

Département Informatique Année :

4`eme année - Semestre 7 2022 - 2023

Conception générale

Analyse d’images de frelons

**Table des matières**

[1 Objectif du programme 3](#_Toc118563624)

[2 Interface et stockage 3](#_Toc118563625)

[2.1 Interface 3](#_Toc118563626)

[2.2 Stockage 3](#_Toc118563627)

[3 Fonctionnement du programme 3](#_Toc118563628)

[3.1 Prise de vue 3](#_Toc118563629)

[3.2 Isolation du sujet 3](#_Toc118563630)

[3.3 Analyse de l’image 3](#_Toc118563631)

[3.4 Obtention des résultats 4](#_Toc118563632)

[4 Principales fonctions 4](#_Toc118563633)

[4.1 color\_mask 4](#_Toc118563634)

[4.2 hornet\_length 5](#_Toc118563635)

[4.3 spacing 5](#_Toc118563636)

[4.4 abdomen\_shape 5](#_Toc118563637)

[4.5 XML\_writer 5](#_Toc118563638)

# Objectif du programme

Ce programme a pour but de reconnaître la caste du frelon (Fondatrice, ouvrière ou mâle) passant par le piège créé par CLEM Concept à partir d’une photo prise dans ce même piège. Le piège étant conçu pour être sélectif, on est assuré que tout insecte capturé est bien un frelon.

# Interface et stockage

## Interface

Le programme ne possèdera pas d’interface particulière, celui-ci ne nécessitant pas d’intervention humaine pour son fonctionnement. Le programme mettra en revanche à disposition un fichier sous le format xml contenant toutes les informations collectées.

## Stockage

Les résultats de chaque image prise seront enregistrés les uns après les autres dans un fichier XML comprenant le nom de la photo, la date et l’heure de la prise de vue, la référence du piège et la caste du frelon photographié (liste non-exhaustive sujette à modifications). Ce format permettra une utilisation de nos résultats facilitée pour tout système d’information voulant les exploiter. Pour cela, nous disposons d’un lecteur de carte SD qui recueillera ce fichier.

# Fonctionnement du programme

## Prise de vue

Lors du passage du frelon dans le piège, celui-ci est pris en photo par un dispositif intégré. Pour maximiser l’efficacité du futur traitement, nous avons demandé un fond blanc à l’endroit de cette prise de vue afin de mieux isoler le sujet. Afin de disposer d’une échelle de mesure, nous avons également d’une règle au sol gradué en centimètre.

## Isolation du sujet

Une fois la photo prise, celle-ci passe par un premier traitement par le programme : la fonction *colour\_mask.py* applique deux masques à l’image, un premier prenant en compte les couleurs de jaune à orange, puis un deuxième prenant en compte les couleurs marrons à noires. Ces pixels de couleur isolés, on additionne les deux masques pour obtenir une image ne contenant que le frelon.

## Analyse de l’image

Le sujet étant isolé, nous pouvons maintenant l’analyser afin de déterminer sa caste. Pour cela, on suit le tableau suivant :

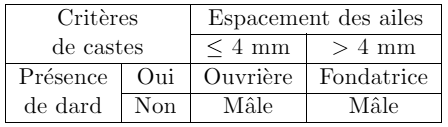


Figure 1 : Critères de décision de caste pour le frelon asiatique

Ce tableau est plutôt général ; en effet, selon la saison, les frelons peuvent changer de taille, ce qui peut fausser les résultats. Pour le moment, nous suivrons ces données car elles sont en général exactes.

Grâce à la librairie OpenCV qui contient beaucoup d’outils d’analyse d’image, le programme ajoute ces deux informations nécessaires au fichier XML de sortie.

## Obtention des résultats

Quand il le souhaite, l’utilisateur peut venir récupérer la carte SD présente dans le piège afin de récupérer les informations. Celle-ci peuvent être utilisées pour des statistiques ou des analyses diverses.

# Principales fonctions

Nous allons ici présenter les principales fonctions comme suit :

Entrée : [paramètres d’entrée]

Nécessite : [préconditions]

Sortie : [sortie]

Entraîne : [post-conditions]

Le programme étant sous python, nous indiquerons les types comme définis sur ce langage (par exemple, on écrira « str » pour une chaîne de caractère).

## color\_mask

Cette fonction est celle qui vient appliquer les masques orange-jaune et marron-noir sur l’image afin d’isoler le frelon du fond.

Entrée : img : np.ndarray(1) ; filename : str

Nécessite : img ≠ NULL ET filename ≠ NULL

Sortie : Ø

Entraîne : Création d’un fichier .jpg du sujet détouré

(1)np.ndarray : type provenant de la bibliothèque NumPy représentant l’image de manière multi-dimensionnelle (ici une matrice à 3 « dimensions » : X, Y et la couleur).

## hornet\_length

Cette fonction mesure le sujet.

Entrée : filename : str

Nécessite : filename ≠ NULL

Sortie : size : float

Entraîne : size ≥ 0

## spacing

Cette fonction mesure l’espacement des ailes du sujet.

Entrée : filename : str

Nécessite : filename ≠ NULL

Sortie : spacing : float

Entraîne : spacing ≥ 0

## abdomen\_shape

Cette fonction détermine la présence ou non d’un dard sur le sujet.

Entrée : filename : str

Nécessite : filename ≠ NULL

Sortie : sting : bool

Entraîne : Ø

## XML\_writer

Cette fonction écrit toutes les informations recueillies dans le fichier XML présent sur la carte SD.

Entrée : size : float ; spacing : float ; sting : bool ; filename : str

Nécessite : filename ≠ NULL

Sortie : Ø

Entraîne : Ecrit les informations dans le fichier XML