

Projet Python - S2 2024 - GAN Images

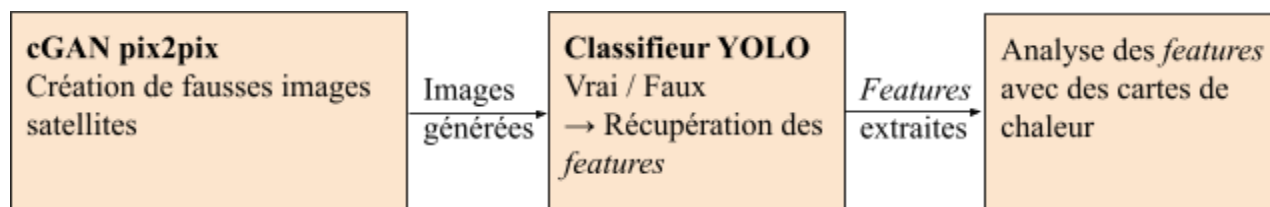
Groupe 1

Lise Bernard - Charlotte Berthier - Axelle Salvador - Albina Toumarkine

But du projet

Notre projet portait sur la génération d'images, ainsi que sur la classification des images. Il questionnait les éléments sur lesquels se concentre un réseau de neurones pour réaliser une classification d'images.

Dans ce cadre, nous avons mis à l'épreuve un réseau de neurones YOLO en lui soumettant de vraies images satellites et de fausses images satellites, générées avec le réseau antagoniste génératif sous contrainte (cGAN) Pix2Pix. Nous avons ensuite extrait et analysé les *features* du réseau, pour comprendre ce sur quoi la convolution s'appuie pour distinguer le vrai du faux.



Présentation et préparation des données

Pour récupérer nos vraies images satellites et créer nos fausses images satellites, nous avons travaillé à partir de deux jeux de données mis à disposition du public par l'Institut national de l'information géographique (IGN).

Nos vraies images satellites sont les images satellites du département du Gers (32), millésime 2019. Elles sont extraites de la base de données “WMS_ORTHOHISTO_Photos_aériennes_2019” de l’IGN, disponible sur la Géoplateforme¹. Ces données sont des images raster, géocodées en Lambert93 (le référentiel spatial règlementaire pour la France). Ces images sont d’une très grande résolution.

Pour créer nos fausses images satellites, nous avons récupéré les masques d’occupation des sols correspondants à nos images satellites source. Pour cela, nous avons récupéré les données officielles d’occupation des sols du Gers en 2019, également mises à disposition sur la géoplateforme de l’IGN, dans la base de données “OCSGE_WTMS_RASTER”². Ces données correspondent au référentiel national d’Occupation des Sols à Grande Echelle (OCS GE)³ de l’IGN. C’est une base de données vectorielle décrivant l’occupation des sols sur l’ensemble du territoire national. Elle est géoréférencée dans le même système de projection géographique que nos images satellites : Lambert93. Les données d’occupation des sols sont produites chaque année par l’IGN à partir des images satellites officielles d’un territoire. Par conséquent, nos images satellites et nos masques d’occupation des sols se correspondent, et se superposent, exactement.

La base OCS GE sur l’occupation des sols distingue la couverture spatiale et l’usage des sols. Puisque nous cherchions à générer de fausses images satellites, nous avons récupéré la couche de couverture des sols. Cela décrit la physionomie d’un territoire et les éléments structurants du paysage.

Nous avons téléchargé les images satellites et les masques d’occupation des sols en connectant le logiciel QGIS aux API de l’IGN. Nous avons découpé une bounding box autour du Gers, pour extraire uniquement les images satellite du département. Nous

¹ <https://geoservices.ign.fr/services-web-experts-orthohisto> (visité le 10 juin 2024)

² <https://geoservices.ign.fr/services-web-experts-ocsge> (visité le 10 juin 2024)

³ <https://geoservices.ign.fr/ocsge> (visité le 10 juin 2024)

avons fait de même pour la couche vecteur d'occupation des sols. Nous avons ensuite transformé cette couche vectorielle en raster, en appliquant un algorithme de rasterisation pré-implémenté dans QGIS. Nous avons ainsi nos images satellites et nos masques d'occupation des sols exactement superposés. Nous avons exporté ces très grandes images au format TIF.

