



HAUTE ÉCOLE
D'INGÉNIERIE ET DE GESTION
DU CANTON DE VAUD
www.heig-vd.ch

HEIG-VD

RAPPORT INTERMÉDIAIRE

La Terre de nuit vue de l'espace

Antoine FRIANT
Haute École d'Ingénierie et de Gestion du
Canton de Vaud
Yverdon-les-Bains, VD, CH
antoine.friant@gmail.com

21 juillet 2018

Table des matières

1 Cahier des charges	iii
1.1 Résumé du problème	iii
1.2 Objectifs	iii
1.3 Limitations	iv
1.4 Description fonctionnelle	iv
1.5 Délais	iv
2 Résumé	v
3 Introduction	1
4 Gleam	2
4.1 Installation	2
4.2 Notebooks	2
4.2.1 country_stats	2
4.2.2 gleam	3
4.2.3 rastercomparator	3
4.2.4 scraper	3
4.2.5 viirs_extractor	3
5 Exploration des données	4
5.1 Jeux de données	4
5.1.1 Images satellite	4
5.1.2 Grilles de population	7

5.1.3	Pays	7
5.2	Recherche de corrélation	9
5.2.1	Pays	9
5.2.2	Grille de population	14
6	Modèle	18
6.1	Environnement de développement	18
6.2	Réseau de neurones	18
6.3	Première topologie	19
6.4	Premiers résultats	19
6.5	Améliorations	21
6.5.1	Pré-traitement	21
6.5.2	Topologie et paramètres	22
6.5.3	Prédiction	22
6.6	Réseau final	23
6.7	Résultats	23
6.8	Discussion	24
7	Conclusion	25
7.1	Discussion finale	25
7.2	Pour aller plus loin	25
8	Authentification	28
9	Symboles et abréviations	29

1 Cahier des charges

1.1 Résumé du problème

Les données géographiques sont nécessaires pour la prise de décisions importantes. Cependant la fiabilité et la disponibilité de ces données ne sont pas homogènes dans le temps et selon le lieu. Certaines de ces données ont une forte corrélation avec la lumière perçue par les satellites pendant la nuit.

Grâce à l'apprentissage automatique (*machine learning*), il est possible d'entraîner un réseau de neurones sur des données d'une date et d'un lieu connus pour reconstituer une carte de données géographiques à partir d'une image satellite nocturne.

Le travail à effectuer consiste à explorer différents types de données géographiques afin d'en choisir un, et faire de la prédiction sur ce type de données grâce à un réseau de neurones.

1.2 Objectifs

Le TB consiste dans un premier temps à explorer les données suivantes :

- Images satellites nocturnes de la Terre,
- Population humaine,
- Population animale,
- Densité végétale,
- PIB,

Et toutes autres données jugées pertinentes dans le but d'entraîner un réseau de neurones capable de prédire une estimation d'une donnée utile, à partir d'une image satellite de la terre de nuit.

La réalisation d'une application qui entraîne et exploite ce réseau de neurones est l'objectif de la seconde partie du TB.

Le but final est de pouvoir estimer, grâce au machine learning, des informations dont on ne possède pas de données à jour. Et cela à partir d'images satellites de nuit récentes, ou d'une combinaison de ces images avec une autre donnée à jour.

1.3 Limitations

L’application sera compatible avec Windows 10 et Archlinux, et nécessitera l’installation de librairies tierces (telles que Keras et TensorFlow). Elle ne possèdera pas nécessairement d’interface utilisateur.

L’utilisateur sera responsable de fournir les données à l’application dans un format supporté.

1.4 Description fonctionnelle

L’application prend en argument au moins deux jeux de données géographiques de format imposé : une image satellite nocturne et un autre type de données à déterminer au cours du projet. Après un long temps d’entraînement (une semaine au maximum, dépend de la machine utilisée), un modèle est généré.

Une fois le modèle généré, il est sauvegardé et réutilisable sur une autre image satellite nocturne (d’une date et/ou d’une région différente). Lorsque le modèle est appliqué sur une image satellite, une carte est recréée, affichant le résultat des prédictions.

Par exemple, si au cours du travail de bachelor il s’avère que la population par kilomètre carré est une donnée utile et utilisable, l’application devra prendre en argument une image satellite nocturne ainsi qu’une carte des populations de même taille et de même résolution pour entraîner le réseau de neurones. Une fois le modèle généré, l’application devra être capable de regénérer une approximation de la carte de population par kilomètre carré à partir d’une image satellite.

1.5 Délais

15 juin 2018 : Rapport intermédiaire

27 juillet 2018 : Rapport final et application fonctionnelle

Entre le 3 et le 14 septembre 2018 : Soutenance du travail de bachelor

2 Résumé

Les techniques d'apprentissage automatique offrent de nouvelles opportunités en matière de traitement données volumineuses. Ce projet s'intéresse en particulier au images satellites de la Terre de nuit. La disponibilité d'observations récentes en font une excellente base pour les prédictions d'une intelligence artificielle.

Ce travail consiste à explorer ces images, ainsi que d'autres données géographiques avec lesquelles il serait possible de tirer un parallèle : la démographie, l'écologie, l'économie, etc. Une fois un type d'information choisi, il s'agit d'entraîner un réseau de neurones capable de générer une prédiction pour ces valeurs à partir d'une image satellite nocturne.

Le modèle ainsi entraîné est capable de prédire, par exemple, la répartition de la population en Colombie avec un résolution au quart de kilomètre carré. Cette alternative aux comptages, bien que moins fiable, est disponible dès lors que l'image satellite est publiée car la génération d'une prédiction ne prend que quelques secondes. Celle-ci sera également de résolution plus élevée et bien moins coûteuse à produire.

3 Introduction

Les produits d'imagerie satellite sont devenus abondants et largement accessibles au cours des dernières décennies. De nombreux satellites prennent des photographies de la Terre à chaque heure du jour *et de la nuit*. Les observations nocturnes révèlent des caractéristiques peu évidentes de jour, parfois même cachées. Les routes apparaissent, les villes montrent leurs lumières, même les bateaux de pêche aveuglent les océans avec des projecteurs pour attirer les poissons.

La disponibilité, la résolution et l'uniformité de la qualité de ces données contrastent fortement avec le manque de fiabilité d'autres informations géographiques utiles lors de prises de décisions importantes. Par exemple, la répartition de la population est une estimation précise en Suisse mais très approximative au Kenya. D'autres mesures intéressantes incluent : la consommation en électricité, les émissions de C0₂, la couverture végétale et la présence de faune. Les lumières nocturnes observées depuis l'espace donnent des indications sur chacune de ces mesures alors qu'elles peuvent manquer dans une région à une date donnée.

Le but de ce projet est d'extraire autant d'informations que possible de l'imagerie satellite nocturne en utilisant l'apprentissage automatique (*machine learning*) sous la forme de réseau de neurones.

4 Gleam

Le projet Gleam ("lueur") se présente comme un ensemble de notebooks Jupyter disponible sur GitHub à l'adresse <https://github.com/Bertral/Gleam>. Il contient tous les scripts écrits au cours de ce travail. Les données brutes telles que les grilles de population, images satellites, cartes des frontières et statistiques officielles ne sont pas incluses dans le projet. Il est donc de la responsabilité de l'utilisateur de les obtenir depuis leur source d'origine, ainsi que de les traiter de manière à les rendre utilisables à l'aide de logiciels GIS (tels que QGIS) et des scripts du projet Gleam.

4.1 Installation

Les notebooks nécessitent la version 3.7 de Python. Ils ont été testés sur Windows 10 et Archlinux, mais devraient fonctionner sur toutes les versions de Windows et Linux.

Les librairies tierces `shapely`, `gdal`, `fiona` et `rasterio` sont compilées à partir de code C dans le but d'optimiser leur performance, mais cela a pour conséquence de n'être compatibles qu'avec Linux. Les utilisateurs Windows doivent donc les télécharger et les installer manuellement (`pip install nomdupackage.whl`) depuis le site de Christoph Gohlke [10], qui regroupe un grand nombre de librairies scientifiques recompilées pour Windows. Ce sont les versions "cp37" qui sont compatibles avec Python 3.7.

Pour installer le reste des pré-requis sur Linux et Windows, il suffit de lancer la ligne de commande `pip install -r requirements.txt` depuis la racine du projet.

