Brief produit & marketing — améliorer la détection des scellés

Statut : brief partageable au 2026-05-18.

Audience : produit, marketing, go-to-market, fondateurs.

PDF : version PDF française

Message court

SKN a déjà un premier scanner lab pour produits Pokémon scellés. Il sait appeler un endpoint dédié, comparer une photo à un index de références scellées et retourner des candidats avec leurs métadonnées. Cela prouve que la direction technique est bonne.

Ce qui manque pour une détection fiable en conditions réelles n'est pas "un meilleur seuil". Il manque surtout quatre blocs produit :

1. un dataset d'évaluation terrain, séparé des images de référence ;
2. un catalogue de références scellées légalement exploitable ;
3. une détection de typologie et un crop adaptés aux packagings ;
4. un reranking avec OCR, métadonnées et feedback utilisateur.

La précision actuelle est donc normale pour un lab. Pour atteindre une expérience commercialisable, il faut transformer le scanner scellé en boucle produit mesurable : capturer, labelliser, évaluer, corriger, puis enrichir.

Ce qui existe aujourd'hui

Le backend expose déjà un chemin scellé dédié :

- endpoint `POST /scan-sealed-lab` ;
- tables séparées `sealed_products`, `sealed_product_images`, `sealed_product_embeddings_lab` ;
- embeddings visuels DINOv2 sur image complète ;
- recherche vectorielle pgvector sur les références scellées ;
- filtres optionnels par type produit et langue ;
- diagnostics top-K, runner-up, score et raison d'ambiguïté ;
- mode mobile scellé branché dans l'application de test ;
- écran Scan Debug qui permet de regarder les candidats.

Le lab utilise deux types de sources :

- `ptcg-assets` pour prouver le concept et bénéficier d'une taxonomie large de packagings ;
- des références `skn-owned` issues de photos possédées par SKN, utilisables en debug privé.

Ce socle est utile, mais il n'est pas encore une promesse d'accuracy marché. Les photos propres de catalogue et les vraies captures téléphone ne posent pas le même problème.

Pourquoi les scellés sont plus difficiles que les cartes

Les cartes raw ont un format stable. Un produit scellé n'a pas cette simplicité.

Les difficultés principales :

- les formes changent : booster, blister, display, ETB, tin, collection box, deck, bundle ;
- les packagings ont du plastique, des reflets, des coins abîmés, des pliures et des fonds variés ;
- deux produits peuvent partager le même artwork mais avoir une langue, un SKU ou un format différent ;
- les photos utilisateur sont souvent en angle, partielles, prises sur écran ou mal éclairées ;
- les images de référence disponibles publiquement sont souvent des packshots propres, pas des photos terrain ;
- les droits d'image empêchent de redistribuer librement certaines références ;
- une erreur de type produit peut envoyer le matching vers la mauvaise famille, par exemple booster au lieu de blister.

Conséquence produit : la bonne UX ne doit pas seulement afficher "trouvé / pas trouvé". Elle doit savoir dire "candidat probable", "ambigu", "besoin de confirmation", ou "choisir parmi ces candidats".

Ce qu'il manque pour améliorer la détection

1. Dataset terrain non biaisé

Il faut alimenter `sealed-eval` avec des photos réelles qui ne sont jamais indexées comme références. Sans cela, un score élevé peut simplement prouver que le système reconnaît une image qu'il a déjà vue.

Minimum utile pour un POC solide :

Niveau	Volume cible	Usage
POC utile	30 à 50 photos réelles par classe prioritaire, environ 200 à 350 photos	Savoir où le scanner échoue vraiment
Bêta stable	100 à 200 photos par classe, environ 700 à 1 400 photos	Mesurer par typologie et langue
Niveau produit	300+ photos par classe, 2 000+ photos cumulées	Couvrir les cas difficiles et suivre la régression

Classes prioritaires :

- booster pack ;
- blister ;
- booster display ;
- Elite Trainer Box ;
- collection box ;
- tin / mini-tin ;
- slab ;
- carte raw comme négatif ou fallback.