À partir des images tif extraites de l'IGN, nous avons effectué un pré-traitement pour récupérer des images au format 256 par 256 avec les masques associés. Un pixel des images tif correspond à 10 mètres de distance, donc les images faisant 9 077 pixels par 12 394 pixels couvrent les 90 km par 123 km du Gers. Lorsque l'on découpe les images en carrés de 256 par 256 pixels, on obtient une grille de 35 cases par 48 soit un dataset d'environ 1 680 images satellites et 1 680 masques (voir le script `projet_python.ipynb`).

	Dimensions des images (en pixels)	Dimensions requis par Pix2Pix	Calcul de la grille
Hauteur	9 077	256	35,4
Largeur	12 394	256	48,4

Après extraction, on obtient donc 3 528 images. Parmi celles-ci, certaines ne sont pas au bon format (bords des images source) et d'autres contiennent uniquement des pixels noirs, liés aux limites territoriales du Gers qui n'est pas un département de forme parfaitement rectangulaire. Nous avons donc décidé de supprimer toutes les images détenant moins de 90% de pixels noirs. Il nous reste alors 2 114 images.

Nous avons d'abord essayé de fournir les images et les masques de manière séparée au réseau mais nous avons une erreur au moment de la conversion en tenseurs. Pour résoudre ce problème, à partir de ces images découpées, nous avons créé un dataset similaire à celui utilisé dans l'exemple proposé par Pix2Pix⁴. Pour cela,

⁴ Nous sommes parties du modèle Pix2Pix suivant : <https://www.tensorflow.org/tutorials/generative/pix2pix?hl=fr>

nous créons des images composées de la juxtaposition des images d'origine et de leur masque.



Exemple d'une image découpée et de son masque juxtaposé

Méthodes utilisées

Le réseau génératif (GAN) pix2pix

Un réseau antagoniste génératif (*Generative Adversarial Network*, GAN) est un modèle d'intelligence artificielle utilisé pour générer de nouvelles données à partir d'un ensemble de données d'entraînement. Dans le cas de pix2pix, il s'agit d'un cGAN (*conditional Generative Adversarial Network*) : le modèle génère de nouvelles images à partir des masques des images d'origine. Nous fournissons donc au GAN des images de 256 par 512, qu'il sépare en deux images, d'un côté le masque et de l'autre l'image satellite originale.

Un GAN est composé d'un générateur et d'un discriminateur. Le générateur crée de nouvelles données. Le discriminateur détermine si chaque *patch* d'images en sortie est vrai ou non, puis entraîne le générateur en retour. Le générateur de pix2pix repose sur une architecture U-Net, composé d'un encodeur (couche convolutive et Leaky ReLU) et d'un décodeur (couche de convolution transformée et ReLU).

Le discriminateur de pix2pix repose lui sur un classifieur convolutionnel PatchGAN. Il est composé de couches de convolution et d'un *Leaky ReLU*. Il reçoit en entrée soit le masque et l'image créés par le générateur qu'il doit classifier comme fausses, soit le masque et l'image originaux qu'il doit classifier comme vrais. Au cours de l'entraînement, les deux parties du réseau sont optimisées par la descente de gradient stochastique Adam.

L'entraînement final utilisé a été réalisé sur Google Colab, en 30 000 *steps* (après une tentative à 40 000 *steps* qui a saturé le GPU, et une à 10 000 *steps* qui donnaient des images de mauvaises qualité). Nous avons ensuite généré 212 fausses images satellites grâce à notre modèle entraîné.

YOLO

Les modèles YOLO sont des modèles de détection d'objets et de segmentation d'images en temps réel⁵, qui reposent sur des réseaux neuronaux convolutifs. Nous utilisons dans le cadre de ce projet YOLOv8, la dernière version du modèle développée par Ultralytics, qui en propose des variantes destinées à cinq tâches différentes : la détection, la segmentation, l'estimation de pose, le suivi et la classification. C'est le plus petit des modèles de classification (yolov8n-cls) que nous utilisons pour entraîner un modèle capable de distinguer les vraies images satellites de celles générées par le GAN.