Pour naviguer à travers les notebook sur Windows, il faut lancer la commande `python -m notebook`, ou `jupyter notebook` sur Linux.

4.2 Notebooks

Chaque notebook se trouve dans le répertoire portant son nom, à la racine du projet. Les fichiers générés par ces notebooks (cartes, graphes, logs et modèles) seront enregistrés dans le dossier du notebook.

4.2.1 country_stats

Ce notebook permet de compiler des statistiques sur la luminosité émise par chaque pays à partir d'une image satellite. Il produit ensuite une série de graphes montrant l'évolution de la

lumière vue de l'espace au fil des années pour chaque pays, ainsi qu'un nuage de point plaçant les pays selon leur lumière émise et leur population, PIB ou énergie consommée, colorés par développement économique.

Cette visualisation a pour but de chercher une corrélation à exploiter lors de la création du modèle.

4.2.2 gleam

Ce notebook est l'aboutissement du projet. Il entraîne un réseau de neurone à convolutions puis fait un prédition sur la répartition de la population sur une zone géographique à partir d'une image satellite nocturne.

4.2.3 rastercomparator

Ce notebook regroupe différentes manières de comparer deux fichiers GeoTIFF de même emprise : carte des différences par pixel, coefficient de corrélation et différence entre la somme des pixels. Il est utile pour évaluer la qualité d'une prédition ou observer l'évolution d'une carte d'une année à l'autre.

4.2.4 scraper

Ce script télécharge des images satellite fragmentées au format PNG et les assemble pour recréer des images globales. Comme le format GeoTIFF est largement supérieur au format PNG pour le travail à effectuer, ce notebook n'a pas servi pour le reste du projet.

4.2.5 viirs_extractor

Ce script répond au besoin spécifique d'extraire les archives volumineuses de la NOAA [7] contenant des images satellites haute résolution. Puis, il calcule une médiane annuelle pour chaque pixel des images mensuelles afin d'en éliminer les valeurs atypiques dues à l'absence d'observations sans nuages ou bruit.

5 Exploration des données

5.1 Jeux de données

5.1.1 Images satellite

NASA Worldview

La première source de données explorée est l'application "Worldview" de la NASA [14]. Elle permet de visionner un grand nombre d'images satellite composites sur un globe en trois dimensions (voir figure 5.1). Parmi les jeux de données disponibles sont trois jeux d'images nocturnes.



FIGURE 5.1 – Outil de visualisation NASA Worldview [14].

Le premier jeu de données est une série d'images composites capturées par le satellite Suomi NPP opéré par la NASA, la NOAA et le Département de la Défense des États-Unis. Il est mis à jour toutes les quelques heures, et présente une image composite chaque jour depuis le 30 novembre 2016. Elle possède deux défauts éliminatoires : la période d'observation actuellement disponible (à peine plus d'une année) n'est pas suffisamment longue pour observer une évolution

significative des villes depuis l'espace, et les images ne sont pas traitées. Cela signifie que celles-ci sont très fortement bruitées par les nuages et la lumière ambiante due aux différentes phases de la Lune.

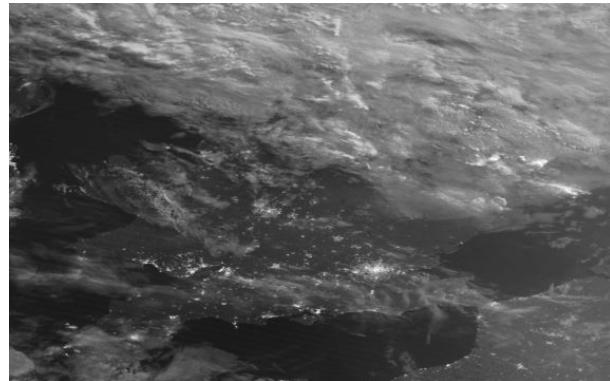


FIGURE 5.2 – Image satellite quotidienne servie par NASA Worldview [14], représentant la Grande Bretagne et son climat nuageux.

Les deux autres jeux de données nocturnes sont des images composites : des clichés pris tout au long de l'année ont permis de fabriquer une seule image du globe dont la luminosité ambiante est constante (moyennée) et sur laquelle les nuages n'apparaissent pas. Malheureusement, l'outil Worldview ne permet pas un téléchargement direct de ces images *dans leur pleine résolution*. Heureusement, la NASA a mis à disposition une API REST (<https://wiki.earthdata.nasa.gov/display/GIBS/GIBS+API+for+Developers>) pour télécharger des "tuiles" de n'importe laquelle de leur image. Seulement le format PNG est disponible. Ce format ne contient pas d'informations géographiques, ce qui complique leur utilisation pour la suite de ce travail. Un script Python suffit pour télécharger et assembler les tuiles (figure 5.3) et reconstituer une image complète du globe de plus de 800 millions de pixels (figure 5.4).

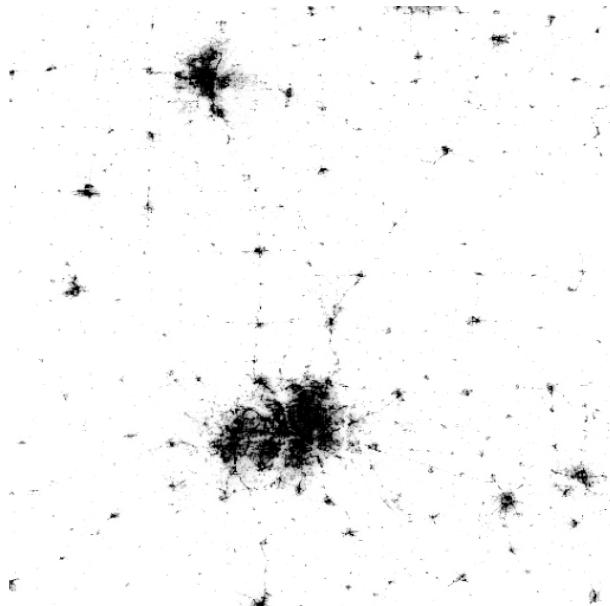


FIGURE 5.3 – Une tuile de l'image de 2016 montrant la ville de Dallas (USA) après avoir été mise en couleurs négatives.

Le script Python utilisé se trouve dans le notebook `scraper` et nécessite l'installation de la librairie Pillow pour le traitement des images, ainsi qu'urllib pour le téléchargement en soi. Son



FIGURE 5.4 – Image globale annuelle (2016) reconstituée à partir de tuiles téléchargées, puis mise en couleurs négatives.

exécution peut demander plus d'une heure pour le téléchargement (vitesse limitée par le serveur), et plus de 4 Go de RAM pour l'assemblage des tuiles.

Agence américaine d'observation océanique et atmosphérique

La source d'images satellite retenue pour la suite du travail est celle de l'Agence américaine d'observation océanique et atmosphérique (abrégé NOAA). Des images satellite nocturnes composites ("Average Lights X Pct") sont disponibles pour les années 1992 à 2013 [8] (1km par pixel, unités arbitraire de 0 à 63) et 2012 à 2018 [7] (mensuelles, 500m par pixel, radiance en nanoWatts/cm²/sr), une période suffisante pour observer des changements depuis l'espace. De plus, ces données sont disponibles en format GeoTIFF, qui contient les informations géographiques nécessaires pour superposer cette carte sur autre. Il est donc possible d'explorer et manipuler ces cartes à l'aide de logiciels libres tels que QGIS. Il s'agit en réalité des mêmes clichés fournis par Worldview (et Google Earth), mais plus nombreux et dans un format bien plus exploitable.

Ces images ont été créées en moyennant la valeur de luminosité de chaque pixel sur une année, en ignorant les pixels couverts par des nuages, et en multipliant cette moyenne par la fréquence de détection de lumière sur le pixel au cours de l'année. Les images mensuelles de 2012 à 2018, en revanche, possèdent des pixels pour lesquels il n'y a pas d'observation sans nuages ou lumière parasite. Pour éliminer ces valeurs atypiques, le notebook `viirs_extractor` créé une carte compilant la médiane annuelle de chaque pixel.