Chaque photo doit avoir au minimum :

- nom produit attendu ;
- type produit ;
- langue / région si connue ;
- source et droits : `owned_by_skn` ou `explicit_permission` ;
- capture source : téléphone réel, capture écran, photo de pack en main, etc. ;
- statut : inbox non labellisé ou labellisé `evaluationOnly: true`.

2. Catalogue de références production-safe

`ptcg-assets` est utile pour le lab interne, mais ce n'est pas une source production-safe à elle seule. Le repo n'a pas de licence déclarée et contient probablement des images de packaging protégées.

Pour la production, il faut au moins une des options suivantes :

- permission écrite d'une source d'assets ;
- fournisseur commercial avec conditions compatibles ;
- images possédées par SKN ;
- images uploadées par l'utilisateur avec consentement clair ;

- politique d'affichage qui ne renvoie pas les images candidates quand les droits ne sont pas approuvés.

Sans catalogue de références robuste, le modèle peut être bon mais incapable de trouver le bon produit parce que la bonne référence n'existe pas.

3. Détection de typologie et crop scellé

Aujourd'hui, le lab encode souvent l'image entière. Cela fonctionne pour un premier test, mais ce n'est pas optimal pour un téléphone réel.

La première amélioration livrée sans nouvelles images d'entraînement est `preprocess=sealed` : un crop de premier plan OpenCV avant DINOv2. Il réduit le bruit de table, main, étagère ou écran dans les captures téléphone tout en gardant le ratio du packaging. Les premiers tests téléphone après déploiement montrent une amélioration qualitative nette de la précision, parce que la recherche visuelle reçoit un signal produit plus propre et moins de bruit de fond. Ce n'est toujours pas un détecteur entraîné ; l'étape YOLO ou segmentation réelle demandera encore des données de détection labellisées et des poids entraînés, et l'amélioration doit être quantifiée sur le benchmark terrain séparé `sealed-eval` avant de devenir une promesse marché.

Il faut un détecteur qui sait localiser les grandes familles :

- `card` ;
- `slab` ;
- `sealed` ;
- `graded_label` ;
- `barcode` ;
- puis, plus tard : `booster_pack`, `blister`, `booster_display`, `elite_trainer_box`, `collection_box`, `tin`, `booster_bundle`.

Le rôle de YOLO ou d'un équivalent n'est pas d'identifier "Base Set Blastoise Booster FR". Son rôle est de trouver la zone utile et la typologie. L'identité exacte reste le travail du retrieval visuel, de l'OCR, des métadonnées et du reranking.

Impact produit :

- moins de faux positifs entre familles ;
- scan plus rapide parce qu'on compare dans une famille plus petite ;
- meilleure UX caméra, avec un cadre adapté aux packagings et pas seulement aux cartes.

4. OCR ciblé et reranking hybride

L'OCR pleine image est fragile sur les packagings brillants. Il doit arriver après le retrieval visuel.

Pipeline cible :

```
photo utilisateur
-> détection typologie / crop
-> top-K visuel DINOv2 + pgvector
-> OCR ciblé sur titre, langue, code produit, code barre, label slab
-> reranking hybride
-> résultat, ambiguïté ou choix utilisateur
```

L'OCR est particulièrement utile pour :

- distinguer deux variantes avec le même artwork ;
- séparer FR / EN / DE / JP ;
- lire un nom de produit, un code de set ou un code barre ;
- améliorer les slabs et labels gradés plus tard.

5. Boucle de feedback dans l'app

Le scanner doit devenir un outil d'apprentissage produit.

À chaque scan scellé, l'app devrait pouvoir capturer :

- résultat accepté ;
- résultat corrigé ;
- "aucun de ceux-là" ;
- candidat choisi dans le top-K ;
- photo et contexte si l'utilisateur accepte l'usage d'amélioration.

Cela permet de prioriser les améliorations par volume réel au lieu de débattre à l'aveugle.

