

# ÉCOLE NATIONALE DE LA STATISTIQUE ET DE L'ADMINISTRATION ÉCONOMIQUE



2024-2025  
PROJET STATAPP

---

## Deep learning pour la détection d'installations solaires individuelles à Madagascar

---

Maxime CHANSAT  
Malo DAVID  
Thomas LAMBELIN  
Kwame MBOBDA-KUATE

*Sous la direction de :*  
Gabriel KASMI (Mines Paris)



## **Remerciements**

Nous tenons à exprimer notre gratitude à Gabriel KASMI pour son accompagnement attentif et sa disponibilité tout au long du projet. Son regard critique et sa capacité à clarifier des points techniques parfois complexes nous ont été d'une aide précieuse. Au-delà de son expertise, la qualité des échanges et l'ambiance de travail qu'il a su instaurer ont joué un rôle moteur dans notre progression et dans la dynamique collective du groupe.

Avec tous nos remerciements,  
Maxime, Malo, Thomas et Kwame.

# Table des matières

<b>1 Cartographier le solaire diffus : un impératif pour les réseaux du XXI<sup>e</sup> siècle</b>	<b>2</b>
1.1 Un réseau électrique sous tension : variabilité, complexité, imprévisibilité . . . . .	2
1.2 Le tournant du solaire diffus : un basculement silencieux... . . . . .	2
1.3 .. et l'émergence d'une nécessité de détection . . . . .	3
1.4 Typologie des méthodes algorithmiques de détection . . . . .	4
<b>2 Automatiser la détection d'installations solaires : fondements des réseaux neuronaux et état de l'art des modèles existants</b>	<b>5</b>
2.1 L'apprentissage profond . . . . .	5
2.1.1 Le perceptron : la brique élémentaire . . . . .	5
2.1.2 Les réseaux de neurones . . . . .	5
2.2 Les réseaux de neurones convolutifs . . . . .	6
2.2.1 Idée générale . . . . .	6
2.2.2 Les composantes d'un CNN . . . . .	7
2.2.3 Un exemple d'architecture . . . . .	8
2.3 Les métriques et l'entraînement . . . . .	9
2.3.1 Fonctions de perte . . . . .	9
2.3.2 Minimisation du risque empirique . . . . .	9
2.3.3 Descente de gradient . . . . .	9
2.3.4 Choix des métriques de performances . . . . .	10
2.4 Historique de modèles de détection de panneaux photovoltaïques sur images aériennes . . . . .	12
2.4.1 Avant le deep learning, forêts aléatoires . . . . .	12
2.4.2 Utilisation du deep learning . . . . .	12
2.4.3 DeepSolar . . . . .	13
2.5 Apprentissage par transfert . . . . .	13
<b>3 Un large éventail de données en amont de la construction du réseau de neurones convolutifs</b>	<b>14</b>
3.1 Contexte . . . . .	14
3.2 Présentation des variables les plus utiles . . . . .	14
3.3 Description des images . . . . .	14

<b>4 Détection automatique par réseaux convolutifs : mise en œuvre et résultats</b>	<b>15</b>
4.1 Prétraitement et enrichissement des données . . . . .	15
4.2 Architectures et choix d'implémentation . . . . .	16
4.3 Résultats expérimentaux et choix d'évaluation . . . . .	16
4.3.1 Évolution des performances du segmenteur : comparaison de deux modèles et discussion critique . . . . .	16
4.3.2 Évaluation du modèle de classification : stabilité, optimisation du seuil et pouvoir discriminant . . . . .	18
4.3.3 Le pipeline complet . . . . .	19
<b>A Exemple d'images de drones</b>	<b>24</b>
<b>B Architecture du Excel</b>	<b>25</b>
<b>C Diverses représentations graphiques des performances du modèle de classification</b>	<b>26</b>

# Notations

Notations relatives aux réseaux de neurones :

- $w_i$  : i-ème valeur du vecteur poids du perceptron
- $a_i$  : i-ème valeur d'entrée du perceptron
- $a'$  : valeur de sortie du perceptron
- $b$  : biais du perceptron
- $L$  : nombre de couches du réseau de neurones
- $q_\ell$  : nombre de neurones dans la couche  $\ell$  du réseau
- $W^{[\ell]} = (w_{i,j}^{[\ell]})_{1 \leq i \leq q_\ell, 1 \leq j \leq q_{\ell-1}}$  : matrice de dimension  $q_\ell \times q_{\ell-1}$  contenant les poids reliant la couche  $\ell - 1$  à la couche  $\ell$
- $B^{[\ell]} = (b_1^{[\ell]}, \dots, b_{q_\ell}^{[\ell]})^T$  : vecteur de taille  $q_\ell$  contenant les biais de la couche  $\ell$
- $A^{[\ell]} = (a_1^{[\ell]}, \dots, a_{q_\ell}^{[\ell]})^T$  : vecteur de taille  $q_\ell$  contenant les sorties de la couche  $\ell$
- $Z^{[\ell]} = (z_1^{[\ell]}, \dots, z_{q_\ell}^{[\ell]})^T$  : vecteur de taille  $q_\ell$  contenant les sorties de la couche avant la fonction d'activation  $\ell$
- $C^{[\ell]} = (c_{i,j}^{[\ell]})$  : matrice de convolution contenant les poids reliant la couche  $\ell - 1$  à la couche  $\ell$
- $*$  : produit de convolution

Notations relatives au risque empirique :

- $l$  : fonction de perte
- $g$  : fonction de prédiction appartenant à l'ensemble  $\mathcal{G}$
- $X$  : variable aléatoire représentant une observation (ou un vecteur de caractéristiques), prenant ses valeurs dans l'ensemble  $\mathcal{X}$
- $Y$  : variable aléatoire représentant la réponse (ou l'étiquette) associée à  $\mathcal{X}$ , prenant ses valeurs dans l'ensemble  $\mathcal{Y}$

# Introduction

À Madagascar, comme dans une grande partie de l’Afrique subsaharienne, l’accès à l’électricité reste marginal : moins de 40% de la population est raccordée au réseau national en 2023, et moins de 15% la même année pour les zones rurales (GBM [2023]). Ce faible taux d’électrification n’est pas seulement un indicateur de retard économique. Il conditionne directement les possibilités d’accès à l’éducation, aux soins, au numérique et, plus généralement, à tout ce que l’on associe au développement. En l’absence d’infrastructures publiques étendues et d’un réseau électrique centralisé, les panneaux solaires individuels se multiplient. Discrètement, et souvent sans aucune trace officielle, ils apportent un minimum d’autonomie énergétique dans des zones où l’État est peu présent. L’Agence Internationale de l’Énergie estime que la puissance délivrée par panneaux photovoltaïques à Madagascar est ainsi passée de nulle en 2017 à 53GWh (Agence internationale de l’énergie).

Ce développement, bien qu’essentiel, échappe largement aux outils traditionnels de planification. À ce jour, on ne dispose d’aucune cartographie fiable et exhaustive des installations solaires individuelles dans le pays. Et sans données précises, difficile de savoir où concentrer les efforts : faut-il renforcer l’offre, ou au contraire constater qu’elle se structure déjà de manière autonome ? Quels sont les profils géographiques ou socio-économiques des zones déjà équipées, et de celles qui restent à la marge ? Toutes ces questions restent ouvertes. Pourtant, elles sont fondamentales pour anticiper la trajectoire de la transition énergétique dans un pays comme Madagascar ainsi que pour déterminer, à terme, la puissance installée réellement disponible. Une telle information, croisée avec des prévisions météorologiques, permettrait d’estimer plus finement la production effective d’électricité, facilitant ainsi l’intégration du pays dans les marchés régionaux de l’électricité et contribuant à une allocation plus efficiente des ressources énergétiques.

C’est là qu’intervient l’imagerie satellite. Elle permet, en principe, de détecter à grande échelle et sans intervention humaine directe la présence de panneaux solaires. Dans d’autres contextes – urbains, normés, bien documentés – cette approche a déjà fait ses preuves (Malof et al. [2016]). Des méthodes basées sur le *deep learning*, et notamment sur les réseaux de neurones convolutifs, ont été utilisées pour cartographier les installations solaires aux États-Unis, en Europe ou en Australie, avec des résultats fiables. Mais le contexte malgache présente des défis uniques : faible densité urbaine, habitats très variés, toitures de formes et de matériaux peu standardisés, images souvent bruitées.

Ce rapport explore donc la problématique suivante : peut-on, à partir d’images satellites et d’algorithmes de *deep learning*, détecter précisément les installations photovoltaïques individuelles à Madagascar ? Quels algorithmes sont les plus pertinents à cet effet ? Nous nous appuyons sur une base de donnée composée d’images satellites et de drones et sur plusieurs modèles de détection pour interroger les limites de ce type d’approche. Au-delà de la faisabilité algorithmique, c’est aussi la robustesse du modèle à un contexte hétérogène et peu documenté qui est mise à l’épreuve.

Les enjeux plus larges d’une telle détection – en matière de politique énergétique, d’aide au développement, ou encore de gouvernance locale – seront abordés dans la première partie. Les aspects techniques, eux, seront détaillés par la suite, depuis les fondements du *machine learning* (et plus spécialement du *deep learning* et des CNN) jusqu’au choix

des architectures. Cette introduction se contente de poser le cadre : un besoin urgent de données, une technologie prometteuse, et entre les deux, un terrain d’expérimentation exigeant.

