

PyImage Labeling

Les nouveautés : l'IA aide à annoter

Jeudi 22 octobre 2026

IA^2 2026 – Institut d'Automne en Intelligence Artificielle
Le Touquet-Paris-Plage, France
CRIL, Université d'Artois & CNRS

Pierre-Marie FAYETTE
Nicolas SZCZEPANSKI

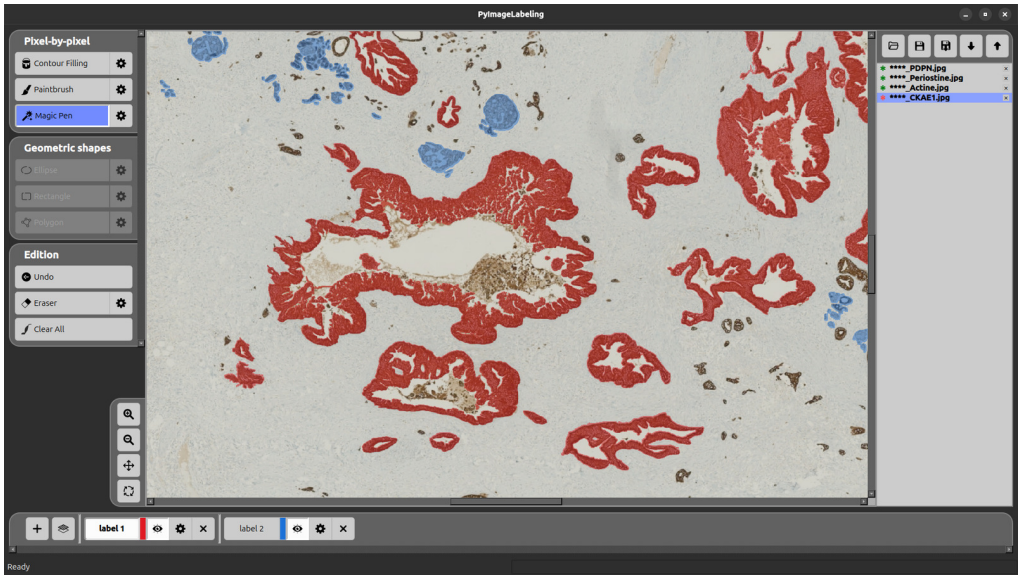








PyImageLabeling avant ces nouveautés



Annotation de coupes histologiques

- **Pixel par pixel** : pinceau, baguette magique, contours
- **Formes** : polygone, rectangle, ellipse
- **Mode ML** (PyTorch) : entraîner, prédire, exporter
- Images de plus de 100 Mo
- Libre (MIT) : Windows, Mac, Linux

Annoter reste long : l'expert **peint chaque pixel** lui-même

Les nouveautés en un coup d'œil

Fonction	Ce que fait l'expert	Technologie
SAM Assist	un clic ou une boîte sur l'objet	SAM 2.1, MedSAM
Texte vers masque	« *le chien rouge à gauche* »	Grounding DINO, CLIPSeg, CLIP, SAM
Propagation	un masque de référence pour tout le lot	DINOv2 + SAM, sans entraînement
Tête DINOv2	entraîne son modèle de segmentation	DINOv2 gelé + couche linéaire
Apprentissage actif	annote d'abord l'image où le modèle hésite	entropie des prédictions
Fiabilité	ne perd plus ses annotations	écritures atomiques, reprise, tests

L'IA **propose**, l'expert **valide** : rien n'est peint sans son accord

Développé par Pierre-Marie Fayette du 30 septembre au 5 octobre 2026. Disponible sur la branche `master` du dépôt GitHub, pas encore sur PyPI (dernière version publiée : 1.0.13).