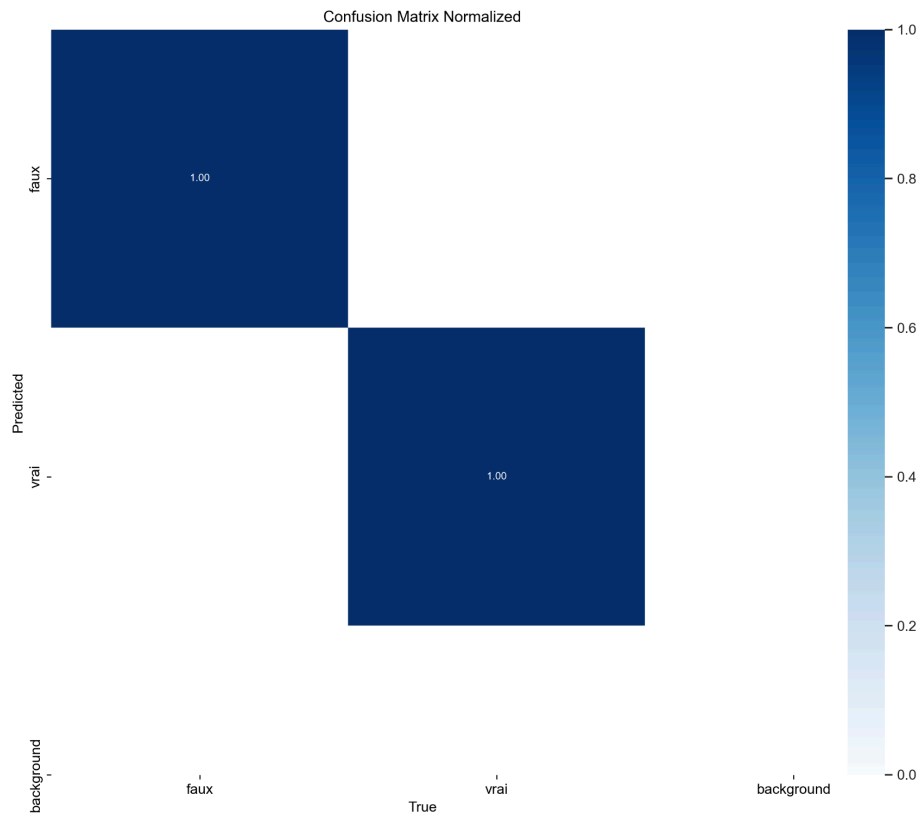
Pour entraîner YOLO, nous avons pris un total de 424 images, 212 fausses images satellites et 212 vraies images satellites. Nous les avons réparties en deux ensembles de données : les données d'entraînement (80%, soit 338 images) et les données de validation (20%, soit 86 images). Nous avons équilibré les datasets, de telle sorte à avoir autant d'images vraies que de fausses dans chaque ensemble. Nous avons donc entraîné YOLO sur 338 images (169 vraies et 169 fausses). L'entraînement est réalisé en 10 époques avec des *batches* de 32 images.

⁵ Nous avons travaillé à partir du modèle YOLO v8 : <https://docs.ultralytics.com/>

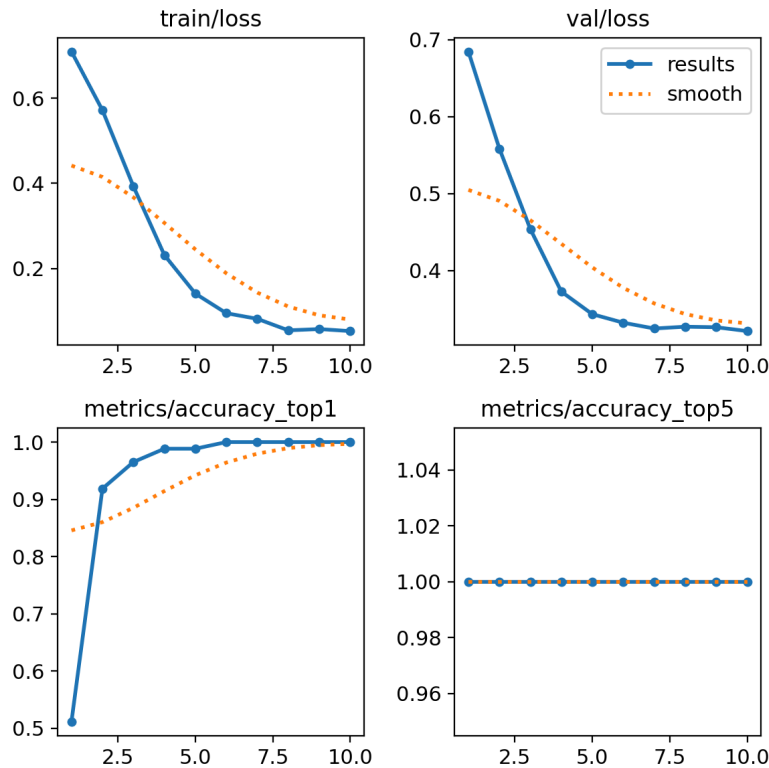
Résultats

Analyse des performances de l'entraînement YOLO

Le classifieur YOLO distingue systématiquement les images générées des images originales, comme le montre la matrice de confusion suivante, qui indique que la totalité des images de validation ont été classifiées dans la bonne classe (vrai ou faux). Dès la septième époque, le réseau atteint 100% de précision.