## 1 Cartographier le solaire diffus : un impératif pour les réseaux du XXI<sup>e</sup> siècle

### 1.1 Un réseau électrique sous tension : variabilité, complexité, imprévisibilité

Le système électrique repose sur un principe fondamental : l’instantanéité de l’équilibre entre production et consommation. Contrairement à d’autres formes d’énergie, l’électricité se stocke difficilement à grande échelle ; elle doit donc être consommée au moment même où elle est produite et dans des quantités égales (RTE). Dès lors, tout déséquilibre peut avoir des conséquences importantes sur la stabilité du réseau : surtension, sous-tension, voire effondrement local ou généralisé de l’approvisionnement. Le *black-out* qui a frappé le Texas en février 2021 illustre cette fragilité : des températures extrêmes ont provoqué une hausse de la demande d’énergie tandis que le froid impactait sa production.

Dans ce contexte, la transition énergétique, loin d’être un simple remplacement de technologies, bouleverse les fondements de l’architecture du réseau. En introduisant massivement des sources météo-dépendantes (éolien, photovoltaïque), elle remplace des moyens pilotables, historiquement centralisés, par des moyens décentralisés dont la production dépend de conditions météorologiques difficilement maîtrisables. Le solaire en particulier, assez variable à l’échelle infra-journalière (comme annuelle), constitue un défi d’autant plus grand que sa croissance s’opère dans un cadre hétérogène : toitures domestiques, bâtiments publics, entrepôts, parkings, etc.

En outre, le réseau électrique est complexe et souvent segmenté. En France, la production est assurée d’une part par des entreprises (comme EDF), tandis que le transport de cette électricité revient à RTE, et enfin sa distribution depuis les centres de consommation vers les foyers à Enedis. L’émergence des installations photovoltaïques individuelles transforme ce schéma **production→transport→distribution** et oblige les organisations de gestion comme RTE à repenser la répartition et l’équilibre offre-demande à l’échelle nationale. Si à Madagascar le réseau entier, de la production à la distribution, est centralisé par la JIRAMA, d’autres problématiques compromettent sa stabilité et l’intégration des sources d’énergies diffuses et individuelles. En effet, cette dernière subit de lourds revers financiers et de nombreuses affaires de corruption depuis plusieurs décennies de monopole (World Bank [2023], RFI [2022]).

### 1.2 Le tournant du solaire diffus : un basculement silencieux...

Loin de rester cantonnée à quelques pays industrialisés, l’adoption du solaire diffus – c’est-à-dire installé de manière décentralisée sur des toits, hangars ou espaces individuels – s’accélère à l’échelle mondiale. Selon l’IRENA, la puissance installée photovoltaïque des structures décentralisées a quadruplé dans le monde entre 2015 et 2024, et quintuplé en

Afrique sur la même période (IRENA [2025]). Ce basculement tient à plusieurs facteurs conjoints : la baisse drastique du coût des panneaux, l'amélioration de leur rendement, la simplification des processus d'installation, et l'engouement politique ou citoyen pour une énergie plus locale, visible et perçue comme autonome (Shakeel et al. [2023]).

À Madagascar, ce tournant est moins une opportunité qu'une nécessité. Le taux d'électrification reste l'un des plus bas au monde (comme mentionné en introduction), avec de fortes inégalités territoriales (GBM [2023]). Face à un réseau vétuste, souvent défaillant, et à la difficulté qu'éprouve la JIRAMA à satisfaire la demande, les panneaux solaires individuels apparaissent comme la seule voie réaliste d'accès à une électricité minimale (World Bank [2023]). Leur prolifération est très marquée, avec une augmentation de la puissance installée des structures décentralisées estimée de 1769% entre 2015 et 2024 (IRENA [2025]).

Mais cette transition reste difficile à quantifier. Faute de suivi centralisé, d'enregistrement systématique ou de cartographie à jour, l'ampleur réelle du solaire diffus échappe aux gestionnaires comme aux chercheurs (Kasmi et al. [2022]). Or, sans connaissance fine de cette réalité énergétique émergente, il est impossible d'adapter les réseaux, de planifier des extensions ou de concevoir des politiques cohérentes d'électrification.

### 1.3 .. et l'émergence d'une nécessité de détection

Ce basculement silencieux, fondé sur la multiplication d'installations solaires individuelles, soulève un défi structurel : ces équipements, bien que croissants, échappent en grande partie à l'observation institutionnelle. Contrairement aux centrales solaires raccordées, les installations diffuses sont souvent non télémesurées, mal recensées, et dispersées hors des radars des gestionnaires de réseau. En l'absence de registres complets, ni leur nombre exact, ni leur puissance cumulée, ni leur localisation ne sont connus avec précision<sup>1</sup> (Kasmi et al. [2022]).

Cette opacité complique la planification énergétique, l'équilibrage en temps réel du réseau, et le développement d'outils de prévision fiables (Mayer et al. [2022]). Pour y répondre, la détection automatique des panneaux à partir d'images satellites ou aériennes est devenue une piste active de recherche, mobilisant notamment le *machine learning*. Elle vise à restituer, de manière indépendante, une cartographie aussi fine que possible de la capacité photovoltaïque hors réseau officiel, afin d'informer à la fois les opérateurs, les autorités publiques et les acteurs du marché de l'électricité (Yu et al. [2018]). Par ailleurs, des registres publics comme aux Pays-Bas intègrent déjà des dispositifs de détection automatisée. Le Kadaster, agence nationale de cartographie néerlandaise, a mis en place une base de données couvrant l'ensemble du territoire, construite à partir d'images aériennes à très haute résolution traitées par des réseaux de neurones convolutifs (notamment TernausNet, voir Kausika et al. [2021]). Dans la prochaine section, nous nous proposons de faire un tour d'horizon des principales méthodes de détection automatiques existantes.

---

1. Des registres existent, notamment en Allemagne (*Marktstammdatenregister*) où la déclaration de toute installation photovoltaïque diffuse est obligatoire, mais la qualité de leurs données est discutée (Kotthoff et al. [2023]).

## 1.4 Typologie des méthodes algorithmiques de détection

Déetecter les installations photovoltaïques à partir d'images aériennes ne relève pas d'un unique procédé, mais d'un éventail de stratégies, adaptées aux données disponibles et à la précision visée. On distingue schématiquement deux logiques principales.

La première consiste à poser le problème comme une **classification** : chaque image, ou segment d'image, est analysé pour établir si elle contient ou non des panneaux. Cette approche, simple à mettre en œuvre, fournit une cartographie binaire utile pour des estimations globales, du nombre d'installations notamment, mais ne permet pas d'identifier l'emplacement ou l'étendue exacte des installations.

La seconde, plus fine mais également plus coûteuse en calcul, repose sur la localisation explicite des panneaux, on parle de **segmentation**. Il ne s'agit plus seulement de détecter leur présence, mais de les délimiter dans l'espace, pixel par pixel. Cela suppose de recourir à des méthodes complexes, comme le *deep learning*, capables de capturer la structure visuelle des objets dans des environnements variés. Ces techniques, parmi lesquelles les réseaux convolutifs occupent une place centrale, permettent non seulement d'extraire la forme des panneaux, mais aussi d'estimer indirectement leur capacité ou leur orientation (voir par exemple DeepSolar, le premier modèle de ce genre Yu et al. [2018])

D'autres approches, fondées sur des modèles plus classiques comme les arbres de décision, ont été explorées pour des usages ciblés (Malof et al. [2016]). Cependant, ces méthodes ne fournissent pas d'information concernant la forme ni, a fortiori, la surface des installations photovoltaïque, ce qui ne permet pas de répondre à l'enjeu d'estimation de la puissance réelle installée.

Dans la prochaine section, nous nous proposons d'examiner plus en détail le fonctionnement des réseaux de neurones convolutifs et les raisons de leur efficacité dans ce type de tâche (notamment de segmentation sémantique et de classification). Nous discuterons également à cette occasion de l'état de l'art des modèles existants.

## 2 Automatiser la détection d'installations solaires : fondements des réseaux neuronaux et état de l'art des modèles existants

### 2.1 L'apprentissage profond

L'apprentissage profond, ou *deep learning*, est une branche de l'intelligence artificielle qui repose sur des réseaux de neurones artificiels (dont la structure est inspirée du fonctionnement du cerveau) pour traiter et apprendre des données complexes. Ces réseaux, composés de plusieurs couches de **perceptrons**, permettent de modéliser des relations non linéaires et d'effectuer des tâches variées, telles que la reconnaissance d'images ou la prédiction.

Nous commençons par examiner le perceptron, l'élément de base des réseaux de neurones, dont nous détaillerons ensuite le fonctionnement. Enfin, nous nous concentrerons à partir de la section 2.2 sur les réseaux de neurones convolutifs, qui sont largement utilisés dans des applications comme la détection d'installations photovoltaïques.

#### 2.1.1 Le perceptron : la brique élémentaire

Le perceptron est l'élément de base d'un réseau de neurones, à l'instar d'un neurone pour le cerveau. Il prend en entrée un vecteur de variables (noté  $(a_i)_{i \in \llbracket 1, n \rrbracket}$  ci-dessous) qu'il multiplie vectoriellement par un vecteur de poids  $((w_i)_{i \in \llbracket 1, n \rrbracket}$ ), auquel il ajoute finalement un biais ( $b$ ). À ce stade, le modèle reste linéaire. Afin d'ajouter de la non-linéarité, ce résultat est passé en argument d'une fonction d'activation (Bodin and Recher [2021]). Des exemples classiques de fonctions d'activation sont la fonction ReLU et la sigmoïde (ci-dessous notée  $\sigma$ )<sup>2</sup>. Mathématiquement, un perceptron est donc de la forme :

$$a' = \sigma\left(\sum_{i=1}^n w_i \cdot a_i + b\right) \text{ où } \sigma(x) = \frac{1}{1 + e^{-x}} \text{ pour sigmoïde et } \max(0, x) \text{ pour ReLU}$$

#### 2.1.2 Les réseaux de neurones

Pour construire un réseau de neurones, on assemble plusieurs perceptrons en couches successives. Dans une même couche, chaque perceptron reçoit le même vecteur en entrée et produit une sortie. L'ensemble des sorties forme alors un vecteur, transmis à la couche suivante. En empilant plusieurs couches ainsi reliées, on obtient un réseau dit dense ou entièrement connecté.

