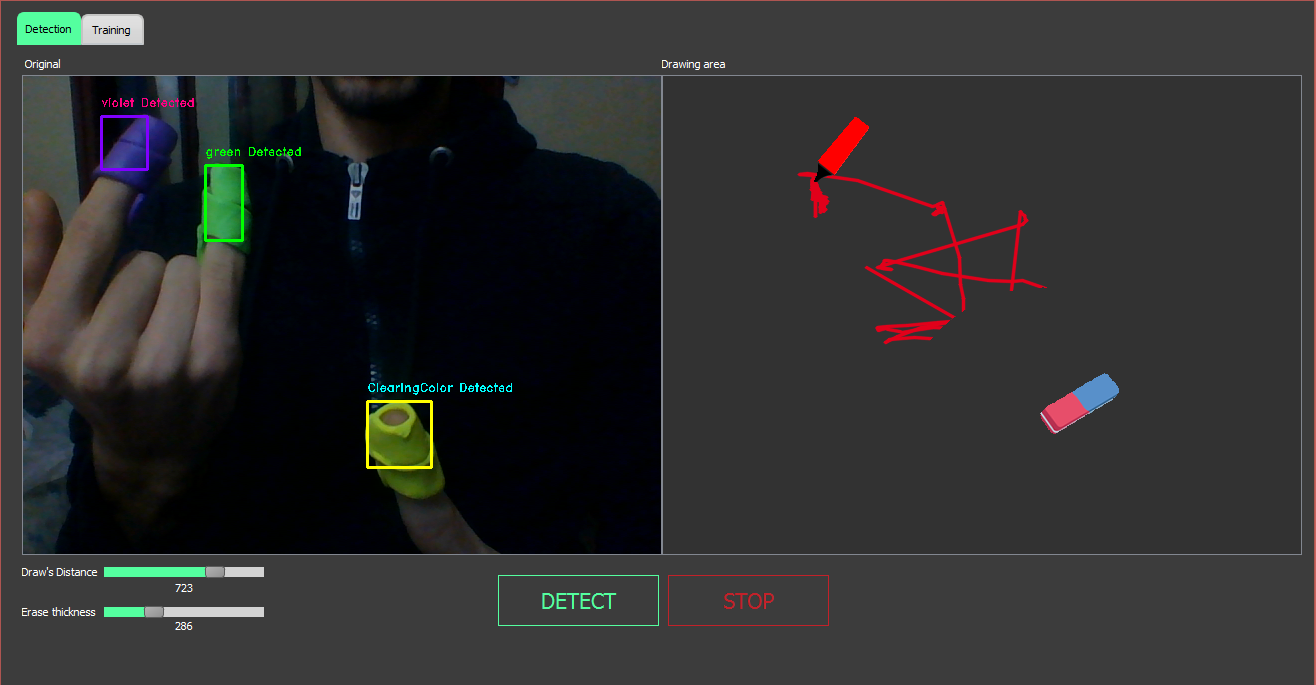
****

figure 7: Illustration sur La gomme