Sous le capot

- **PyQt6** pour l'interface, **PyTorch** pour les calculs
- **Hugging Face `transformers`** pour SAM, DINO et CLIP : pas de compilation, fonctionne aussi sous Windows
- Modèles téléchargés **au premier usage**, puis gardés en cache local
- Calculs longs en **arrière-plan** : l'interface reste réactive
- Sur GPU, calcul en demi-précision : SAM 1,5 à 2 fois plus rapide

Modèle	Taille
SAM 2.1 (tiny à large)	156 à 898 Mo
MedSAM	375 Mo
Grounding DINO tiny	689 Mo
CLIPSeg	603 Mo
CLIP ViT-B/32	605 Mo
DINOv2 small / base	88 / 346 Mo

Poids publiés sur Hugging Face, utilisés sans réentraînement

Tout tourne **en local** : aucune image ne quitte la machine

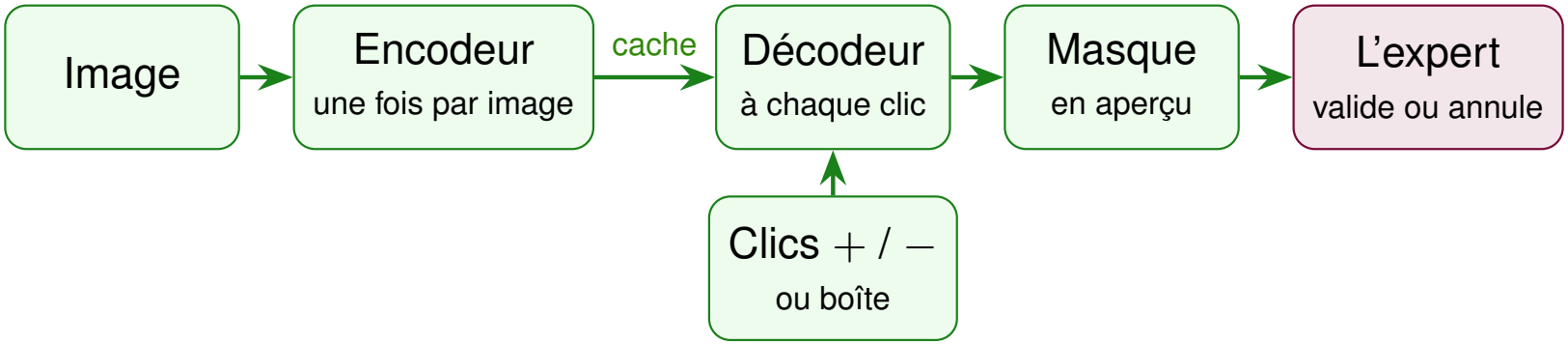
D'où viennent les modèles ?

Modèle (poids sur Hugging Face)	**Organisation**	**Publication**
SAM 2.1 `facebook/sam2.1-hiera-*`	Meta AI (FAIR)	Ravi et al., ICLR 2025
MedSAM `wanglab/medsam-vit-base`	Université de Toronto	Ma et al., Nature Comm. 2024
Grounding DINO `IDEA-Research/grounding-dino-tiny`	IDEA Research, Tsinghua	Liu et al., ECCV 2024
CLIPSeg `CIDAS/clipseg-rd64-refined`	Université de Göttingen	Lüddecke et Ecker, CVPR 2022
CLIP `openai/clip-vit-base-patch32`	OpenAI	Radford et al., ICML 2021
DINOv2 `facebook/dinov2-small, -base`	Meta AI (FAIR), Inria	Oquab et al., TMLR 2024
transformers (bibliothèque)	Hugging Face	Wolf et al., EMNLP 2020
PyTorch (bibliothèque)	Meta AI, PyTorch Foundation	Paszke et al., NeurIPS 2019

Des modèles **publiés et libres**, assemblés dans un outil d'annotation

Licences : Apache 2.0 pour tous les modèles, sauf CLIP (MIT). Homonymes : le DINO de Grounding DINO est un détecteur d'objets (IDEA), celui de DINOv2 une méthode d'apprentissage sans étiquettes (Meta).