Matrice de confusion pour la validation du classifieur YOLO



Courbes d'apprentissage et précision lors de l'apprentissage

Extraction et analyse des *features* par cartes de chaleur

L'extraction des features avec la fonction predict de YOLO n'est pas concluante. Les données sont illisibles. Nous avons donc voulu générer des cartes de chaleur pour observer les pixels déterminants dans la classification YOLO. Nous appliquons une stratégie d'occlusions successives de nos images, c'est-à-dire que nous supprimons certains pixels au moment de la classification pour voir s'ils sont déterminants ou non dans celle-ci.

Quand elle est réalisée au niveau du pixel, l'occlusion est extrêmement chronophage. Chaque pixel prend environ 15 millisecondes. Or, les images à partir desquelles nous travaillons sont composée de 65 536 pixels : il faut environ 16 minutes pour obtenir le résultat d'une seule occlusion. Trop lourde, cette méthode a systématiquement tué les noyaux de nos notebooks. Nous avons donc décidé d'augmenter le pas de l'occlusion, en supprimant des zones de 5 pixels par 5 pixels sur

les images. Les zones seront évidemment moins précises mais cela nous permet d'obtenir des résultats plus rapidement avec de premières pistes de compréhension de la classification YOLO.

Pour chaque carré de 5 par 5 pixels, l'algorithme donne le degré de confiance dans la classe attribuée. L'occlusion sert à supprimer certains pixels temporairement. Si la classification des pixels alentour change avec l'occlusion, le pourcentage normalisé de confiance en la classification diminue. Les zones rouges correspondent aux zones pour lesquelles le modèle YOLO est très sûr de sa classification avec passage des occlusions d'images. Plus la zone est bleue, plus la probabilité de classification a diminué pendant le traitement.

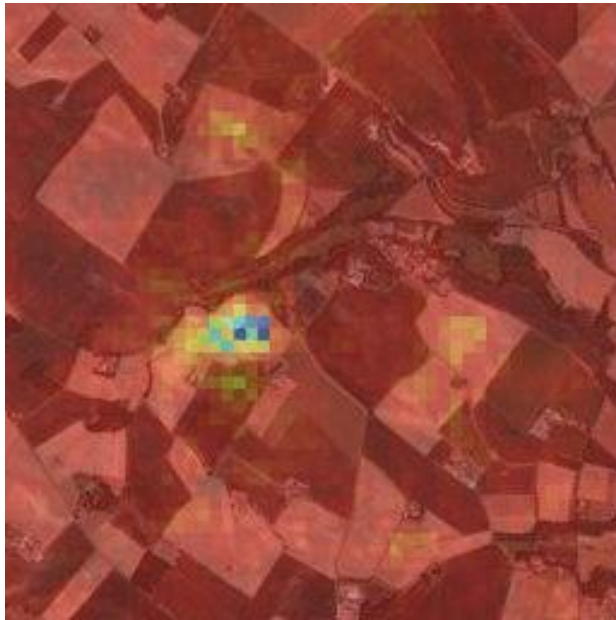
Pour réaliser nos cartes de chaleur, le modèle regarde le rang de la classe parmi les 2 classes les plus probables au moment de la classification de l'image. Dans notre cas, la classe de référence est 0, soit "image vraie". Les pixels rouges dans l'ensemble des cartes de chaleur sont ceux qui sont les plus discriminants dans l'image pour indiquer qu'elle est vraie. Inversement, les pixels tirant vers le bleu sont ceux qui distinguent les images fausses des vraies.

Par conséquent, dans les cartes de chaleur, les fausses images satellite apparaissent comme des négatifs des vraies images, puisque la même échelle de couleurs est utilisée pour générer les cartes de chaleur.