Notons également que sous certaines conditions portant sur la profondeur du réseau, le nombre de neurones et le choix des poids, un tel réseau est capable d'approximer toute fonction continue sur un compact avec une précision arbitraire (Bodin and Recher [2021]).

---

2. D'autres fonctions d'activation existent comme la linéaire :  $\sigma(x) = (x - x_{min}) / (x_{max} - x_{min})$ .

Formellement, à la couche  $\ell$  du réseau, on a :

$$Z^{[\ell]} = W^{[\ell]} \cdot A^{[\ell-1]} + B^{[\ell]} \quad \text{où} \quad A^{[\ell]} = \sigma(Z^{[\ell]})$$

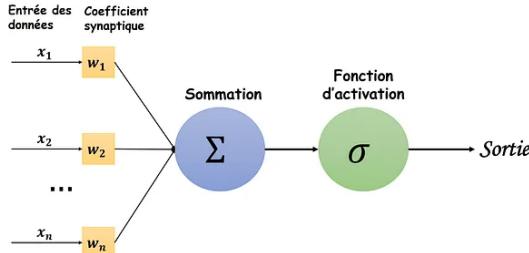


FIGURE 1 – Schéma fonctionnel d'un perceptron (Luminis [2023]).

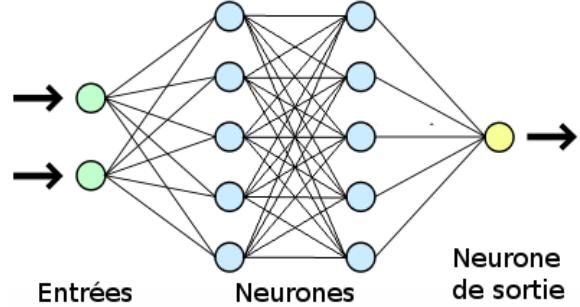


FIGURE 2 – Schéma de l'architecture d'un réseau de neurones (Semeurand [2021]). Présentant ici une couche d'entrée, deux couches cachées entièrement connectées et une couche de sortie.

## 2.2 Les réseaux de neurones convolutifs

Les images sont des objets particuliers qui nécessitent des méthodes d'analyse adaptées, notamment lorsqu'il s'agit de les traiter automatiquement (Prince [2023]). Elles présentent en effet trois caractéristiques fondamentales. D'abord, leur dimension est très élevée : une image standard de  $1920 \times 1080$  pixels contient plus de deux millions de points, chacun associé à trois valeurs de couleur (rouge, vert, bleu), soit plus de six millions d'entrées au total. Ensuite, les images possèdent une structure spatiale forte : les pixels proches sont généralement corrélés, et les motifs visuels pertinents – comme les bords ou les textures – reposent sur cette proximité locale. Enfin, elles sont stables par rapport à certaines transformations géométriques : une translation, une rotation ou une légère déformation de l'image n'en altèrent pas fondamentalement le contenu. Ces propriétés rendent les approches classiques par réseaux entièrement connectés inadaptées. Non seulement ces dernières ne tiennent pas compte de la structure spatiale locale, mais elles impliquent un nombre colossal de paramètres. Les réseaux de neurones convolutifs (CNN, pour *Convolutional Neural Networks*) répondent précisément à ces enjeux, et sont donc amplement utilisés (Prince [2023]).

### 2.2.1 Idée générale

Les réseaux convolutifs s'appuient sur la notion de convolution discrète, une opération qui consiste à appliquer un filtre (ou noyau) de petite taille à des sous-régions locales de l'image. Contrairement aux réseaux entièrement connectés, où chaque neurone reçoit l'ensemble de l'image en entrée, ici le filtre se déplace sur l'image et traite successivement des zones limitées, en appliquant les mêmes poids à chaque position. Cette structure locale et partagée permet deux choses essentielles : réduire drastiquement le nombre de paramètres du modèle et assurer une forme d'invariance par translation, puisque les motifs détectés sont identifiés indépendamment de leur position dans l'image.

La formule d'une convolution simple est la suivante :

$$Z^{[\ell]} = C^{[\ell]} * A^{[\ell-1]} + B^{[\ell]} \quad \text{avec } * : \text{le produit de convolution}$$

## 2.2.2 Les composantes d'un CNN

### A. Les couches convolutionnelles

Comme introduit précédemment, une couche convolutionnelle repose sur l'opération de convolution discrète. Elle est caractérisée par quatre paramètres essentiels.

Le premier est la **taille du noyau**, généralement un carré de dimension  $k \times k$ . Ce noyau détermine la zone de l'image sur laquelle la convolution est appliquée. Le deuxième paramètre est la **dilatation**  $d$ , représentée en figure 4. Elle permet d'élargir le champ réceptif du filtre sans en augmenter la taille ni le nombre de paramètres, en insérant des espacements entre les éléments du noyau. Vient ensuite la **stride**  $s$ . Il s'agit du pas de déplacement du filtre sur l'image. Une *stride* de  $s$  signifie que le noyau est appliqué tous les  $s$  pixels, ce qui réduit la résolution de la sortie. Enfin, le **padding**, illustré en figure 5, consiste à ajouter une bordure (souvent remplie de zéros, c'est à dire de pixels noirs) autour de l'image afin de contrôler la taille de la sortie. Cela permet notamment d'éviter une réduction progressive des dimensions au fil des convolutions (Prince [2023]).

### B. Le sous-échantillonnage et sur-échantillonnage

Le sous-échantillonnage est une technique permettant de réduire les dimensions des données d'entrée afin d'alléger les calculs tout en conservant l'essentiel de l'information (Zhao and Zhang [2024]). La méthode la plus courante est le *pooling*, illustré en figure 6.

Deux types de *pooling* sont fréquemment utilisés : le *max pooling*, qui retient la valeur maximale dans chaque région, et le *average pooling*, qui en calcule la moyenne.

À l'inverse, le sur-échantillonnage permet d'augmenter la taille d'une entrée, généralement via des techniques d'interpolation ou de duplication.

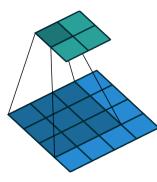


FIGURE 3 – Convolution sans padding (Cnam [n.d.]).

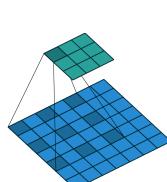


FIGURE 4 – Convolution avec dilatation (Cnam [n.d.]).

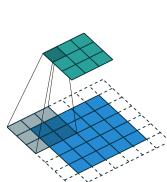


FIGURE 5 – Convolution avec padding (Cnam [n.d.]).

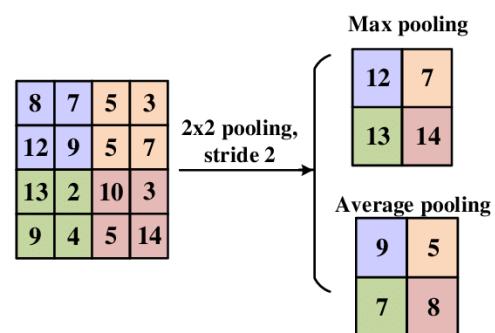


FIGURE 6 – Illustration d'une opération de pooling (Cnam [n.d.]).

## C. Les canaux

Une seule convolution ne suffit généralement pas à capturer toute la richesse d'une image, car elle ne peut extraire qu'un type limité de caractéristiques (Prince [2023]). Pour pallier cette perte d'information, plusieurs convolutions distinctes sont appliquées en parallèle, chacune avec ses propres paramètres. Cela génère plusieurs canaux, chacun capturant des motifs différents dans l'image. La figure 7 illustre ce mécanisme : plusieurs convolutions produisent des canaux distincts, qui peuvent ensuite être combinés ou réduits selon les besoins de l'architecture.

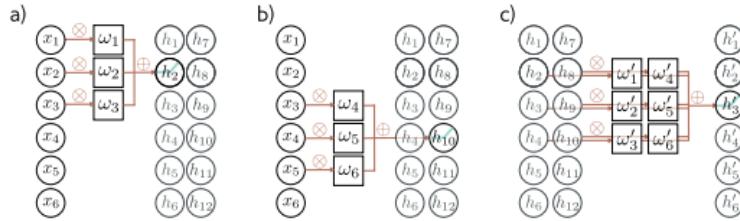


FIGURE 7 – Schéma d'une convolution multiple. a) Une première convolution est appliquée pour créer le premier canal. b) Une deuxième convolution, avec des paramètres différents, permet d'obtenir le second canal. c) Une troisième convolution est utilisée pour réduire le nombre de canaux.

### 2.2.3 Un exemple d'architecture

Nous nous penchons ici sur l'architecture d'un réseau de neurones convolutifs à travers un exemple emblématique : AlexNet. Ce réseau, vainqueur du concours ImageNet en 2012, a marqué un tournant en démontrant l'efficacité des CNN dans la reconnaissance d'images, et constitue depuis une référence majeure dans le domaine (Alom et al. [2018], Alex Krizhevsky [2012]).

