



SYNDICAT NATIONAL D'APICULTURE



# Challenge electro-apicole

## COMPTAGE PAR TRAITEMENT D'IMAGE DES VARROAS TOMBÉS SUR LE LANGE

Equipe Polytech Angers :

- Juliette Lucas
- Stanislas Launay



université  
angers



POLYTECH<sup>®</sup>  
ANGERS

# Table des matières

<b>Table des matières</b>	<b>2</b>
<b>Introduction</b>	<b>3</b>
<b>Partie 1 - Traitement d'image</b>	<b>3</b>
<b>Partie 2 - Étapes de sélection</b>	<b>5</b>
Étape 1 : Critère de surface	5
Étape 2 : Critère de forme : forme elliptique	6
Étape 3 : Critère : la couleur moyenne	7
Étape 4 : L'interface graphique	8
<b>Conclusion</b>	<b>8</b>

# Introduction

Pour répondre à la problématique qui nous a été posée concernant la détection et le comptage de varroas dans une ruche, nous avons identifié une solution que nous pouvions, à notre échelle, mettre en œuvre.

Grâce à nos recherches pour constituer l'état de l'art sur le comptage des varroas, nous nous sommes rendus compte que certaines applications/plateformes web existaient déjà pour compter le nombre de varroas à partir d'une photo. En revanche, ces solutions sont coûteuses, et ne permettent pas toujours de s'adapter à la luminosité de l'image. Pour mieux expliquer, deux photos prises à deux moments différents de la journée n'auront pas la même luminosité, surtout si la prise de vue est différente, et si le programme ne s'adapte pas, alors les varroas seront moins bien détectés.

Nous avons donc décidé dans un premier temps de créer un programme en langage Python qui reconnaît et compte les varroas sur une image donnée. Il y a une condition pour que ce programme fonctionne : il faut que sur la feuille des varroas, une pièce de 1€ soit posée. Elle va nous servir pour trouver la surface des varroas sur l'image, grâce aux proportionnalités, comme on connaît la surface d'une pièce et celle d'un varroa dans la réalité. Nous avons implanté dans ce programme plusieurs critères afin de reconnaître les varroas sur les feuilles, vous les découvrirez en détail ci-dessous. Dans un deuxième temps nous avons intégré ce programme sur une interface graphique, qui nous a permis de le tester et de le rendre accessible au plus grand nombre. Ce n'est pas la même chose qu'une application ni qu'une plateforme web, mais nous avons ajouté des fonctionnalités qui n'étaient pas présentes dans les solutions déjà existantes.

Notre interface permet non seulement à l'apiculteur de connaître le nombre de varroas sur une image qu'il aura importée, mais il a également à sa disposition des curseurs permettant de jouer sur la luminosité et le contraste de la photo, pour avoir en temps réel le nombre de varroas qui change et finalement obtenir le nombre optimal.

Nous allons tout au long de ce rapport vous expliquer en détail notre démarche pour arriver à ce résultat, et le travail que nous avons mis en œuvre.

## Partie 1 - Traitement d'image

Pour faire du traitement d'image, il existe de nombreuses fonctionnalités offertes par les bibliothèques en Python. Nous avons choisi certaines solutions, mais il faut savoir que ce n'était pas la seule et unique manière d'arriver à notre résultat, le traitement d'image est un sujet très large, qui permet d'aboutir à de nombreuses solutions par des chemins différents.

Notre démarche est la suivante :

1. Importation de l'image et application d'une valeur de luminosité et une autre de contraste (explication des calculs dans l'Étape 4). Ces deux valeurs sont celles que l'utilisateur pourra modifier grâce à l'interface proposée.

2. Transposition de l'image en niveaux de gris. Cette étape permet de récupérer l'histogramme de l'image, qui présente le nombre de pixels pour chaque valeur de gris. Une image en niveaux de gris est différente d'une image en noir et blanc.
3. Application d'un seuil. Il va nous permettre d'obtenir une image en noir et blanc, en mettant tous les pixels en dessous du seuil à 0 (noir), et tous ceux au-dessus à 1 (blanc). Explication par une image :