5.1.2 Grilles de population

Sedac [6] met à disposition des grilles de populations pour le monde entier, sous forme de fichier GeoTIFF. Chaque "case" de 1 km² est représentée comme un pixel et contient une estimation du nombre de personnes vivant dans cette case. QGIS s'est à nouveau montré d'une grande aide pour visualiser et manipuler ces données volumineuses (exemple en figure 5.5).

On remarque que la valeur de densité de population ne varie souvent pas à l'intérieur d'une sous-région. En effet, la qualité des données fournies par ces grilles est très variable selon les pays. Parmi les régions les plus détaillées on trouve les USA, l'Italie, le Portugal et le Brésil. En plus de cette disparité, il faut garder à l'esprit que ces répartitions sont les résultats de comptages qui ont eu lieu entre 2005 et 2014, puis ajustés en fonction du décompte des populations par pays. Il en résulte que les totaux de population sont fiables, mais la répartition de ces personnes à l'intérieur des territoires peut être périmée. Ces données ont été créées à partir de plusieurs mesures lorsque le décompte de la population pour une sous-région n'a pas été considéré comme fiable. Les images satellites font d'ailleurs partie de ces mesures supplémentaires, ce qui pourrait avoir comme effet d'améliorer la capacité de notre réseau de neurones à apprendre de ces données (corrélation forte).

Ces cartes sont disponibles pour les années 2000, 2005, 2010, 2015 et 2020.

5.1.3 Pays

Les grilles de données globales sont difficiles à créer, il n'en existe donc pas pour tous les types et toutes les dates. Afin de contourner ce problème, il y a des outils pour combiner des grilles (ici :

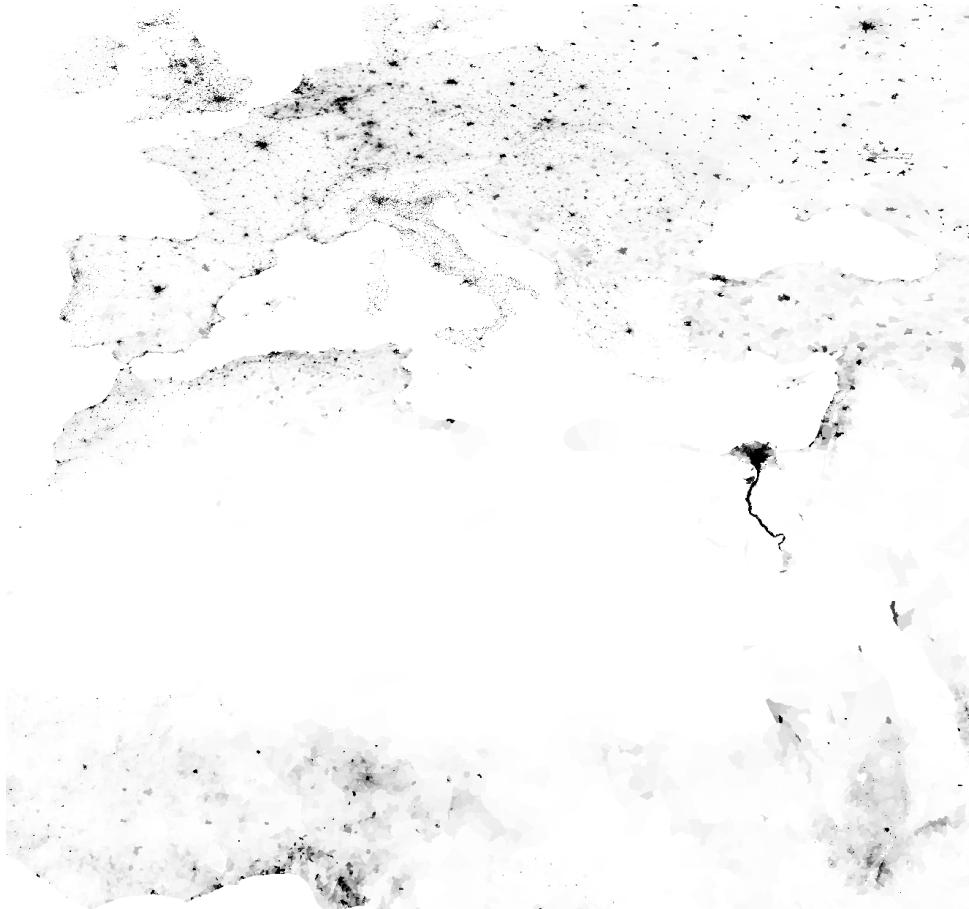


FIGURE 5.5 – Extrait de la grille de population [6] rendu avec QGIS. Le blanc indique une absence d'habitants, le noir indique au moins 1000 habitants par kilomètre carré.

l'image satellite) avec des données vectorielles telles que les frontières des pays. L'outil utilisé ici est la librairie Python **rasterstats**.

Le jeu de données "Admin 0 - Countries" de Natural Earth [11] contient les frontières des pays actuelles, ainsi que des méta-données sur chacun de ces pays (population estimée, indice de développement, différentes appellations et abréviations, etc.). En cas de conflits ou ambiguïté politique sur les frontières, c'est le pays qui contrôle le terrains qui est marqué comme souverain. Voici à quoi ressemblent les méta-données pour la Tunisie :

('scalerank', 0), ('featurecla', 'Admin-0 country'), ('LABELRANK', 3.0), ('SOVEREIGNT', 'Tunisia'), ('SOV_A3', 'TUN'), ('ADM0_DIF', 0.0), ('LEVEL', 2.0), ('TYPE', 'Sovereign country'), ('ADMIN', 'Tunisia'), ('ADM0_A3', 'TUN'), ('GEOU_DIF', 0.0), ('GEOUNIT', 'Tunisia'), ('GU_A3', 'TUN'), ('SU_DIF', 0.0), ('SUBUNIT', 'Tunisia'), ('SU_A3', 'TUN'), ('BRK_DIFF', 0.0), ('NAME', 'Tunisia'), ('NAME_LONG', 'Tunisia'), ('BRK_A3', 'TUN'), ('BRK_NAME', 'Tunisia'), ('BRK_GROUP', None), ('ABBREV', 'Tun.'), ('POSTAL', 'TN'), ('FORMAL_EN', 'Republic of Tunisia'), ('FORMAL_FR', None), ('NAME_CIAWF', 'Tunisia'), ('NOTE_ADMIN0', None), ('NOTE_BRK', None), ('NAME_SORT', 'Tunisia'), ('NAME_ALT', None), ('MAPCOLOR7', 4.0), ('MAPCOLOR8', 3.0), ('MAPCOLOR9', 3.0), ('MAPCOLOR13', 2.0), ('POP_EST', 11403800.0), ('POP_RANK', 14.0), ('GDP_MD_EST', 130800.0), ('POP_YEAR', 2017.0), ('LASTCENSUS', 2004.0), ('GDP_YEAR', 2016.0), ('ECONOMY', '6. Developing region'), ('INCOME_GRP', '3. Upper middle income'), ('WIKIPEDIA', -99.0), ('FIPS_10_', 'TS'), ('ISO_A2', 'TN'), ('ISO_A3', 'TUN'), ('ISO_A3_EH', 'TUN'), ('ISO_N3', '788'), ('UN_A3', '788'), ('WB_A2', 'TN'), ('WB_A3', 'TUN'), ('WOE_ID', 23424967.0), ('WOE_ID_EH', 23424967.0), ('WOE_NOTE', 'Exact WOE match as country'), ('ADM0_A3_IS', 'TUN'), ('ADM0_A3_US', 'TUN'), ('ADM0_A3_UN', -99.0), ('ADM0_A3_WB', -99.0), ('CONTINENT', 'Africa'), ('REGION_UN', 'Africa'), ('SUBREGION', 'Northern Africa'), ('REGION_WB', 'Middle East & North Africa'), ('NAME_LEN', 7.0), ('LONG_LEN', 7.0), ('ABBREV_LEN', 4.0), ('TINY', -99.0), ('HOMEPART', 1.0), ('MIN_ZOOM', 0.0), ('MIN_LABEL', 3.0), ('MAX_LABEL', 8.0)

Le notebook country_stats superpose ces données vectorielles à l'image satellite pour ajouter à chaque pays sa luminosité moyenne sur une échelle de 0 à 63 ainsi que l'écart-type de luminosité par pixel. Puis ces données sont enregistrées dans le fichier stats.pickle pour être utilisées plus tard.