SAM Assist : un clic, un masque



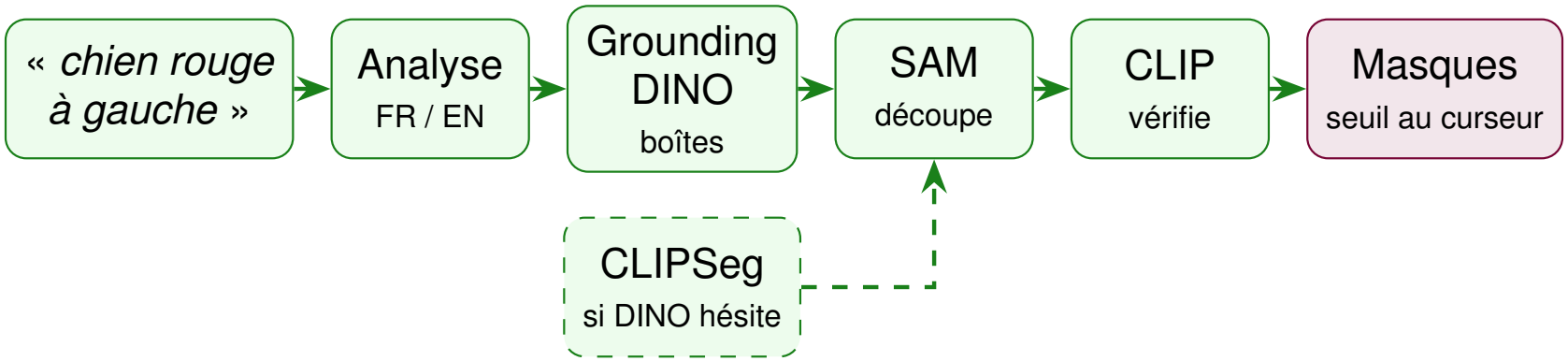
- **Clic gauche** : sur l'objet
- **Clic droit** : hors de l'objet
- **Glisser** : boîte englobante
- Plusieurs variantes de masque au choix

- **SAM 2.1** (Meta) : de *tiny* (rapide, CPU possible) à *large* (GPU)
- **MedSAM** : SAM affiné sur l'imagerie médicale, à utiliser avec des boîtes

Image analysée **une fois**, puis chaque clic est quasi instantané

SAM 2.1 : Meta AI (Ravi et al., ICLR 2025), successeur de SAM (Kirillov et al., ICCV 2023). MedSAM : Université de Toronto (Ma et al., Nature Communications 2024), SAM affiné sur 1,57 million de paires image-masque médicales.

Texte vers masque

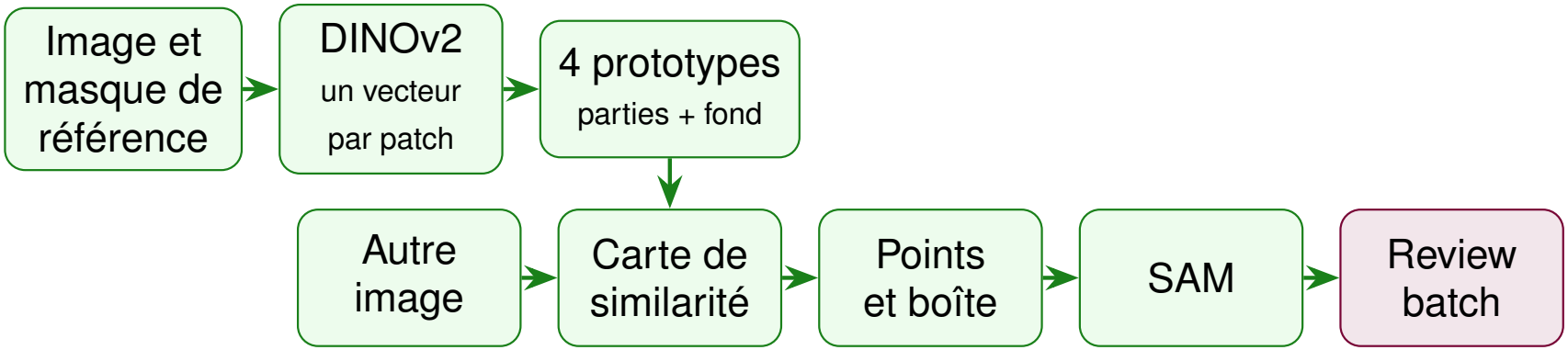


- **Analyse** locale, en français ou en anglais : objet, couleur, position (gauche, haut, coins...)
- **Grounding DINO** trouve les boîtes, à deux échelles pour les petits objets
- **CLIPSeg** complète quand DINO trouve peu de boîtes ou hésite
- **SAM** découpe chaque boîte, puis **CLIP** et la couleur écartent les mauvais candidats ; le **seuil** se règle ensuite au curseur, sans recalcul