Un CNN est composé de plusieurs couches de convolution, qui permettent d'extraire les caractéristiques pertinentes à partir des données d'entrée. Ces couches sont souvent suivies de couches de sous-échantillonnage (*pooling*). Après cette phase d'extraction des caractéristiques, les données passent par des réseaux entièrement connectés, qui combinent les caractéristiques extraites pour réaliser des tâches telles que la classification, la détection, ou la segmentation sémantique.

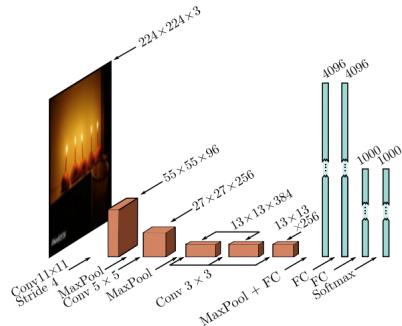


FIGURE 8 – Schéma de l'architecture d'un AlexNet.

## 2.3 Les métriques et l'entraînement

### 2.3.1 Fonctions de perte

La fonction de perte (*loss function* en anglais),  $\ell$ , est la fonction que l'on cherche à optimiser dans notre modèle. D'un point de vue conceptuel, elle est analogue à la vraisemblance en statistique.

Les fonctions de perte les plus courantes sont :

- Pour la classification : l'entropie croisée binaire (BCE pour *Binary Cross-Entropy Loss*), qui évalue la divergence entre les distributions de probabilité prédites et réelles.
- Pour la segmentation : la BCE également et la distance de Jaccard (qui est liée à l'indice de Jaccard<sup>3</sup>, présenté en 2.3.4).

### 2.3.2 Minimisation du risque empirique

Une fois la fonction de perte choisie, l'objectif de l'entraînement est de minimiser le risque empirique associé. Pour rappel le risque d'une fonction de prédiction  $g : \mathcal{X} \rightarrow \mathcal{Y}$  est défini par :

$$R_P(g) = \mathbb{E}_P[\ell(Y, g(X))].$$

Cependant, la distribution  $\mathcal{P}$  des données étant inconnue, le risque est estimé par son équivalent empirique :

$$\hat{R}_n(g) = \frac{1}{n} \sum_{i=1}^n \ell(Y_i, g(X_i))$$

Ainsi l'entraînement consiste à trouver la fonction minimisant ce risque empirique, soit :

$$\hat{g}_{n,\mathcal{G}} = \arg \min_{g \in \mathcal{G}} \hat{R}_n(g)$$

Le principal risque lors de l'apprentissage est le sur-apprentissage, qui survient lorsque le modèle s'ajuste trop fortement aux données d'entraînement, au détriment de sa capacité à généraliser. Cela provient du fait que le risque réel est approximé par le risque empirique, qui peut favoriser des modèles trop complexes. Pour limiter ce phénomène, on ajoute un terme de pénalisation à la fonction de perte, afin de contraindre la complexité du modèle. Cela permet de trouver un compromis entre biais (lié à une modélisation trop simple) et variance (lié à une modélisation trop flexible), et ainsi de favoriser une meilleure généralisation.

### 2.3.3 Descente de gradient

Une fois la fonction de perte choisie, un sous-ensemble de fonctions de prédiction fixé et une éventuelle pénalisation ajoutée, il reste à déterminer une méthode pour identifier la meilleure fonction de prédiction dans cet ensemble.

---

3. La distance de Jaccard vaut  $J(A, B) = 1 - IoU(A, B)$ , où l'IoU est une métrique à maximiser. Cette reformulation permet donc de transformer le problème en une minimisation, conformément à la logique des fonctions de perte

Dans le cas des CNN, l'ensemble des fonctions de prédiction correspond à l'ensemble des combinaisons possibles des poids du réseau. L'objectif devient donc l'optimisation de ces poids pour minimiser le risque empirique. Pour cela, on utilise la méthode de descente de gradient. Elle consiste à calculer la dérivée du risque empirique par rapport à chaque paramètre du modèle, puis à ajuster les poids dans la direction opposée au gradient.

Dans le cas des réseaux de neurones, cette procédure s'appelle la rétro-propagation (ou *backpropagation*). Elle repose sur le fait que, pour calculer la dérivée du risque par rapport aux paramètres d'une couche  $i$ , on utilise les dérivées calculées dans les couches suivantes. En effet, considérons un réseau de neurones défini de la manière suivante :

$$\begin{aligned}\forall l \leq c, \quad Z^{[l]} &= W^{[l]} \cdot A^{[l-1]} + B^{[l]} \\ \forall i \leq c, \quad A^{[l]} &= \sigma[Z^{[l]}] \\ \ell &= \ell[A^{[c]}, Y]\end{aligned}$$

Alors

$$\frac{\hat{R}_n(g)}{\partial W^{[l]}} = \left[ \left( W^{[l+1]}\top \frac{\hat{R}_n(g)}{\partial Z^{[l+1]}} \right) \odot \sigma'(Z^{[l]}) \right] A^{[l]}\top$$

où  $\odot$  est le produit terme à terme

Ainsi les poids sont mis à jour en fonction du gradient et du taux d'apprentissage  $\eta$  :

$$W_{i+1}^{[l]} \leftarrow W_i^{[l]} - \eta \frac{\hat{R}_n(g)}{\partial W^{[l]}}$$

#### 2.3.4 Choix des métriques de performances

Les métriques de performance permettent d'évaluer la qualité des prédictions réalisées par un modèle, qu'il soit de régression, de classification ou de segmentation. Elles sont essentielles pour comparer différents modèles et ajuster les paramètres afin d'optimiser les résultats.

Dans notre cas, le modèle vise à détecter des panneaux photovoltaïques à partir d'images satellites et aériennes. À cet effet, comme évoqué en 1.4 et justifié en 2.4.2, on recourt à deux approches différentes et complémentaires : la segmentation sémantique, qui consiste à assigner une classe à chaque pixel (PV ou non-PV) afin de délimiter localement ces derniers, et la classification. Cette distinction nécessite l'emploi de métriques spécifiques à chacune de ces tâches.

#### A. Les métriques en segmentation

L'évaluation des modèles de segmentation repose sur des métriques quantifiant la qualité du découpage des objets dans une image. Contrairement à la classification, où une seule étiquette est attribuée à chaque échantillon, la segmentation assigne une catégorie à chaque pixel, rendant l'évaluation plus complexe.

### **Intersection over Union (IoU)**

L'IoU (ou indice de Jaccard) est la métrique la plus couramment utilisée pour mesurer le chevauchement entre la région prédite et la vérité terrain (*ground truth*). Elle est définie comme :

$$IoU = \frac{\text{Prédiction} \cap \text{Vérité terrain}}{\text{Prédiction} \cup \text{Vérité terrain}}$$

L'IoU varie entre 0 (aucun chevauchement) et 1 (correspondance parfaite). Elle est utilisée dans des applications nécessitant une segmentation précise, comme la vision par ordinateur pour la détection d'objets, l'imagerie biomédicale ou encore la surveillance satellitaire.

### **Mean Intersection over Union (mIoU)**

Dans le cas de la segmentation multi-classes, l'IoU est calculée individuellement pour chaque classe, puis moyennée sur l'ensemble des classes :

$$mIoU = \frac{1}{N} \sum_{i=1}^N IoU_i$$

où  $N$  est le nombre de classes et  $IoU_i$  est l'IoU de la classe  $i$ . Cette métrique fournit une évaluation plus globale du modèle.

### **Dice Coefficient (F1-score de segmentation)**

Le *Dice Coefficient*, également appelé coefficient de Sørensen-Dice, est une alternative à l'IoU, particulièrement populaire en segmentation d'images médicales. Il est défini comme :

$$Dice = \frac{2 \times |\text{Prédiction} \cap \text{Vérité terrain}|}{|\text{Prédiction}| + |\text{Vérité terrain}|}$$

Il est similaire à l'IoU mais donne plus d'importance aux pixels en intersection, ce qui est utile pour des applications où la distinction des contours est cruciale.

## **B. Les métriques en classification**

Plusieurs métriques sont utilisées pour évaluer les performances d'un modèle de classification binaire, la plus connue étant la précision globale. Mais il existe aussi la précision, le rappel, le F1-score etc.

La précision globale (ou *Accuracy*) mesure la proportion des prédictions correctes sur l'ensemble des échantillons.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

- $TP$  : *True Positives*, prédictions correctes de la classe positive.
- $TN$  : *True Negatives*, prédictions correctes de la classe négative.
- $FP$  : *False Positives*, prédictions incorrectes attribuées à la classe positive.
- $FN$  : *False Negatives*, échantillons positifs classés comme négatifs.

La précision correspond quant à elle à la proportion de vrais positifs parmi les prédictions positives ; et le rappel à la proportion de vrais positifs parmi les échantillons positifs. Le F1-score est la moyenne harmonique de ces deux dernières métriques.

## 2.4 Historique de modèles de détection de panneaux photovoltaïques sur images aériennes

Plusieurs modèles de détection de panneaux solaires à partir d'images aériennes ont été développés depuis quelques années, nous nous concentrerons ici sur les plus notables.

### 2.4.1 Avant le deep learning, forêts aléatoires

Les premiers modèles de détection de panneaux solaires, comme *Automatic detection of solar photovoltaic arrays in high resolution aerial imagery* (Malof et al. [2016]), se basent sur des méthodes traditionnelles de détection, comme les forêts aléatoires. Ce travail pionnier utilise des forêts aléatoires pour classifier les pixels sur la base de statistiques locales de couleur (moyennes et écarts-types en RGB) extraites à différentes distances sur des fenêtres 3x3. En sortie, une carte de probabilité est générée, affinée par un algorithme de détection d'objets qui regroupe les zones de haute confiance pour isoler les installations de panneaux photovoltaïques. Cette approche a introduit la première manière de collecte automatique d'informations PV depuis des images aériennes. Ce premier modèle a été entraîné sur environ 400 images.