Grâce à cette information sur la luminosité perçue par pays, on peut faire un parallèle avec une grande variété de données, telles que le produit intérieur brut (GDP), la consommation en énergie, le niveau de développement, l'indice économique, les émissions de CO₂, etc.

La source utilisée pour les données de population par pays et par année vient du site du Département des Affaires Économiques et Sociales des Nations Unies [13], plus précisément de la feuille Excel "Total Population - Both Sexes" de 2017.

La source de données pour le produit intérieur brut en USD par année est fournie par The World Bank [12] et téléchargée depuis http://data.un.org/Data.aspx?q=gdp&d=WDI&f=Indicator_Code%3aNY.GDP.MKTP.CD (visité le 07.06.2018, dernière mise à jour le 12.10.2016).

Les données sur la consommation en électricité en millions de kWh par pays et par année sont tirées de <http://data.un.org/Data.aspx?d=EDATA&f=cmID%3AEL> (données issues de la Division Statistique des Nations Unies, mises à jour en janvier 2018). Le filtre "Electricity - Final energy consumption" a été utilisé pour ne garder que la consommation totale, sans les nombreux détails proposés par la base de données sur l'usage de l'électricité.

5.2 Recherche de corrélation

5.2.1 Pays

La superposition de deux grilles (population et image satellite) de dimensions différentes n'est pas évidente, c'est pourquoi les premières données à avoir été mises en parallèles dans ce

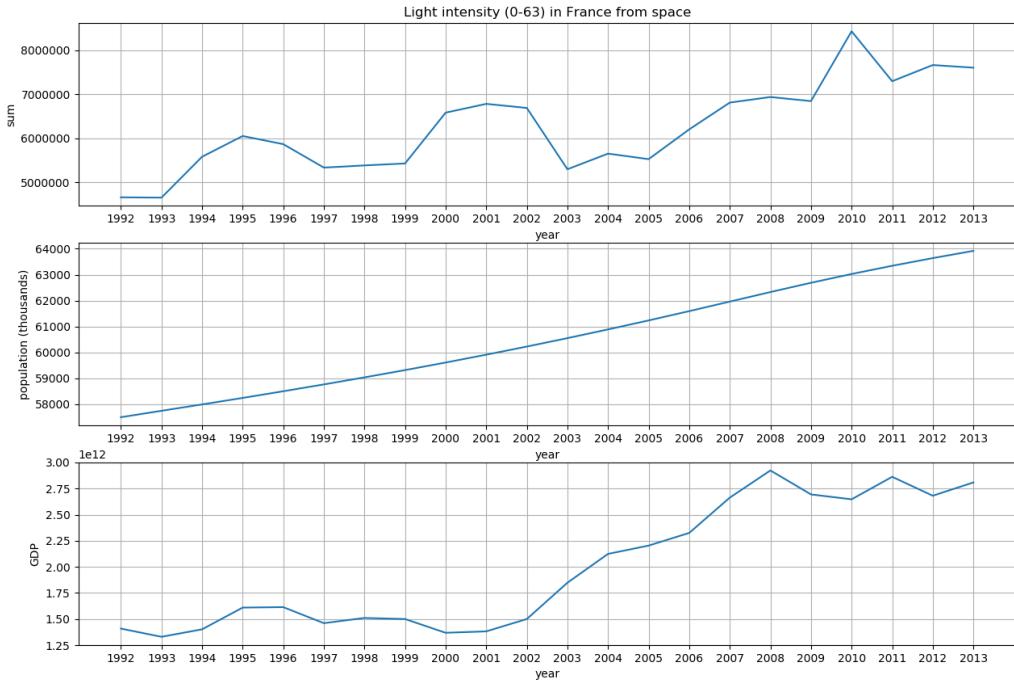


FIGURE 5.6 – Quantité de lumière perçue depuis l'espace, population et PIB de la France entre 1992 et 2013.

travail sont la population par pays [13], le produit intérieur brut [12], et la luminosité totale du pays perçue depuis l'espace (extraite par script Python (notebook country_stats) à partir des vecteurs de frontières de Natural Earth [11] et des images de la NOAA [8]).

Pour chaque année disponible de 1992 à 2013, il est donc possible de dessiner pour la majorité des pays les histogrammes de luminosité, population et PIB (exemples en figures 5.6, 5.7, 5.8 et 5.9). Cette représentation n'est pas idéale car l'échelle n'est pas constante entre les pays. Ce qui apparaît comme une grande variation de luminosité peut ne pas en être du tout. Certains pays ne portent pas exactement le même nom dans tous les jeux de données, ils sont donc mis de côté dans les résultats.

La figure 5.6 est typique des résultats obtenus. Elle laisse présager une forte corrélation entre les trois données observées. Cependant, la figure 5.7 soulève déjà des doutes : la population croît de façon linéaire, mais le PIB et la lumière perçue augmentent de plus en plus vite. Les graphes du Japon (figure 5.8) semblent indiquer que la luminosité suit le PIB également lorsque ce dernier diminue. Enfin le cas de l'Arménie (figure 5.9) nous pousse à croire que le PIB a une corrélation plus forte que la population avec la luminosité.

Jusqu'ici, rien n'est confirmé. Nous ne possédons que des observations par pays sur des échelles variables qui sont regroupées par pays, ce qui représente une résolution plutôt basse. Sans parler de fait que 320 pays sur 21 années n'est pas une quantité de données acceptable pour entraîner un réseau de neurones.

Il est également possible de générer des nuages de points comparant la lumière émise et, au choix, l'énergie consommée, la population du pays ou le PIB, et de colorer ces points par indice de développement économique (tiré des données vectorielles Natural Earth [11]). Ces graphes

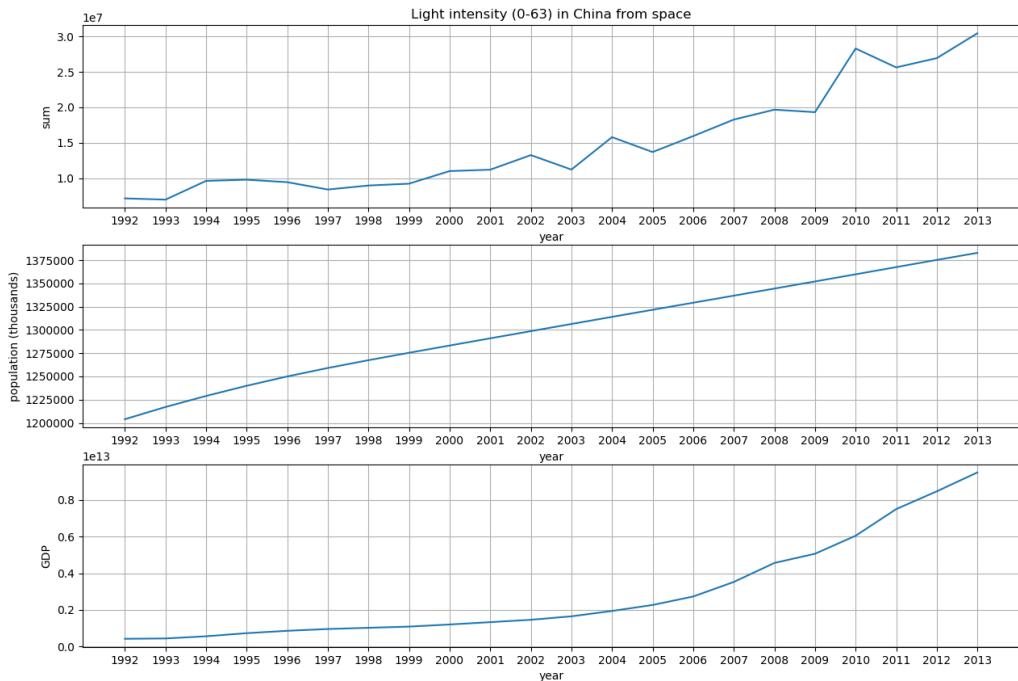


FIGURE 5.7 – Quantité de lumière perçue depuis l'espace, population et PIB de la Chine entre 1992 et 2013.