Décrire l'objet suffit, **même en français**

Grounding DINO : IDEA Research (ECCV 2024). CLIPSeg : Univ. de Göttingen (CVPR 2022). CLIP : OpenAI (ICML 2021). Même association DINO + SAM que Grounded SAM (IDEA Research, 2024).

Propagation : un masque pour tout le lot

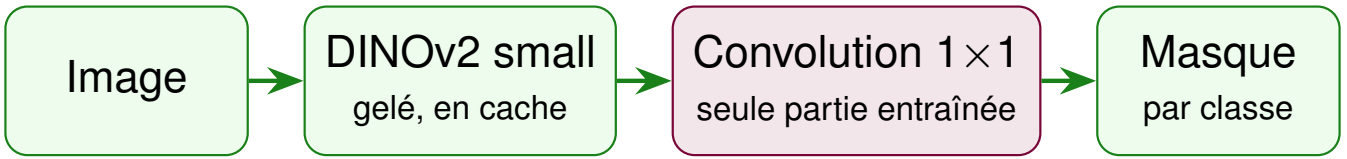


- **Sans entraînement** : DINOv2 retrouve l'objet, SAM le découpe
- Robuste : plusieurs **prototypes** (tête, corps...), image de référence **retournée** (animal tourné à gauche ou à droite), **deux échelles**
- Une image sans correspondance assez sûre est **ignorée**, pas mal annotée
- **Review batch** passe en revue les images peintes pour corriger

Annoter **une** image, relire **les autres**

DINOv2 : Meta AI (Oquab et al., TMLR 2024). Idée voisine : Matcher (Liu et al., ICLR 2024) associe aussi DINOv2 et SAM, sans entraînement. Les masques propagés s'ajoutent à ceux du label : partir d'un label vide.

Tête DINOv2 : un meilleur modèle, plus vite



0,83 mIoU

contre 0,62 avec la tête ResNet

≈ **19**×

entraînement plus rapide

- **Option** dans les réglages ML (*Segmentation Head*) : ResNet reste le choix par défaut
- Encodeur **partagé** avec la propagation : chaque image n'est encodée qu'une fois
- Segmentation seulement : avec des formes géométriques, retour automatique à ResNet

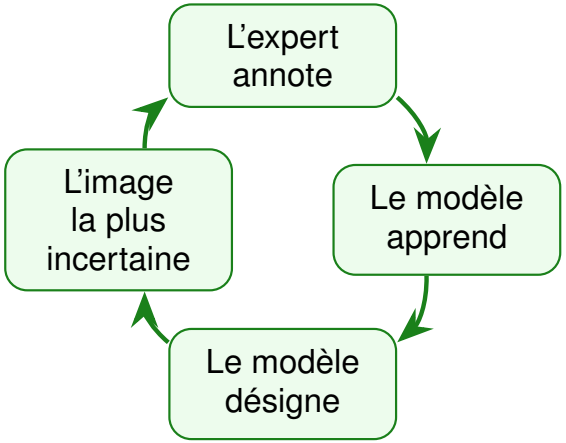
Mesuré par P.-M. Fayette sur des photos réelles annotées, même protocole pour les deux têtes (images, découpage, métrique). La couche linéaire sur DINOv2 gelé reprend le protocole d'évaluation de l'article DINOv2 (Oquab et al., TMLR 2024). ResNet : Microsoft Research (He et al., CVPR 2016).

Apprentissage actif

- Le modèle entraîné prédit sur chaque image **non annotée**
- Score d'incertitude : **entropie** des prédictions, normalisée entre 0 et 1

$$u(x) = \frac{1}{|x|} \sum_{\text{pixel}} \frac{-\sum_c p_c \log p_c}{\log C}$$

- **Suggest next to label** ouvre l'image la plus incertaine
- Découpage entraînement/validation et métriques de validation affichées



L'IA dans la boucle

Chaque annotation sert là où le modèle **en a le plus besoin**

Score calculé par le réseau ResNet (détection et/ou segmentation), C : nombre de classes. Échantillonnage par incertitude : Lewis et Gale (SIGIR 1994), synthèse dans Settles (2009).