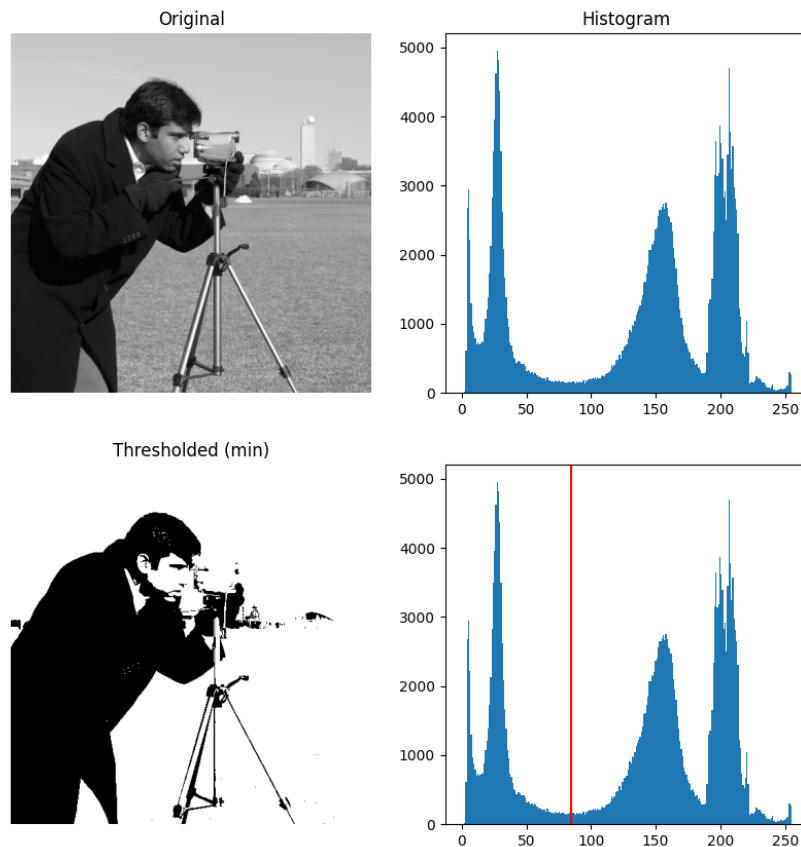


Image 1

4. Extraction des contours : on détecte tous les contours de toutes les formes sur l'image (dans le cadre d'une image seuillée, il sépare les pixels blancs des pixels noirs), cette étape permet de les séparer pour les exploiter plus tard.
5. Remplissage des trous : à cause des reflets, les contours peuvent ne pas être pleins après le seuillage. Il est alors nécessaire de remplir les trous pour que tous les contours traités soient pleins. Par exemple, pour la pièce de monnaie, les trous après seuillage sont fréquents (dus à la lumière reflétée par la pièce). Il est donc nécessaire de la remplir pour être précis dans nos calculs futurs (pour connaître la surface exacte de la pièce par exemple). On remplit les trous présents à l'intérieur des contours dessinés par l'extraction de contours.
6. Étiquetage des contours : on attribue à chaque contour une couleur différente, pour le différencier des autres et pouvoir le repérer plus tard.

Cette étape marque la fin du traitement d'image.

## Partie 2 - Étapes de sélection

Une fois que chaque contour de la feuille est détecté et isolé, nous avons décidé d'appliquer des filtres pour reconnaître les varroas. À chaque condition, certains sont éliminés, pour au final ne garder que les varroas. Grâce à l'état de l'art, nous avons pu définir les critères de sélection des varroas, ceux qui nous ont paru pertinents à aborder.

D'autres critères pourraient être ajoutés dans une démarche d'amélioration de notre travail.

- Critère 1 : la surface en pixels
- Critère 2 : la forme elliptique
- Critère 3 : la couleur moyenne dans le contour détecté

Nous avons bien entendu défini une marge d'erreur sur chacun des trois critères, de 5%, ce qui nous paraît être une valeur correcte. Par exemple, si la surface d'un varroa est de 100 pixels (taille calibrée grâce à la pièce de 1€), un contour détecté sera susceptible d'être un varroa uniquement si sa surface est comprise entre 95 et 105 pixels. De même pour les deux autres critères.

### Étape 1 : Critère de surface

Nous avons voulu rendre le programme applicable à n'importe quelle image, quelle que soit la distance à laquelle l'apiculteur a pris la photo. Il faut bien comprendre que si elle est prise de loin, les varroas sur la feuille auront un plus petit nombre de pixels que si la photo était prise de près. Si nous avons donné des valeurs fixes pour comparer les tailles en pixels, le programme aurait été applicable seulement pour certaines photos et pas pour d'autres.

Pour réaliser cela, nous avons exploité la pièce de monnaie de 1€ présente sur la feuille. Cette pièce, on en connaît la taille en mm, on connaît également la surface moyenne d'un varroa en mm<sup>2</sup>, donc dans la réalité, indépendamment de l'image.

Si on arrive à récupérer la surface en pixels de la pièce de monnaie sur la photo, avec les proportionnalités on peut très facilement obtenir la taille d'un varroa théorique en pixels sur cette photo en particulier.