### 2.4.2 Utilisation du deep learning

Les méthodes suivantes exploitent les CNN, dont le fonctionnement théorique est détaillé en 2.2, avec des architectures souvent similaires : un modèle pour classifier, un autre pour segmenter. C'est le cas de 3D-PV-Locator (Mayer et al. [2022]), qui repose sur Deeplabv3<sup>4</sup> pour la classification et Inception-v3 pour la segmentation. L'approche inclut du transfert d'apprentissage (voir 2.5) depuis des modèles génériques vers une tâche spécifique de détection de panneaux photovoltaïques, à l'aide d'un jeu de données déséquilibré, c'est à dire où les panneaux photovoltaïques sont rares (environ 35% d'images positives). Elle permet d'associer les polygones détectés à des coordonnées réelles et à des métadonnées 3D utiles à la planification énergétique (inclinaison, azimut<sup>5</sup>...). Enfin, d'autres modèles poursuivent cette approche classifieur-segmenteur, comme HyperSolarNet (Poonam Parhar [2022]), mais en utilisant d'autres modèles comme classifieur (UNet) et segmenteur (EfficientNet-B7). Enfin, GeoAI (Kausika et al. [2021]), quant à lui, n'utilise plus de classifieur, ce qui entraîne d'abord un grand nombre de faux positifs, qui sont ensuite corrigés par un post-traitement utilisant les métadonnées (notamment topographiques).

---

4. Un modèle originellement conçu pour l'imagerie médicale (Liu [2021]).

5. L'orientation du panneau photovoltaïque par rapport au Nord, mesurée en degré.

### 2.4.3 DeepSolar

Le modèle DeepSolar constitue un tournant dans cette série de travaux. Inspiré de 3D-PV-Locator (Mayer et al. [2022]), il bénéficie de jeux de données plus riches, collectés par une association de propriétaires d'installations. Contrairement aux modèles précédents, les données positives et négatives sont équilibrées. Les CNN sont réentraînés (*fine-tuning*) sur ces données, améliorant significativement la précision. Un filtrage géospatial post-inférence est utilisé pour éliminer les faux positifs identifiés en dehors des toits à l'aide de cartes topographiques. L'approche combine ainsi rigueur algorithmique et richesse des données, ce qui en fait un standard de référence.

Enfin, DeepSolarYu (Yu et al. [2018]), qui partage son nom avec le précédent mais suit une philosophie différente, supprime le segmenteur au profit d'une approche fondée sur les cartes d'activation de classe (CAM) extraites d'Inception-v3. Cela permet de combiner classification et localisation en une seule passe, avec une efficacité computationnelle accrue. Le jeu de données massif (plus de 360 000 images, déséquilibré à 12,6% d'images positives) est compensé par une fonction de perte pondérée et des méthodes d'augmentation (rotations aléatoires).

## 2.5 Apprentissage par transfert

L'apprentissage par transfert (*transfert learning*) consiste à utiliser un modèle déjà entraîné sur une tâche similaire à celle visée. Les paramètres de ce modèle préalablement entraîné servent alors à initialiser le nouveau modèle. Il est possible d'entraîner l'intégralité du modèle (on parle alors de *fine tuning*) ou de laisser certaines couches intactes et d'entraîner uniquement lac ou les dernières couches.

Cette méthode permet d'obtenir une situation initiale fonctionnelle, d'accélérer le processus d'entraînement complet et, enfin, d'obtenir de meilleurs résultats qu'avec un modèle entraîné de zéro (Torrey and Shavlik [2010]).

### 3 Un large éventail de données en amont de la construction du réseau de neurones convolutifs

#### 3.1 Contexte

Les données sont issues d'OpenStat Madagascar (OpenStat Madagascar [2025]) et sont composées de deux éléments : des images et un fichier Excel. Les images proviennent de satellites (relativement petites, c'est à dire de l'ordre de  $700 \times 400$  pixels) et de drones (haute définition, de l'ordre de  $4000 \times 3000$  pixels). OpenStat a sélectionné les images en observant à l'œil nu la présence de panneaux solaires sur les photos satellites, puis en vérifiant cette présence sur le terrain. Le tableau Excel, quant à lui, comprend 3 feuilles. La première contient les caractéristiques des images, la deuxième répertorie les différents éléments visibles sur les images (panneaux solaires ou ballons d'eau chaude), sachant que, dans certains cas, plusieurs panneaux sont considérés comme une seule entité, et la troisième définit la zone de l'image correspondant à chaque objet. Un tableau des variables ainsi que quelques images de la base de donnée sont disponibles en annexe B et A.

#### 3.2 Présentation des variables les plus utiles

Pour cette section veuillez vous référer à l'annexe B.

Dans la première feuille, chaque image est identifiée par `img_name`, qui correspond au numéro de l'image, avec un ajout potentiel d'ID si la photo a été prise par drone. Cette caractéristique peut être retrouvée dans `img_origin`. La variable `type1` indique les types d'éléments présents, et leurs quantités sont données par `elt_number`. Dans la deuxième feuille, chaque élément est identifié par `elt_name` et caractérisé par `type1` (*pan* ou *boil*), l'emplacement étant donné par `type2` (*roof*, *openspace*, *r\_openspace*, *window*). Le nombre de panneaux solaires présents par élément est indiqué par `pan_nbr` (un élément ne correspond pas forcément à un seul panneau solaire comme mentionné précédemment). Dans la troisième feuille, à chaque `elt_name` sont associés des points `edge_rank` pour déterminer son emplacement sur l'image.

#### 3.3 Description des images

Nous disposons ainsi de 10 000 images contenant des panneaux solaires (hors parcs solaires), correspondant à 8 454 géolocalisations différentes, certaines maisons étant photographiées plusieurs fois. Les images ont été prises en 2023, entre juillet et septembre. 9 000 d'entre elles proviennent de drones au format  $4000 \times 3000$  ou  $5280 \times 3956$ , et le reste (1000) provient de satellites avec des formats divers.

Les photos se concentrent le long des routes principales et des zones les plus peuplées, la capitale Antananarivo étant au centre. Au total, 130 500 panneaux solaires sont répertoriés, dont environ 60 000 ne font pas partie de centrales. Parmi l'ensemble complet des photos, 89% d'entre elles contiennent un panneau solaire.

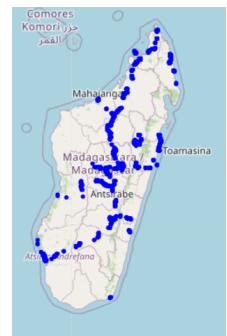


FIGURE 9 – Carte des positions des panneaux solaires.

## 4 Détection automatique par réseaux convolutifs : mise en œuvre et résultats

Afin de répondre à la problématique de détection posée, nous avons entraîné deux réseaux convolutifs profonds distincts. Le premier, un classifieur, a pour rôle de détecter la présence ou non de panneaux photovoltaïques dans une image donnée. Le second, un modèle de segmentation sémantique, permet de localiser finement les panneaux sur les images. L'idée étant – à terme – de combiner ces deux étapes pour tirer parti de la rapidité du classifieur et de la précision du segmenteur, tout en réduisant le nombre d'images à segmenter en donnant à ce dernier seulement les images détectées comme positives par le classifieur. Tous les entraînements ont été réalisés sur la plateforme SSP Cloud<sup>6</sup>, en exploitant une machine dotée de 15 Go de mémoire vidéo GPU, ce qui nous a permis d'entraîner nos modèles sans interruption, sur des lots de taille confortable (parfois pendant 10h consécutives). Le code complet est disponible en *open source* à l'adresse <https://github.com/maximechansat/StatApp>, avec des instructions détaillées pour garantir la reproductibilité des résultats, de la préparation des données à l'évaluation finale.

### 4.1 Prétraitement et enrichissement des données

Afin d'exploiter efficacement les images satellitaires et aériennes pour la détection de panneaux photovoltaïques, un ensemble de transformations a été appliqué aux données brutes. Ces opérations visent – d'une part – à homogénéiser les données d'entrée pour les rendre compatibles avec les architectures de réseaux de neurones utilisées et d'autre part à augmenter leur diversité apparente pour améliorer la robustesse du modèle. Il convient donc de distinguer les transformations ayant pour but de formater correctement les données et celles visant à accroître artificiellement leur variabilité (appelées augmentations).

Les images ont d'abord été redimensionnées en  $299 \times 299$  pixels, conformément aux exigences d'entrée des CNN pré-entraînés mobilisés pour nos tâches de classification et de segmentation (sur lesquels nous reviendront en 4.2). Les images subissent ensuite des rognages aléatoires pour en extraire une sous-image (ou *crops*), et la même modification a été appliquée au masque. Un seuil de 0,05% de surface occupée par des panneaux a été fixé : si un *crop* contient moins de 0,05% de pixels correspondant à un panneau, il est considéré comme négatif pour le modèle de classification. Ce choix heuristique permet d'éviter d'étiqueter comme positives des images contenant une part négligeable de panneau.