Annoter au quotidien

Importer des masques existants

Binaire, indexé (*trimaps*) ou couleur RGB, avec un filtre sur les noms de fichiers (`Abyssinian*`). À l'ouverture d'un dossier, les masques sont reconnus : un label par classe.

Gomme intelligente

Critère de couleur optionnel, remplissage OpenCV, compteurs en direct.

Pinceau fluide

Les touches sont interpolées sur les tracés rapides : plus de trous, coût constant par tracé.

Annuler, sans surprise

Clear All devient une étape d'annulation comme les autres, et la profondeur d'historique s'applique à tous les calques.

Repartir des annotations **déjà faites**, au lieu de tout refaire

Ne plus perdre une annotation

- **Écritures atomiques** (fichier temporaire, puis renommage) : un crash ne laisse plus de fichier vide
- **Sauvegarde automatique restaurable** : proposée au démarrage, ou par *File* > *Recover Unsaved Annotations*
- Un fichier corrompu est **mis de côté** : le reste du projet se charge
- **Rapport de crash** avec les 400 dernières actions : *Help* > *Crash Report & Logs*
- Plus de **130 tests automatisés** (pytest, sans écran), dont un détecteur de noms non définis

Des heures d'annotation ne disparaissent plus **sur un crash**

Pour suivre la démo

```
git clone https://github.com/crillab/PyImageLabeling.git
cd PyImageLabeling
python3 -m pip install -e .      # installe aussi transformers
python3 -m PyImageLabeling
```

Jeu de données	Contenu	On y montre
Oxford-IIIT Pet	7 390 photos de chats et chiens, 37 races	SAM Assist, texte vers masque, propagation
CamVid	images de vidéos de rue, 32 classes	propagation d'une image à la suivante
Flood Area Segmentation	290 photos aériennes d'inondations	tête DINOv2, apprentissage actif

Au premier usage, les modèles sont téléchargés (88 Mo pour DINOv2 small, 156 Mo pour SAM 2.1 tiny), puis tout fonctionne hors ligne.

Ressources

- Site : `cril.univ-artois.fr/pyimagelabeling`
- Code, avec les nouveautés sur `master` : `github.com/crillab/PyImageLabeling`
- PyPI : `pypi.org/project/PyImageLabeling`

Code sur GitHub

`pip install PyImageLabeling` (licence MIT)

Références

Année	Lieu	Titre
2025	ICLR	SAM 2: Segment Anything in Images and Videos
2024	ECCV	Grounding DINO: Marrying DINO with Grounded Pre-Training for Open-Set Object Detection
2024	Nature Communications	Segment Anything in Medical Images *(MedSAM)*
2024	TMLR	DINOv2: Learning Robust Visual Features without Supervision
2024	ICLR	Matcher: Segment Anything with One Shot Using All-Purpose Feature Matching
2024	arXiv	Grounded SAM: Assembling Open-World Models for Diverse Visual Tasks
2023	ICCV	Segment Anything *(SAM)*
2022	CVPR	Image Segmentation Using Text and Image Prompts *(CLIPSeg)*
2021	ICML	Learning Transferable Visual Models From Natural Language Supervision *(CLIP)*
2020	EMNLP (démo)	Transformers: State-of-the-Art Natural Language Processing
2019	NeurIPS	PyTorch: An Imperative Style, High-Performance Deep Learning Library
2016	CVPR	Deep Residual Learning for Image Recognition *(ResNet)*
2009	Rapport technique	Active Learning Literature Survey *(B. Settles, Univ. du Wisconsin)*
1994	SIGIR	A Sequential Algorithm for Training Text Classifiers *(D. Lewis et W. Gale)*

Les modèles sont utilisés tels que publiés ; l'intégration dans PyImageLabeling est de P.-M. Fayette (CRIL).