6. Métriques lisibles par produit

Le benchmark doit être lu avec des métriques simples :

Métrique	Ce qu'elle dit
Top-1 exact accuracy	Le bon produit est-il premier ?
Top-K recovery	Le bon produit est-il au moins parmi les candidats ?
Ambiguous-neighbor rate	Le système hésite-t-il trop souvent ?
Wrong-product-type rate	Le scanner compare-t-il contre la mauvaise famille ?
Wrong-language rate	Les variantes langue/SKU posent-elles problème ?
False positive rate	Le système affirme-t-il trop vite un mauvais résultat ?
P50/P95 latency	Est-ce acceptable pendant un scan mobile ?
Correction success	L'utilisateur peut-il sauver un scan imparfait ?

Les nouveaux champs `summary.diagnoses` et `summary.nextActions` dans le benchmark servent justement à grouper les échecs : OCR/reranking, typologie/crop, langue/références, ou manque de référence.

Ce que produit et marketing peuvent faire maintenant

Collecte terrain

Priorité immédiate : créer une base de photos réelles, possédées ou autorisées.

Utiliser le format de collecte des photos terrain scellées pour la structure des lots, les colonnes CSV, les règles de nommage et la metadata de droits.

Bonnes photos :

- produit entier visible ;
- plusieurs angles ;
- lumière naturelle et lumière difficile ;
- reflet plastique ;
- fond de table, main, étagère ;
- photo légèrement ratée mais réaliste ;
- capture d'un écran si c'est un cas utilisateur probable.

À éviter pour le test set :

- uniquement des packshots propres ;
- uniquement des photos parfaites ;

- réutiliser une image de référence comme image d'évaluation ;
- mélanger des images sans preuve de droits.

Positionnement

Formulation saine aujourd'hui :

- "scanner scellé en lab" ;
- "détection assistée avec candidats et confirmation" ;
- "portfolio scellé-first" ;
- "amélioration continue via benchmark terrain".

Formulations à éviter tant que la bêta n'est pas mesurée :

- "100% accurate" ;
- "reconnaît tous les scellés Pokémon" ;
- "base de prix parfaite" ;
- "identification automatique garantie".

Décisions à prendre

Produit et marketing doivent aider à trancher :

1. Quelles typologies scellées sont prioritaires pour la démo : boosters, ETB, displays, blisters ?
2. Quelle langue est prioritaire : FR d'abord, EN ensuite, ou international dès le début ?
3. Peut-on obtenir des droits ou partenariats sur des images de références ?
4. Quel niveau de confiance est acceptable avant affichage public ?
5. Faut-il assumer une UX de confirmation plutôt qu'un résultat unique au début ?

Roadmap recommandée

Phase	Objectif	Résultat attendu
1. Dataset terrain	200 à 350 photos réelles labellisées	Savoir où la reconnaissance échoue
2. Filtres typologie UI	L'utilisateur ou l'app indique booster / ETB / blister	Moins de mauvaises familles
3. Références FR/EN safe	Ajouter des références possédées ou licenciées	Moins de produits absents
4. Détecteur crop	Localiser sealed / slab / card puis cropper	Meilleur signal visuel
5. OCR reranking	Lire texte utile sur top-K	Moins d'ambiguïtés
6. Feedback app	Corriger et apprendre depuis les vrais scans	Amélioration continue

Conclusion

La détection scellée peut devenir un vrai différenciateur SKN parce que peu d'apps traitent correctement les produits scellés comme des objets first-class. Mais l'avantage ne viendra pas d'un seul modèle magique. Il viendra d'un système complet : références propres, photos terrain, typologie, OCR, métriques, correction utilisateur et règles légales claires.

Le prochain meilleur investissement est donc la collecte labellisée et l'évaluation honnête, pas une promesse marketing prématurée.

Pages liées

- Lab scanner scellé
- Stratégie d'optimisation de la reconnaissance
- Registre des risques licences et sources d'assets
- Revue UI/UX concurrentielle du scanner