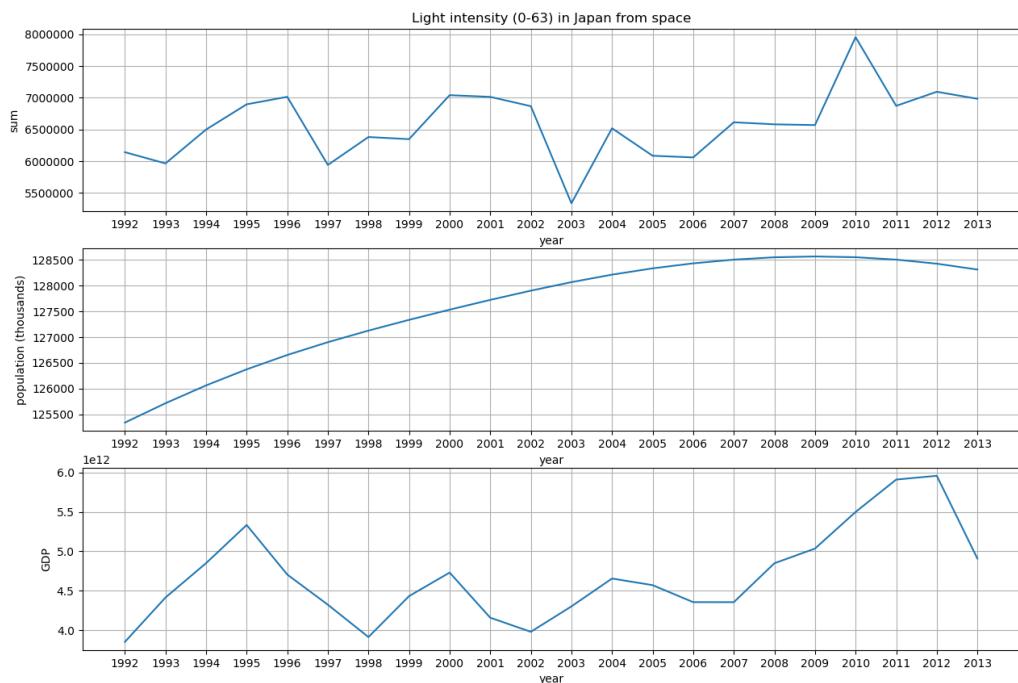


FIGURE 5.8 – Quantité de lumière perçue depuis l'espace, population et PIB du Japon entre 1992 et 2013.

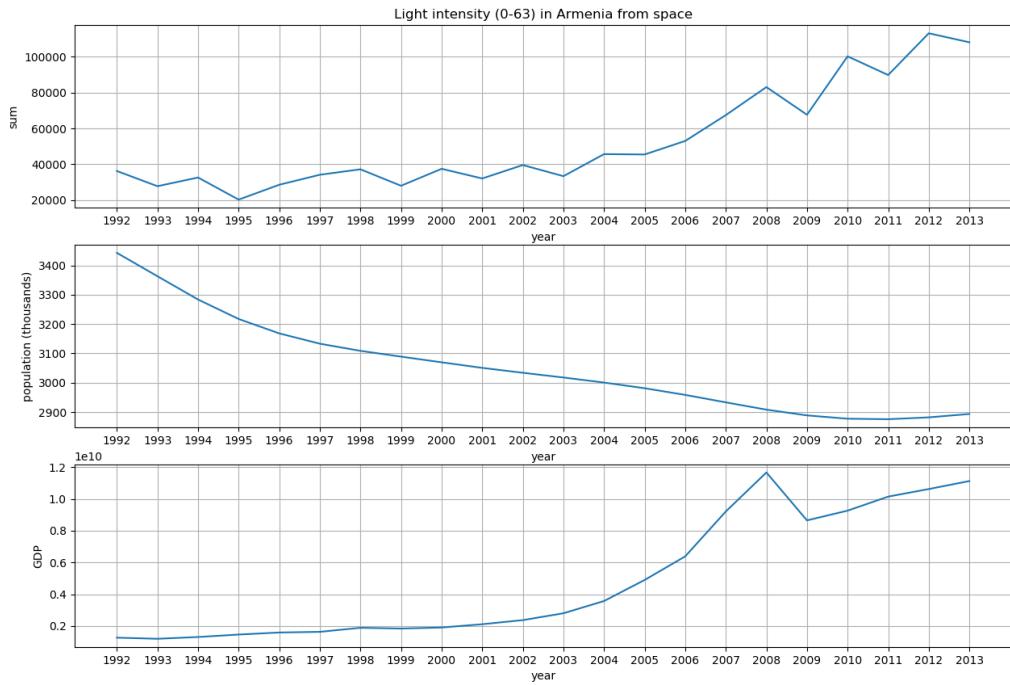


FIGURE 5.9 – Quantité de lumière perçue depuis l'espace, population et PIB de l'Arménie entre 1992 et 2013.

sont générés grâce au notebook `country_stats`.

Le premier de ces nuages de points compare la luminosité et la population de 2013 pour chaque pays (figure 5.10). On observe immédiatement qu'il existe une corrélation. Son coefficient de Pearson est 0.56. Il est très important de constater que cette corrélation n'apparaît que lorsque l'échelle est logarithmique, il faudra donc retenir que cette relation n'est pas linéaire. On ne peut pas encore déterminer s'il y a une relation directe, ou si ces deux variables sont simplement corrélées à la taille du territoire. En effet, on ne fait que sommer la population et la quantité de lumière émise, on ne calcule pas de moyenne par pays. Sans surprise, on voit également que les pays ayant un index économique élevé émettent plus de lumière que d'autres pays à population équivalente.

Le deuxième graphe généré est beaucoup plus intéressant car il compare le produit intérieur brut de chaque pays avec leur émission de lumière (figure 5.11). On observe une forte corrélation, dont le coefficient de Pearson est 0.819. Naturellement, l'indice de développement économique est très corrélé avec le PIB (*GDP* en Anglais).

Une autre comparaison est possible avec la consommation en électricité, illustrée en figure 5.12. La forte corrélation (coefficient de 0.81) entre émission de lumière et consommation d'énergie n'est pas une surprise, mais on remarque un phénomène intéressant avec l'index économique des pays. Les pays en voie de développement ont tendance à émettre beaucoup plus de lumière pour une consommation d'électricité équivalente aux pays développés. On peut spéculer sur les causes d'une telle différence (data centers, chauffages, etc.) mais ce n'est pas l'objet de ce travail.

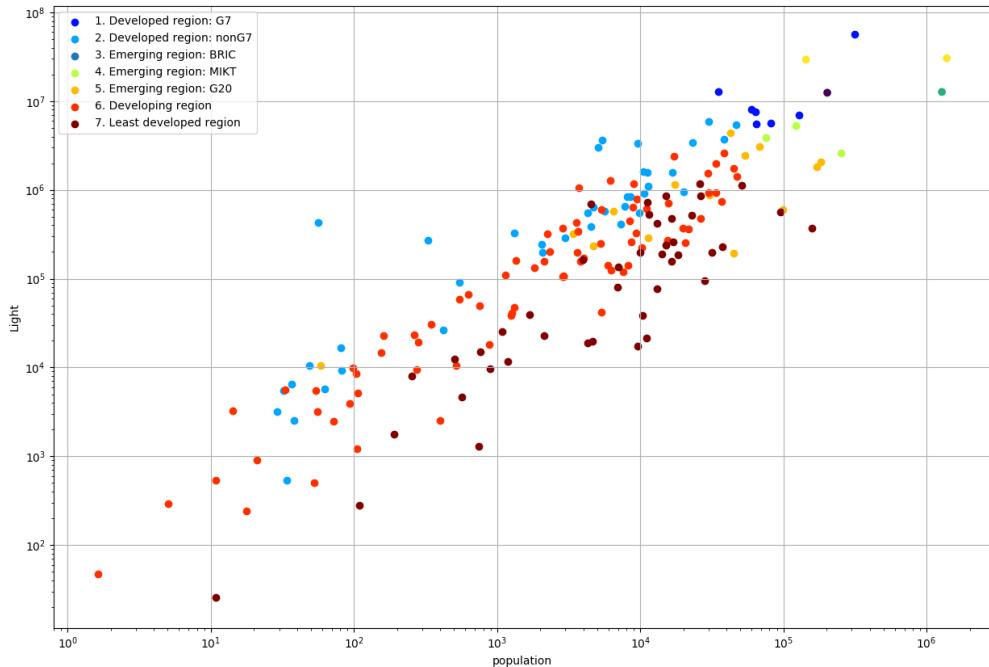


FIGURE 5.10 – Pays placés par population et luminosité totale émise sur une échelle logarithmique en 2013, colorés par indice économique.