Pour enrichir le jeu de données et limiter les risques de surapprentissage, nous avons utilisé plusieurs techniques d'augmentation : rotations et symétries aléatoires ainsi qu'ajout de bruit gaussien. Ces augmentations se concluent par la normalisation des canaux. Cette normalisation permet de réduire la variabilité entre images, notamment liée aux conditions d'éclairage ou à l'acquisition, tout en évitant des distorsions extrêmes qui nuiraient à la convergence du modèle. Notons que ces augmentations sont appliquées uniquement sur le jeu de données d'entraînement, pas sur celui de test, qui censé être une mesure

---

6. Le SSP Cloud (pour Système Statistique Public) est un outil mis à disposition par l'INSEE pour favoriser les expérimentations des statisticiens publics depuis 2020 (INSEE [2023])

fiable et donc constant des performances du modèle.

## 4.2 Architectures et choix d'implémentation

Le modèle de classification repose sur l'architecture InceptionV3 (comprenant 2,4 millions de paramètres), dont la tête a été modifiée pour produire une sortie binaire via une fonction d'activation sigmoïde. Le modèle de segmentation, quant à lui, utilise DeeplabV3 (à 6 millions de paramètres), avec des convolutions dilatées (espacées) : l'ajout d'un paramètre  $r$  permet d'agrandir le champ réceptif sans perte de résolution, crucial pour identifier les grandes installations peu compactes.

Les deux modèles sont entièrement réentraînés, mais initialisés avec les poids de DeepPVMapper (Kasmi et al. [2022]) pour accélérer la convergence (on parle d'apprentissage par transfert, comme détaillé en 2.5). Nous réentraînons complètement les modèles dans les deux cas, mais en employant un taux d'apprentissage faible afin d'ajuster finement les paramètres déjà entraînés.

## 4.3 Résultats expérimentaux et choix d'évaluation

### 4.3.1 Évolution des performances du segmenteur : comparaison de deux modèles et discussion critique

Nous avons, au cours du projet et des expérimentations successives, entraîné à plusieurs reprises notre modèle de segmentation, en faisant évoluer certains choix comme la fonction de perte ou les paramètres d'optimisation. Ces évolutions s'inscrivent dans une dynamique double : d'une part, l'intégration progressive de pratiques que nous aurions pu anticiper dès le début du projet, et d'autre part, des ajustements motivés par l'analyse des performances intermédiaires, nous incitant à explorer de nouvelles pistes pour surmonter certaines limites observées. En somme, ces expérimentations nous permettent de comparer différentes versions du modèle et d'analyser leur impact sur les performances obtenues.

Le premier modèle de segmentation que nous avons entraîné est basé sur la fonction de perte BCE (cf. 2.3.1) et une fonction d'activation linéaire (définie en 2.1.1). Les résultats sont nuancés. D'un côté, il présente une dynamique d'apprentissage classique. La perte décroît rapidement de manière exponentielle au cours des premières itérations, atteignant un minimum autour de la 18ème *epoch* (0,049), avant de légèrement remonter, signe d'un début de surapprentissage, comme présenté en Figure 10. En revanche, l'évolution de l'indice de Jaccard sur l'échantillon de test est plus instable : bien qu'un pic soit atteint dès la 2ème *epoch*, les valeurs suivantes oscillent fortement sans dégager de tendance claire (Figure 11). À l'*epoch* correspondant au minimum de la BCE, l'indice de Jaccard s'établit à 0,46, mais il reste volatil.

Le second entraînement du modèle de segmentation tire parti des enseignements du premier. Nous limitons cette fois le nombre d'*epochs* à une quinzaine, ayant constaté précédemment que la perte atteignait rapidement un plateau. Deux modifications majeures sont introduites : l'utilisation de la distance de Jaccard comme fonction de perte – afin d'aligner l'optimisation sur la métrique réellement utilisée pour l'évaluation – et

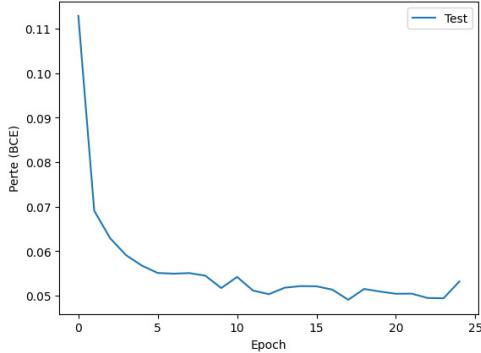


FIGURE 10 – Évolution de la perte sur le *dataset* de test du premier modèle.

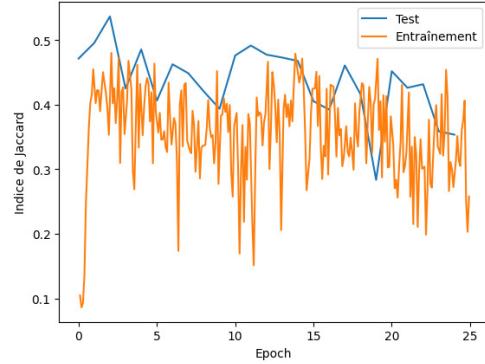


FIGURE 11 – Évolution de l'indice de Jaccard du premier modèle.

l'utilisation d'une fonction d'activation sigmoïde. L'ajout d'un bruit gaussien dans les augmentations de données fait également son apparition, élément que nous n'avions pas anticipé initialement mais qui rend le modèle plus robuste. Cette nouvelle configuration se révèle nettement plus efficace : la perte diminue à nouveau rapidement et de manière stable (Figure 20), tandis que l'indice de Jaccard progresse régulièrement tout au long de l'entraînement, atteignant 0,659 à la dernière *epoch* sans connaître les oscillations marquées observées précédemment (Figure 21). Le modèle paraît ainsi mieux guidé par sa fonction objectif et bénéficie d'un signal plus cohérent au regard de la tâche de segmentation.

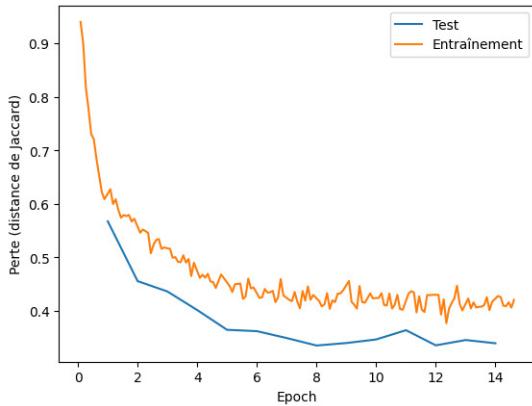


FIGURE 12 – Évolution de la perte (distance de Jaccard) du deuxième modèle.

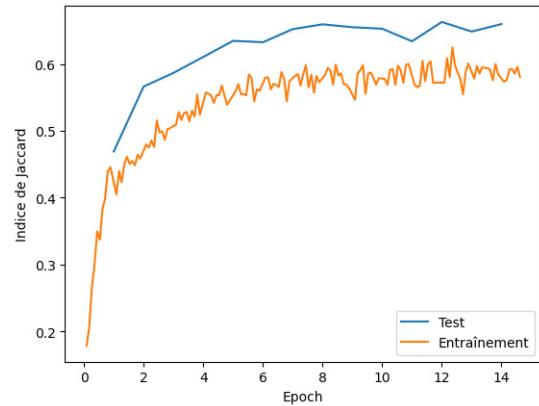


FIGURE 13 – Évolution de l'indice de Jaccard du deuxième modèle.

Cette amélioration peut néanmoins être nuancée : en utilisant une fonction de perte très proche de la métrique elle-même, on tend artificiellement à optimiser celle-ci, ce qui peut biaiser l'évaluation comparative. De plus, nous ne sommes pas en capacité d'isoler l'effet d'un des changements parmi les trois opérés. Il faudrait, pour ce faire, réaliser une multitude d'entraînements *ceteris paribus*, ce que nos contraintes computationnelles ne permettent pas.

### 4.3.2 Évaluation du modèle de classification : stabilité, optimisation du seuil et pouvoir discriminant

Le modèle de classification a été entraîné sur 25 *epochs* à l'aide d'une fonction de perte de type BCE. La perte décroît de manière rapide et régulière lors des premières itérations, traduisant une convergence efficace, avant d'atteindre un minimum à la 24ème *epoch*. Notons néanmoins que la perte observée dès la 9ème *epoch* est très proche de cette valeur minimale et est associée à des performances globalement similaires sur l'ensemble de test. Cela suggère que le modèle a appris l'essentiel des régularités dès les premières itérations, et que les écarts ultérieurs reflètent surtout des fluctuations liées aux effets aléatoires des augmentations de données, plutôt qu'un véritable gain de performance.

En ce qui concerne les performances prédictives du modèle, les métriques classiques de classification binaire sont excellentes. À seuil de décision<sup>7</sup> fixé – arbitrairement – à 0,5, on obtient une précision de 0,961, un rappel de 0,948, une précision globale (*accuracy*) de 0,926 et un F1-score de 0,955 (cf. 2.3.4 pour les définitions de ces métriques). Ce profil de résultats indique une excellente capacité du modèle à détecter les images positives tout en limitant les faux positifs. Afin d'évaluer la robustesse de cette prédiction et d'optimiser davantage le compromis entre précision et rappel, nous avons cherché à maximiser directement le F1-score en faisant varier le seuil de décision. Cette procédure conduit à un seuil optimal de 0,58, pour lequel précision et rappel sont parfaitement équilibrés à 0,959 et 0,958 respectivement, et le F1-score atteint un maximum de 0,958. En Annexe C, sont présentées plusieurs représentations graphiques des métriques (avec un seuil de décision initial de 0,5), ainsi qu'un graphique montrant l'évolution de la *loss* et un autre illustrant la détermination du seuil de décision optimal.