Mais alors comment récupérer la pièce de 1€ sur l'image ? Notre démarche est la suivante : pour tous les objets détectés grâce au traitement d'image, nous sélectionnons uniquement ceux supérieurs à 1000 pixels (très gros objets). Ensuite, nous calculons la longueur du grand et du petit axe du contour de chacun des objets, et nous faisons le ratio :  $\frac{\text{grand axe}}{\text{petit axe}}$ .

Dans la réalité, le contour d'une pièce de monnaie est un cercle, son ratio est donc égal à 1. Si le ratio du contour de l'objet est plus ou moins égal à 1 (marge d'erreur puisque la forme peut être légèrement aplatie à cause de l'image), alors on considère que c'est la pièce de 1€, et nous pouvons l'exploiter.



Un exemple :

Diamètre pièce de 1€ (réalité) : 23.25 mm	Taille en pixels pièce de 1€ (image) : ... px
Grand axe varroa (réalité) : entre ... et ... mm	Taille en pixels varroa (image) : <b>X</b> px

Proportionnalités :

$$\frac{\text{Grand axe varroa (réalité)} \times \text{Taille en pixels pièce de 1€ (image)}}{\text{Diamètre pièce de 1€ (réalité)}} = X$$

Nous aurions pu utiliser le petit axe du varroa pour le calcul. Nous avons bien sûr utilisé un intervalle pour la taille des varroas dans la réalité, puisqu'ils sont tous différents. Cela s'applique par une valeur minimale et une valeur maximale pour le petit axe du varroa, et la même chose pour son grand axe.

## Étape 2 : Critère de forme : forme elliptique

Nous avons remarqué pendant nos recherches qu'un varroa a une forme elliptique, ce qui est pratique pour le différencier d'autres formes sur la feuille qui pourraient être du pollen ou des poussières par exemple.

En utilisant l'intervalle utilisé dans la partie précédente pour la taille réelle d'un varroa, mais aussi les proportionnalités, nous avons pu poser une condition pour différencier les varroas du reste des objets.

Nous avons réussi, pour chaque forme détectée par le programme, à récupérer son grand axe et son petit axe, ce qui nous a permis de calculer un ratio :  $\frac{\text{grand axe}}{\text{petit axe}}$ .

Nous avons comparé ce ratio à ceux obtenus grâce à la taille réelle d'un varroa. Cela signifie que nous avons posé deux ratios constants, ratio\_min et ratio\_max, avec les tailles minimales et maximales d'un varroa dans la réalité (avec min\_grand\_axe, max\_grand\_axe, min\_petit\_axe et max\_petit\_axe).

Il faut savoir qu'un ratio n'a pas d'unité, c'est un facteur qui représente simplement la différence entre le grand et le petit axe. Cela signifie que le ratio d'un varroa dans la réalité est le même que son ratio dans une image. C'est pour cette raison que nous pouvons les comparer.

Finalement la condition que nous avons posée est la suivante : si le ratio calculé dans l'image est inférieur à ratio\_min, ou bien s'il est supérieur à ratio\_max, l'objet sélectionné ne sera pas considéré comme un varroa.

Nous avons ajouté à cela une deuxième condition : en effet, en suivant notre raisonnement comme nous ne prenons que le ratio, un contour 10x plus grand qu'un varroa mais avec le même ratio serait accepté par le programme. Pour pallier ce problème, nous avons comparé le grand axe de chaque contour avec celui dans la réalité, que nous connaissons. Si le grand axe est compris dans l'intervalle [min\_grand\_axe ; max\_grand\_axe] (converti avec les proportionnalités en pixels) alors il est accepté pour la suite du programme.

## Étape 3 : Critère : la couleur moyenne

Grâce aux deux critères précédents, nous avons obtenu des objets de la même taille qu'un varroa, mais aussi de la même forme. Cela fait déjà deux grosses conditions, mais nous ne nous sommes pas arrêtés en si bon chemin. En effet, nous avons décidé de poser une dernière barrière pour réellement obtenir les varroas : la couleur.

Une particule de pollen pourra avoir la même forme qu'un varroa, la même taille, mais il sera beaucoup plus jaune, un varroa lui se rapproche plus du rouge voire du marron.

Nous avons exploré plusieurs pistes pour comparer les couleurs, la première était de regarder les composantes RVB (Rouge-Vert-Bleu) de chaque pixel. C'était pour nous la solution la plus logique, nous avons toujours appris que les couleurs étaient définies grâce à ces trois variables. En revanche, au fur et à mesure de leur utilisation, nous nous sommes vite trouvés débordés puisque c'est très compliqué de gérer trois valeurs différentes, une toute petite variation de la composante Rouge pouvant changer totalement la couleur générale de l'objet par exemple.