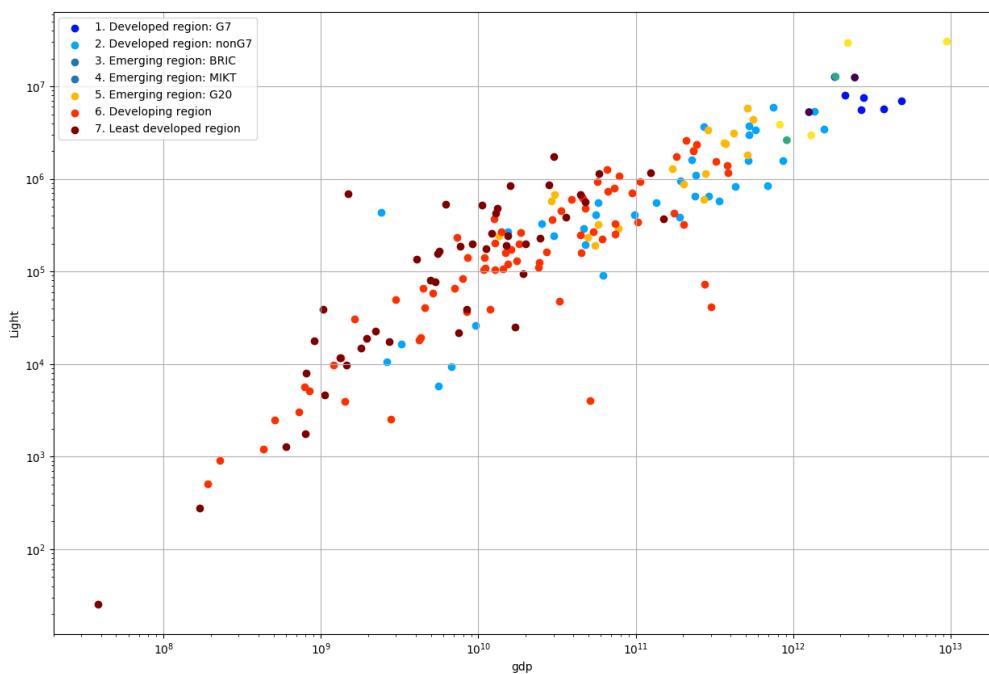


FIGURE 5.11 – Pays placés par PIB en USD et luminosité totale émise sur une échelle logarithmique en 2013, colorés par indice économique.

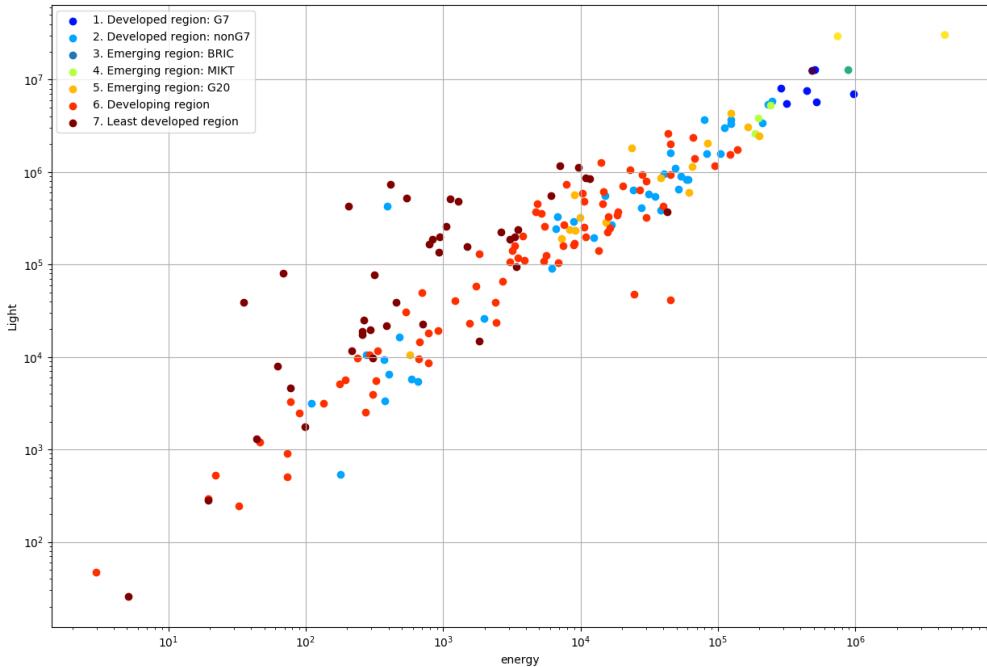


FIGURE 5.12 – Pays placés par consommation en électricité en millions de kWh et luminosité totale émise sur une échelle logarithmique en 2013, colorés par indice économique.

5.2.2 Grille de population

Après avoir tenté d'écrire un script Python pour superposer deux grilles de dimensions différentes, il s'est avéré que l'application QGIS est capable d'effectuer cette opération. Il suffit de :

- Ouvrir les grilles avec QGIS : population et image satellite. Elles doivent apparaître dans la liste des couches.
- Dans la barre d'outils, choisir "Raster" → Divers → Fusionner.
- Cocher l'option "Placer chaque fichier en entrée dans une bande séparée." puis lancer la fusion "Run in Background".

La grille résultante contient donc environ 800 millions de pixels (possédant chacun une valeur de population et de luminosité) qui sont potentiellement autant de données d'entraînement pour chaque année disponible (2000, 2005, 2010, qui correspondent aux dates des grilles de population DMSP-OLS [8]). Après avoir ajusté ses propriétés d'affichage et inversé les couleurs, on obtient la figure 5.13. Le bleu ciel correspond à la lumière visible depuis l'espace, le rose la population (sur une échelle de 1 à 1000 habitants par km²). Le bleu foncé correspond au chevauchement des deux couleurs. On peut déjà observer que, si la luminosité ne suit pas l'évolution de la population dans le temps à l'échelle d'un pays, elle est tout de même concentrée géographiquement sur les points les plus peuplés.

La superposition des grilles de population (1 pixel par km) avec un image satellite issue de VIIRS [7] dont la résolution est 4 fois plus élevée. Les outils de traitement de cartes ont tendance à augmenter la résolution de la grille de population en utilisant la méthode du plus proche voisin.



FIGURE 5.13 – Superposition de l'image satellite nocturne (bleu ciel) et de la grille de population (rose).

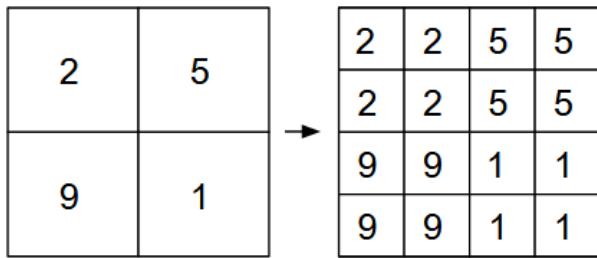


FIGURE 5.14 – Effet de l'augmentation de la résolution de la grille de population lors de la fusion

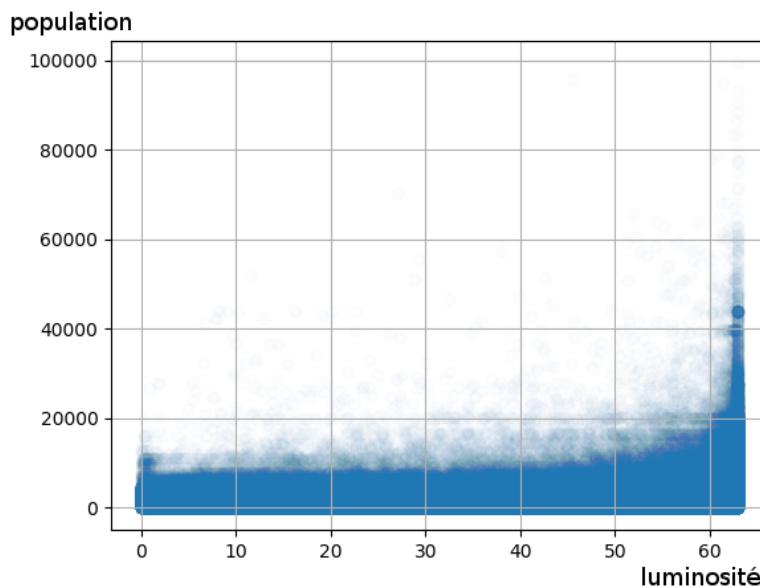


FIGURE 5.15 – Valeur de luminosité et de population mondiale pour chaque km² (année 2000)

Chaque pixel donne donc correctement la densité de population par kilomètre carré, mais la population totale est multipliée par 4. La figure 5.14 illustre ce problème. Pour corriger cela, il est nécessaire d'ajuster la population par pixel en la divisant par 4. C'est le rôle du script `gleam/rescale_pop.py` (attention à ajuster les constantes avant de le lancer).