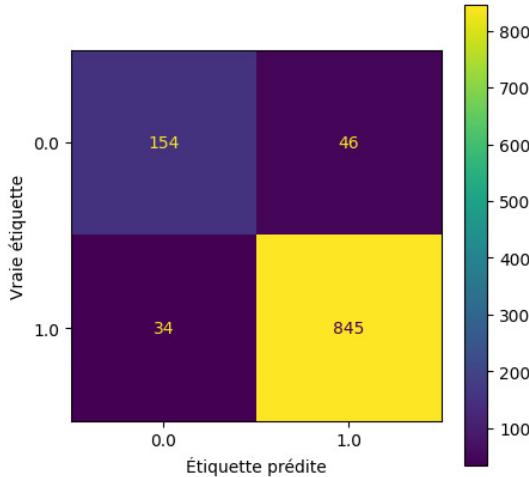


FIGURE 14 – Matrice de confusion du modèle.

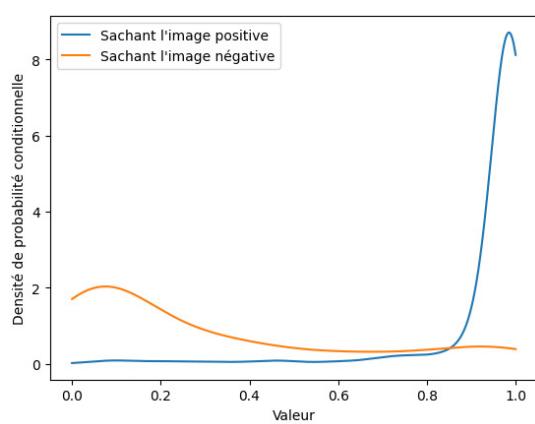


FIGURE 15 – Densité de probabilité conditionnelle par classe.

La pertinence du modèle est également mise en évidence par l'analyse qualitative de sa matrice de confusion (Figure 14). Celle-ci montre un nombre – et une proportion – très faibles de faux positifs et de faux négatifs, ce qui renforce la confiance dans les prédictions produites. Enfin, l'étude des distributions conditionnelles des probabilités de sortie du modèle fournit un éclairage complémentaire. La Figure 15 représentant la densité des scores

7. Il s'agit de la valeur à partir de laquelle une probabilité prédictive est convertie en classe binaire : on classe comme positive toute sortie supérieure à ce seuil.

conditionnellement à la classe réelle montre deux courbes nettes, bien séparées, associées respectivement aux étiquettes positives et négatives. Ce bon pouvoir discriminant suggère que le classifieur a appris des représentations suffisamment distinctives pour opérer une séparation efficace entre les deux classes, ce qui valide empiriquement la pertinence de l'architecture et du protocole d'entraînement retenus.

#### 4.3.3 Le pipeline complet

La combinaison des deux modèles – classifieur puis segmenteur – permet de tirer parti des forces respectives de chacun : rapidité de traitement pour le premier, précision spatiale pour le second. On ne soumet ainsi que les images détectées comme positives au segmenteur. Ce pipeline atteint un indice de Jaccard global de 0,58.

Afin d'obtenir une métrique plus "visuelle", nous avons également calculé le taux d'erreur relatif dans le dénombrement des panneaux après un traitement morphologique minimal des images (nous avons appliqué, à cet effet, une ouverture puis un décompte des composantes connectées). Cette métrique est définie comme suit :

$$\text{erreur relative} = \frac{\text{nombre prédit} - \text{nombre réel}}{\text{nombre réel}}$$

L'erreur relative de dénombrement calculée est de 9,6%, ce qui suggère que notre modèle sur-estime le nombre d'installations solaires installées. Pour autant, un meilleur traitement morphologique permettrait probablement d'améliorer ce chiffre, qui mélange deux effets : la qualité de prédiction de notre modèle et la qualité du traitement permettant le comptage des panneaux.

Enfin, une évaluation basée sur l'erreur relative de superficie prédite des panneaux solaires constituerait un indicateur pertinent. Cependant, nos images sont prises à des angles et altitudes très variables (et non renseignés dans la base de donnée en ce qui concerne l'inclinaison des panneaux), rendant cette estimation actuellement impraticable. Elle représente toutefois une piste de recherche et d'amélioration importante.

## Conclusion

Dans le cadre de ce projet, nous avons développé une approche complète de détection d'installations photovoltaïques à partir d'images aériennes, en réponse à une problématique croissante pour les gestionnaires de réseau. Pour allier efficacité et précision, nous avons choisi d'articuler deux modèles convolutifs complémentaires : un classifieur binaire, chargé d'identifier les images susceptibles de contenir des panneaux, suivi d'un modèle de segmentation sémantique, capable de les localiser à l'échelle du pixel.

Ce choix s'est d'abord appuyé sur une revue approfondie des méthodes existantes, mais aussi sur les contraintes concrètes du problème : données bruitées, angles de vue variables, et limites computationnelles. Nous avons ainsi progressivement affiné nos choix de prétraitements, d'architectures, de fonctions de perte ou d'activation. En particulier, l'entraînement de deux modèles de segmentation différents nous a permis d'identifier plusieurs leviers d'amélioration, comme l'ajout de bruit gaussien ou l'usage d'une fonction de perte plus adaptée.

L'approche combinée fait preuve de résultats encourageants, atteignant un indice de Jaccard global de 0,58, et une erreur relative de 9,6% dans le dénombrement des installations.

Plusieurs pistes restent ouvertes : amélioration du segmenteur à l'aide de données 3D, ce qui permettrait une estimation en surface et donc en puissance installée (à la manière de Mayer et al. [2022]), intégration de données cadastrales ou encore de données socio-économiques. Ces prolongements pourraient enrichir considérablement notre approche et contribuer à faire de ce type d'outil une brique robuste pour le suivi et la planification du solaire diffus, notamment dans des contextes où les données officielles sont limitées.

## Références

- Agence internationale de l'énergie. <https://www.iea.org/data-and-statistics/data-tools/energy-statistics-data-browser?country=MADAGASCAR&fuel=Energy%20supply&indicator=ElecGenByFuelLC>. Consulté le 9 mai 2025.
- G. E. H. Alex Krizhevsky, Ilya Sutskever. Imagenet classification with deep convolutional neural networks. *research highlights*, 60, 2012. URL <https://proceedings.neurips.cc/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf>.
- M. Z. Alom, T. M. Taha, C. Yakopcic, S. Westberg, P. Sidike, M. S. Nasrin, B. C. Van Esen, A. A. S. Awwal, and V. K. Asari. The history began from alexnet : A comprehensive survey on deep learning approaches. *arXiv preprint arXiv :1803.01164*, 2018.
- A. Bodin and F. Recher. *Deepmath : mathématiques des réseaux de neurones*, chapter 5, pages 69–98. Indépendant, 2021. ISBN 979-8692672872.
- Cnam. Cours - réseaux de neurones convolutifs — cnam – ue rcp209, n.d. Consulté sur <https://cedric.cnam.fr/vertigo/cours/m12/coursDeep3.html>.
- GBM. Groupe de la Banque mondiale. Accès à l'électricité, zones rurales/urbaines (% de la population) - Madagascar, Sub-Saharan Africa. <https://donnees.banquemonde.org/indicateur/EG.ELC.ACCTS.RU.ZS?locations=MG-ZG>, 2023.
- INSEE. SSP Cloud – Une plateforme pour la science ouverte et le calcul intensif, 2023. URL <https://www.insee.fr/fr/statistiques/6035940?sommaire=6035950&q=sspcloud>. Consulté le 8 mai 2025.
- IRENA. Off-grid renewable energy capacity statistics 2025, March 2025. URL [https://www.irena.org/-/media/Files/IRENA/Agency/Publication/2025/Mar/IRENA\\_DAT\\_RE\\_Capacity\\_Statistics\\_2025.pdf](https://www.irena.org/-/media/Files/IRENA/Agency/Publication/2025/Mar/IRENA_DAT_RE_Capacity_Statistics_2025.pdf). Accessed 2025-04-30.
- G. Kasmi, L. Dubus, P. Blanc, and Y.-M. Saint-Drenan. Deepolar tracker : towards unsupervised assessment with open-source data of the accuracy of deep learning-based distributed pv mapping. *arXiv preprint arXiv :2207.07466*, 2022.
- B. B. Kausika, D. Nijmeijer, I. Reimerink, P. Brouwer, and V. Liem. Geoai for detection of solar photovoltaic installations in the netherlands. *Energy and AI*, 6 :100111, 2021. ISSN 2666-5468. doi : <https://doi.org/10.1016/j.egyai.2021.100111>. URL <https://www.sciencedirect.com/science/article/pii/S2666546821000604>.
- F. Kotthoff, T. Welzel, S. Sander, and T. Degner. Monitoring Germany's Core Energy System Dataset : A Data Quality Analysis of the Marktstammdatenregister, 2023. URL <https://arxiv.org/pdf/2304.10581.pdf>. arXiv preprint.
- L. L. S. Z. Y. Liu, X. ; Song. A review of deep-learning-based medical image segmentation methods. *Sustainability*, 2021.
- V. Luminis. Tout ce dont vous devez savoir sur le modèle du perceptron. *Medium*, jun 2023. URL <https://medium.com/@VeritasLuminis/tout-ce-dont-vous-devez-savoir-sur-le-mod%C3%A8le-du-perceptron-74da4df52712>.