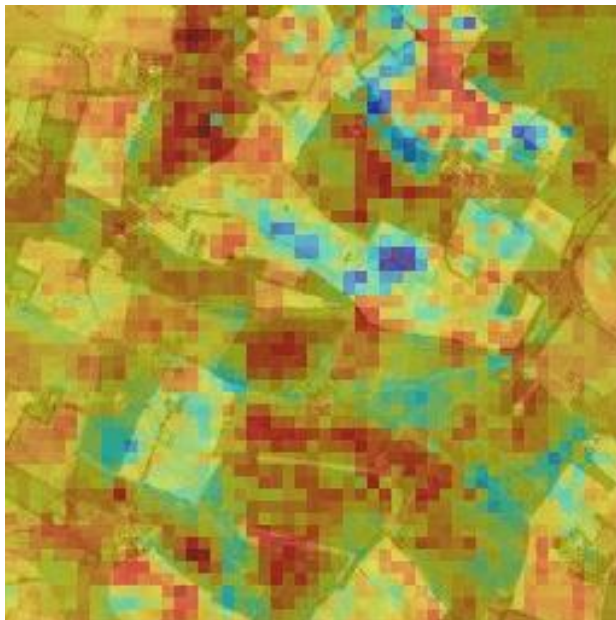
Vraies images satellites

Les cartes de chaleur nous indiquent que le modèle YOLO a une lecture plutôt globale des vraies images satellites. Les zones de confiance moindres correspondent à des zones pixelisées lors de l'export des images satellites source.

Le cas ci-dessous nous montre que les zones de doutes au moment de la classification sont les zones de variations de couleurs, et les effets de pixellisation liés à l'extraction des images.



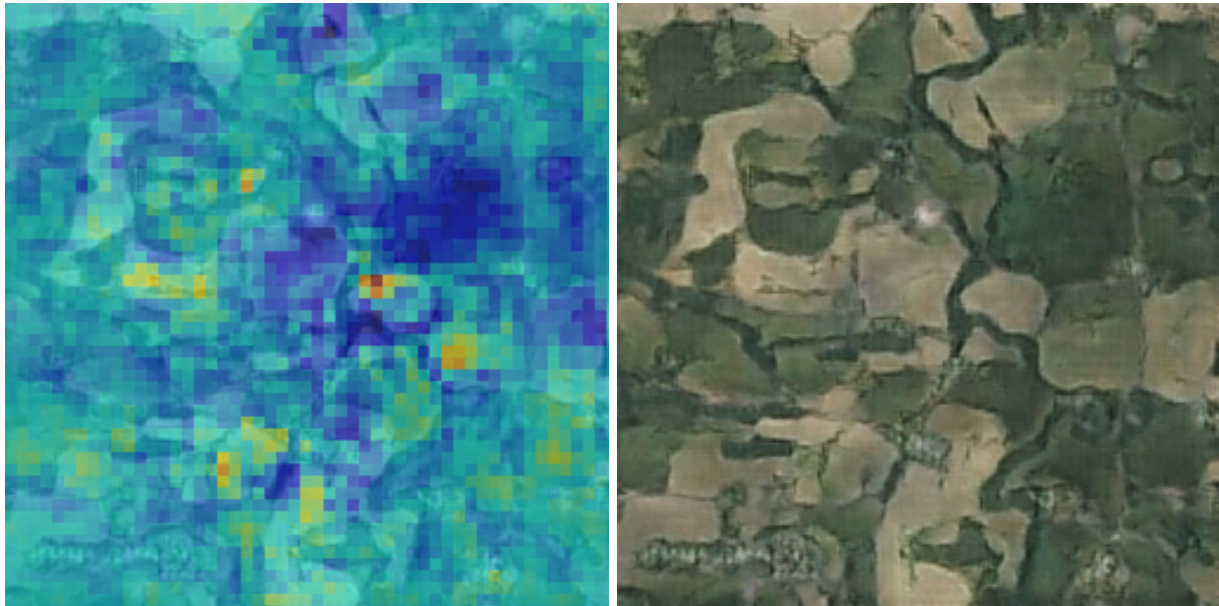
Dans ce second cas, le réseau semble globalement moins convaincu. Les zones de champs contenant des traces de labour semblent être considérées comme véridiques, possiblement par la linéarité des traits qui est moins présente dans les images générées.



Globalement, il semble plutôt confiant sur les zones sombres.

Fausse images satellites

Dans le cas des images générées, les formes arrondies sont détectées comme fausses. Les chevauchements de géométries également. Lorsque les séparations entre les zones claires sont épaisses, il a tendance à considérer qu'il s'agit de zones qui indiquent que l'image est fausse, alors que quand les lignes sont fines, il réagit plutôt inversement.



Dans le cas ci-dessous, on peut observer que les zones qui imitent des nuages surplombant les champs sont considérées comme vraies (car très bien imitées).