Si on pose chaque pixel sur un nuage de point dont les axes sont la luminosité et la population, on obtient la figure 5.15. La figure 5.16 représente uniquement le Brésil en 2015. On constate qu'il sera très difficile de deviner la valeur de population à partir de l'intensité lumineuse d'un seul pixel. Cela nous oriente donc plutôt dans la direction d'un réseau de neurones à convolutions, qui considère les pixels par groupes.

Enfin, les données vectorielles de Natural Earth [11] peuvent nous permettre d'isoler précisément un pays du reste de la grille :

- Ouvrir la grille et le fichier vectoriel des pays avec QGIS dans le même projet.
- Cliquer sur la couche vectorielle, puis sélectionner sur la carte le pays à isoler (avec l'outil de sélection d'entité).
- Dans la barre d'outils, sous "Raster", choisir "Extraction" puis "Découper un raster selon une couche masque".
- Définir la grille comme couche source, et la couche vectorielle comme masque.
- Cocher "Entité(s) sélectionnée(s) uniquement".
- Lancer l'extraction "Run in Background".

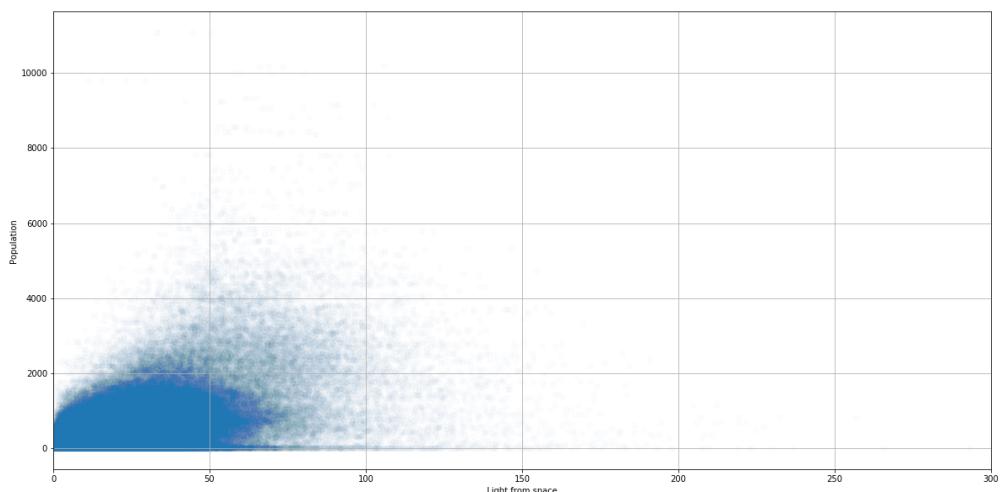


FIGURE 5.16 – Valeur de luminosité et de population pour chaque 0.25km^2 du Brésil (année 2015)

6 Modèle

6.1 Environnement de développement

Python est un langage largement utilisé pour manipuler des données et faire de l'apprentissage automatique. Il dispose de librairies optimisées précisément pour cette tâche, dont Keras.

Keras est une librairie Python open source permettant le prototypage rapide de réseaux de neurones et peut fonctionner au-dessus de TensorFlow (open source et développé par Google), CNTK ou Theano. Une fonctionnalité très attractive de TensorFlow est l'exploitation automatique du processeur graphique s'il est disponible. En effet, ces processeurs sont très performants pour l'apprentissage automatique, car c'est un travail hautement parallélisable.

Afin d'activer l'utilisation d'un GPU Nvidia par Tensorflow sur Windows 10, il faut installer la librairie tensorflow-gpu, le CUDA Toolkit de Nvidia en version 9.0 et le SDK cuDNN (*CUDA Deep Neural Network library*) de Nvidia en version 7.0 (instructions détaillées sur https://www.tensorflow.org/install/install_linux).

6.2 Réseau de neurones

Pour commencer à développer le réseau de neurones, on commencera par tenter de prédire la population d'une petite région à partir de l'image satellite de l'année 2005, en entraînant la machine sur celle de 2000 et la grille de population correspondante [6].

Comme on a pu observer que la prédiction à partir d'un pixel isolé n'est pas réaliste, on aimerait prendre en compte les valeurs des pixels voisins. En effet, s'il y a beaucoup de pixels proches illuminés, il y a de meilleures chances pour que la région soit densément peuplée que si un seul pixel est illuminé. De plus, si le réseau de neurones peut reconnaître les formes, il sera en mesure de différencier le centre et la périphérie d'une ville, ainsi que des routes ou villages isolés.

Le système qui répond à ces exigences est le réseau de neurones à convolutions. Cependant, alors que l'usage habituel d'un tel réseau sert à la classification de données, on a besoin ici d'obtenir un nombre réel. Il s'agit d'adapter le système pour faire de la régression, ce que l'on fait lorsqu'on compte le nombre de voitures sur un parking par exemple. Les excellents articles *Regression Tutorial with the Keras Deep Learning Library in Python* [5], *Crash Course in Convolutional Neural Networks for Machine Learning* [1], *Handwritten Digit Recognition using Convolutional Neural Networks in Python with Keras* [3], *Object Recognition with Convolutional Neural Networks in the Keras Deep Learning Library* [4] et *Evaluate the Performance Of Deep Learning Models in Keras* [2] de Jason Brownlee ont été d'une grande aide pour la compréhension des bonnes pratiques et l'utilisation de Keras.

Avant de commencer l'entraînement, on découpe l'image satellite nocturne en tuiles de 32 sur 32 pixels (1024 km^2). Chacune de ces tuiles sera considéré comme une observation à donner en entrée du modèle. La sortie sera une valeur réelle correspondant au nombre d'habitants dans la zone donnée en entrée.

6.3 Première topologie

La topologie du réseau (figure 6.1) s'inspire grossièrement du travail effectué par l'Arnhold Institute for Global Health [9], qui consiste à estimer les populations de petites régions à partir d'images satellite de jour *et* de nuit. La première version de notre réseau de neurones en utilise une variante très allégée.

Les couches sont définies comme suit :

- Convolution de 32 filtres, chaque kernel fait 3x3 pixels, fonction d'activation ReLU,
- Max pooling pour diviser par 2 la largeur et la hauteur de la tuile,
- Convolution identique à la première couche,
- Flatten,
- Dropout de 20% pour réduire les chances d'overfitting,
- Couche dense de 32 neurones,
- Couche dense d'un seul neurone, qui correspond à la sortie du modèle.

L'optimiseur utilisé est Adam. C'est généralement un bon choix par défaut grâce à sa performance en terme de vitesse de calcul, et à sa capacité à répondre aux besoins d'un grand nombre de problèmes. Le *learning rate* choisi (après quelques essais) est 0.001. Enfin, la fonction objectif à optimiser est la moyenne des erreurs au carré.

6.4 Premiers résultats

Les résultats suivants ont été obtenus en entraînant le modèle sur une région couvrant l'Europe de l'Ouest et l'Afrique du Nord en l'an 2000. La phase de test est effectuée sur l'Amérique du Nord et une partie de l'Amérique du Sud en l'an 2005.

Les figures 6.2 et 6.3 présentent respectivement l'évolution des moyennes des erreurs absolues et des erreurs au carré pendant l'entraînement du modèle sur 100 itérations.