- J. M. Malof, K. Bradbury, L. M. Collins, and R. G. Newell. Automatic detection of solar photovoltaic arrays in high resolution aerial imagery. *Applied Energy*, 183 :229–240, 2016. ISSN 0306-2619. doi : <https://doi.org/10.1016/j.apenergy.2016.08.191>. URL <https://www.sciencedirect.com/science/article/pii/S0306261916313009>.
- K. Mayer, B. Rausch, M.-L. Arlt, G. Gust, Z. Wang, D. Neumann, and R. Rajagopal. 3d-pv-locator : Large-scale detection of rooftop-mounted photovoltaic systems in 3d. *Applied Energy*, 310 :118469, 2022. ISSN 0306-2619. doi : <https://doi.org/10.1016/j.apenergy.2021.118469>. URL <https://www.sciencedirect.com/science/article/pii/S0306261921016937>.
- OpenStat Madagascar. Données sur l'énergie solaire et labellisation d'images de panneaux photovoltaïques à madagascar, 2025. URL <https://openstat-madagascar.com/bdd/energie-et-environnement/131-donnees-sur-l-energie-solaire-et-labellisation-d-images-de-panneaux-photovolta>
- A. T. C. R. H. V. N. N. F. V. Poonam Parhar, Ryan Sawasaki. Hyperionsolarnet solar panel detection from aerial images. 2022. URL <https://arxiv.org/abs/2201.02107>.
- S. J. Prince. *Understanding Deep Learning*. MIT Press, July 2023. Chapitre 10.
- RFI. Madagascar : une affaire de détournements de fonds secoue la compagnie nationale d'électricité, 2022. URL <https://www.rfi.fr/fr/afrique/20221201-madagascar-une-affaire-de-d%C3%A9tournements-de-fonds-secoue-la-compagnie-nationale-d-%C3%A9lectricit%C3%A9>. Consulté le 1 mai 2025.
- RTE. Equilibrer l'offre et la demande d'électricité. <https://www.rte-france.com/chaque-seconde-courant-passe/equilibrer-loffre-et-la-demande-delectricite>.
- M. Semeurand. Présentation des réseaux de neurones artificiels. *SI-NAT BLOG*, oct 2021. URL <https://blog.sinatechnologie.com/presentation-des-reseaux-de-neurones-artificiels>.
- S. R. Shakeel, H. Yousaf, M. Irfan, and A. Rajala. Solar PV adoption at household level : Insights based on a systematic literature review. *Energy Strategy Reviews*, 50, 2023. doi : 10.1016/j.esr.2023.101178. URL <https://doi.org/10.1016/j.esr.2023.101178>.
- L. Torrey and J. Shavlik. Transfer learning. In *Handbook of Research on Machine Learning Applications and Trends : Algorithms, Methods, and Techniques*. IGI Global, 2010. doi : 10.4018/978-1-60566-766-9.ch011. URL <https://doi.org/10.4018/978-1-60566-766-9.ch011>.
- World Bank. Digital and Energy Connectivity for Inclusion in Madagascar (DECIM), 2023. URL <https://documents1.worldbank.org/curated/en/099220002162314434/pdf/P178701085338e04097500b11a2be225df.pdf>. Accessed : 2025-05-01.
- J. Yu, Z. Wang, A. Majumdar, and R. Rajagopal. Deepsolar : A machine learning framework to efficiently construct a solar deployment database in the united states. *Joule*, 2(12) :2605–2617, 2018. ISSN 2542-4351. doi : <https://doi.org/10.1016/j.joule.2018.11.021>. URL <https://www.sciencedirect.com/science/article/pii/S2542435118305701>.

L. Zhao and Z. Zhang. A improved pooling method for convolutional neural networks. *Scientific Reports*, 14(1) :1072, January 2024. doi : 10.1038/s41598-024-51258-6. Méthode T-Max-Avg.

## Annexes

### A Exemple d'images de drones



FIGURE 16 – Exemple d'image de drone.



FIGURE 17 – Exemple d'image de drone.

## B Architecture du Excel

LEV_1	
number	Numéro de l'image
img_name	Nomenclature des images satellites et drones (ajout de ID)
img_long_East	Longitude du Point Centroïde des images drone, tous sont à l'Est
img_lat_South	Latitude du Point Centroïde des images drone, tous sont au Sud
img_origin	Deux valeurs possibles : « S » pour Satellite, « D » pour Drone
type1	Types de matériels d'énergie solaire (Boil, Pan, Mix, Solar_park)
elt_number	Nombre d'éléments identifiés dans l'image (ex : 1, 2, 3)
type2	Emplacement des éléments dans l'image (roof, openspace, r_openspace, Window)
Park_capacity_kw	Capacité de production énergétique (si type1 = solar-park)
Cam_alt	Altitude de la caméra satellite (en mètre, Google Earth)
Mouse_alt	Altitude de la souris satellite (en mètre, Google Earth)
width_pixel	Largeur en pixel des images (longitude)
height_pixel	Hauteur en pixel des images (latitude)
city	Ville ou Village
region	Région géographique (Center, North, East, West, South)
img_date	Date de prise de vue
img_time	Heure de prise de vue (uniquement pour les drones)
img_altsea	Altitude des images par rapport à la mer (uniquement pour les drones)
zoom	Zoom de la caméra (uniquement pour les drones)
img_ISO	ISO de la caméra (uniquement pour les drones)
LEV_2	
img_name	Nom de l'image (identique à Lev_1)
elt_name	Nom de l'élément identifié
type1	Boil ou Pan (exclusivement, pas de Mix)
type2	Emplacement des éléments (roof, openspace, r_openspace, Window)
boil_nbr	Nombre de chauffe-eaux de chaque élément (si type1 = Boil)
pan_nbr	Nombre de panneaux de chaque élément (si type1 = Pan)
pan_area_sqm	Superficie des panneaux (uniquement pour les images satellites)
notes	Notes supplémentaires sur l'image (ex : panneaux sur lampadaire)
LEV_3	
elt_name	Mêmes valeurs que dans LEV_2
edge_rank	Numéros des points reliant les contours des panneaux. Le minimum est de 4 points pour chaque élément.
long	Numéro de pixel sur la largeur de l'image (longitude). Traitement d'image avec GIMP.
lat	Numéro de pixel sur la hauteur de l'image (latitude). Traitement d'image avec GIMP.

TABLE 1 – Tableau des variables

## C Diverses représentations graphiques des performances du modèle de classification

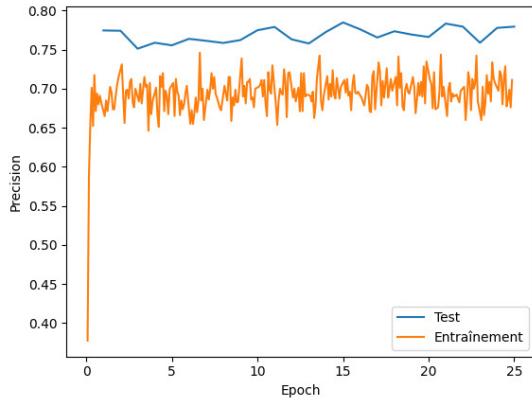


FIGURE 18 – Évolution de la métrique de précision.

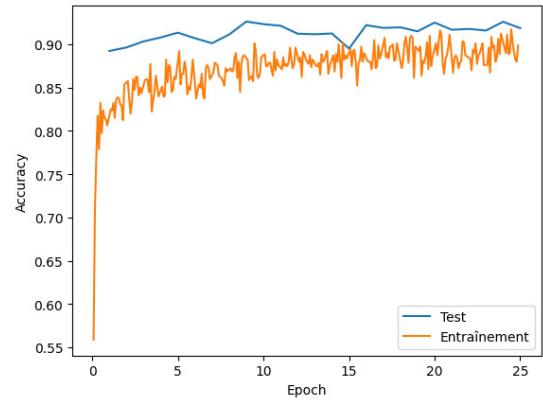


FIGURE 19 – Évolution de la métrique de précision globale.

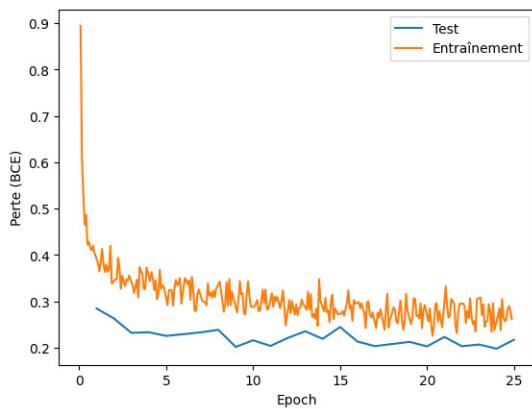


FIGURE 20 – Évolution de la perte (BCE).

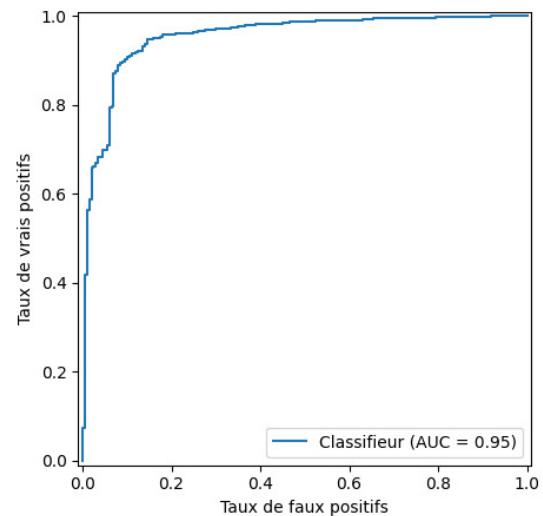


FIGURE 21 – Répartition du taux de vrais positifs (précision) et du taux de faux positifs, paramétrisé en fonction de la valeur du seuil de décision ( $s = 0$  en bas à gauche,  $s = 1$  en haut à droite et  $s^* = 0,58$  optimal en haut à gauche).