Nous nous sommes donc tournés vers les composantes TSV (Teinte-Saturation-Valeur) de chaque pixel, qui permettent de manier les couleurs beaucoup plus simplement. Ici on utilisera simplement la composante Teinte, ce qui est beaucoup plus simple comparée aux trois variables RVB.

De la même manière que pour les deux premières parties, nous voulons vraiment rendre cette condition personnalisable pour chaque image, ce qui signifie trouver un critère qui change à chaque nouvelle photo.

Pour cela, nous avons parcouru une première fois tous les objets qui avaient passé les deux premiers critères, donc tous les contours s'apparentant potentiellement à des varroas. Nous avons fait la médiane de la composante Teinte de chaque pixel de chaque contour, puis la médiane de toutes les médianes des contours réunis. Ainsi nous avons obtenu une valeur représentative de la Teinte d'un varroa. La médiane est plus appropriée ici que la moyenne, puisque par sa définition elle donne plus d'importance aux valeurs qui se ressemblent et qui sont plus nombreuses.

À cette étape du programme, nous considérons qu'il ne reste pratiquement que des varroas, donc le nombre d'autres objets est minime, la médiane permet de ne pas les prendre en compte. Nous sommes finalement sûrs d'avoir une valeur de référence proche de celle d'un varroa.

Mettons un moment la valeur médiane de la Teinte de côté. Nous allons maintenant parcourir une deuxième fois tous les objets qui ont passé avec succès les deux premières étapes. Nous prenons les composantes TSV de chaque pixel de chaque objet, plus particulièrement la Teinte, et nous la comparons à la valeur de référence que nous avons obtenue juste avant. Si, dans un objet, plus de 60 pixels sont différents de 75% de la valeur de référence, alors cet objet n'est pas considéré comme un varroa. Ces deux seuils, nous les avons choisis en testant de nombreuses fois avec différentes photos, pour obtenir le meilleur compromis. En effet, il peut arriver qu'avec l'effet de la photo, un varroa ait un reflet



sur lui, ce qui pourrait le faire être éliminé directement si nous n'avions pas mis un seuil de tolérance pour les pixels.

Notre programme Python est terminé, après l'application de ces trois critères nous considérons avoir obtenu tous les varroas et uniquement les varroas sur l'image.

## Étape 4 : L'interface graphique

Pour appliquer ce programme, nous avons réalisé une interface graphique, qui permet à l'utilisateur d'importer une photo et de connaître le nombre de varroas dessus. Il a également la possibilité de modifier plusieurs critères (luminosité, contraste) sur l'image pour qu'elle soit la plus claire et la plus nette possible.

Tout cela se concrétise simplement avec deux curseurs que l'utilisateur peut bouger, et également un bouton pour appliquer le programme avec les nouveaux critères. Cela signifie qu'au départ l'utilisateur importe une image, le programme compte le nombre de varroas dessus et l'affiche. Ensuite, grâce aux curseurs l'utilisateur peut modifier certains paramètres, et appuyer sur un bouton pour que le programme s'applique sur l'image modifiée. Cette étape peut être faite à l'infini, jusqu'à ce que l'utilisateur obtienne une image la plus claire et nette possible.

Les valeurs (luminosité et contraste) modifiées par l'utilisateur grâce aux curseurs sont utilisées dès le début du programme, c'est-à-dire pendant le traitement de l'image. Juste après l'importation, avant même de transposer l'image en niveaux de gris, on ajuste la valeur des pixels de l'image grâce aux deux valeurs.

Pour la luminosité, on ajoute simplement la valeur à tous les pixels, pour rendre l'image globale plus foncée ou plus claire. Pour le contraste, on multiplie chaque pixel par la valeur comprise entre 0 et 1, ce qui rend l'image plus ou moins nette en fonction de la valeur appliquée.

## Conclusion

Nous espérons avec tout ce rapport avoir bien décrit notre projet et nos différentes réalisations. Ces heures de réflexion et de programmation nous ont permis d'une part de progresser énormément et d'en apprendre beaucoup pour nos études et notre futur métier; et d'autre part cela nous a familiarisé avec le monde de l'apiculture et les problématiques auxquelles il est confronté. Nous avons pris conscience de l'importance de ce sujet, de l'importance de notre implication et des conséquences de la réussite du projet.

Nous avons réalisé cette solution dans le cadre des projets à l'école, ce qui représente presque 100 heures dans le deuxième semestre de notre année. Dans le temps limité qui nous a été imparti, nous avons développé une application de comptage de varroas à partir de 0. Certes notre application est perfectible mais nous avons volontairement laissé le choix à l'utilisateur de régler (via des curseurs) les seuils de détection car la gestion de ces seuils

dépend fortement des conditions de prise de vue des photos (éclairage etc). Dans le temps qui nous a été imparti, nous sommes fiers et satisfaits de vous présenter ce travail.