Les résultats du test sur l'Amérique en 2005 sont plutôt médiocres. La fonction objectif (moyenne des erreurs au carré) vaut 5381938110.04, et la moyenne des erreurs absolues vaut 26599.64. Ce qui veut dire que sur une région de 1024 km^2 , la prédiction du modèle se trompe en moyenne de 26599.64 habitants. C'est significativement plus mauvais que les résultats obtenus sur le jeu d'entraînement. Cela signifie que le modèle apprend car l'erreur diminue lors de l'entraînement, mais il n'apprend rien de généralisable. C'est ce qui s'appelle du surapprentissage. De plus, une erreur moyenne de 26599.64 par tuiles représente une erreur moyenne d'environ 26 habitants par kilomètre carré. À titre de comparaison, la densité de population du monde est d'environ $7000000000 / 500000000 = 14$ habitants par kilomètre carré, océans compris.

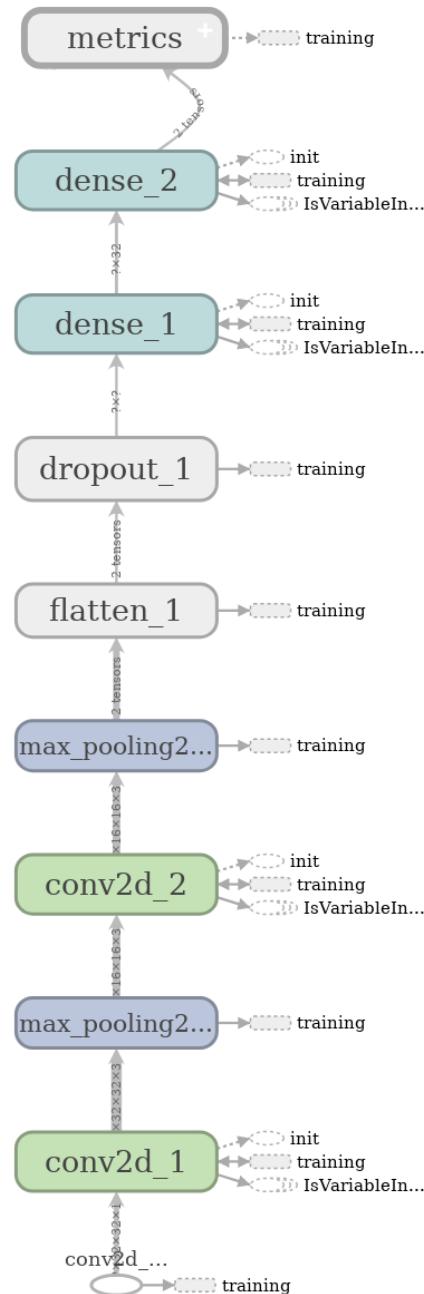


FIGURE 6.1 – Topologie de la toute première version du réseau de neurones.

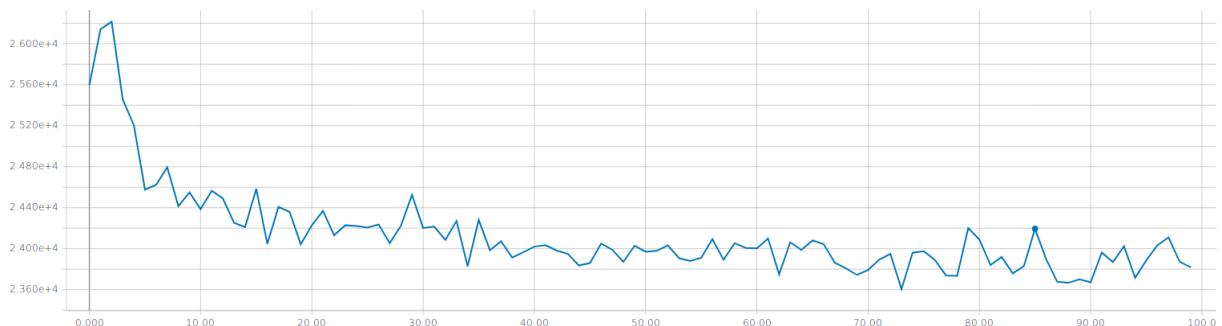


FIGURE 6.2 – Moyennes des erreurs absolues sur 100 itérations.

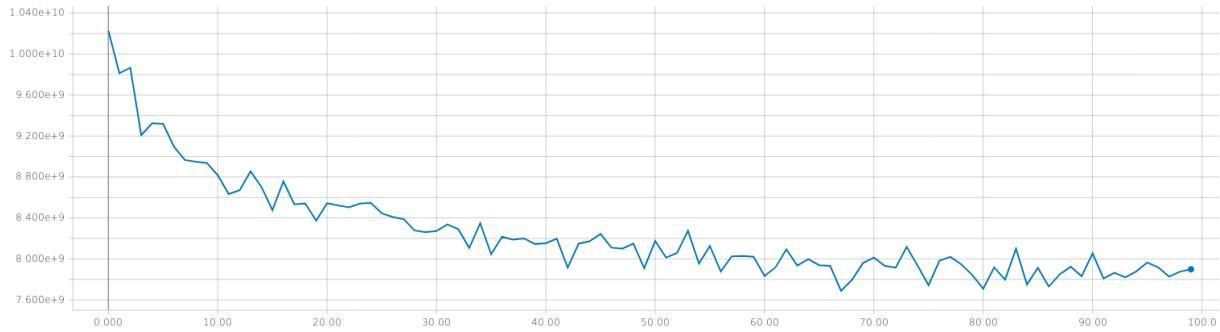


FIGURE 6.3 – Moyennes des erreurs au carré (fonction objectif) sur 100 itérations.

6.5 Améliorations

6.5.1 Pré-traitement

La plus importante amélioration en performance que l'on peut obtenir sur un réseau de neurones vient d'un pré-traitement adapté. La façon dont les informations sont présentées au modèle influence grandement sa capacité à apprendre. Cet aspect n'a pas été travaillé pour le premier modèle. Voici donc les améliorations apportées.

Premièrement, la Terre est recouverte à plus de 70% d'eau. Cette surface n'est pas occupée par des habitations. Le jeu de données comporte donc une grande majorité d'observations superflues qui peuvent interférer avec l'apprentissage de caractéristiques plus importantes. Le processus de création des observations de 32 sur 32 pixels doit donc exclure les éléments vides. En filtrant les tuiles sans population de l'entraînement et la validation, on obtient une erreur moyenne plus grande car il y a une plus importante proportion d'éléments difficiles à prédire.

Ensuite, sans même parler de l'influence de l'économie d'un pays sur ses émissions de lumière, la qualité des données de population est très inégale entre les pays. La figure 6.4 en est un exemple. Sur la grande majorité des pays du monde, et même de l'Europe, la grille de population ne propose qu'une seule estimation de la densité pour de grandes régions. Cela devient particulièrement problématique lorsqu'une observation contient une ville isolée émettant de la lumière, mais qu'elle se trouve au milieu d'une grande région dont la densité de population est basse. Ces cas sont nombreux et réduisent la corrélation entre population et luminosité que le modèle doit exploiter pour apprendre. Afin de gagner en performance, on peut découper la carte en isolant un pays dont les données de population sont de bonne qualité. C'est faisable avec QGIS par exemple, et un fichier vectoriel pour les frontières tel que celui de Natural Earth [11]. Tous les tests dans le but d'améliorer la topologie du réseau de neurones seront faits sur le Brésil uniquement (bonne qualité, grande surface).

Cependant, restreindre l'entraînement à un pays réduit la taille du jeu de données et augmente le risque de surapprentissage. Pour augmenter la quantité d'observations, chaque tuile de 32 sur 32 pixels est décalée de seulement 8 pixels au lieu de 32. Ainsi, aucune observation n'est identique mais elles se superposent partiellement. Il est donc judicieux de ne pas utiliser cette méthode lors de la validation croisée.

Enfin, réduire la taille des tuiles a donné de moins bons résultats lors de la validation, leur taille restera donc de 32 sur 32 pixels. L'image satellite et la grille de populations utilisée [6] seront celles de 2015 afin d'exploiter les images de plus haute résolution (500m par pixel) de la NOAA [7].

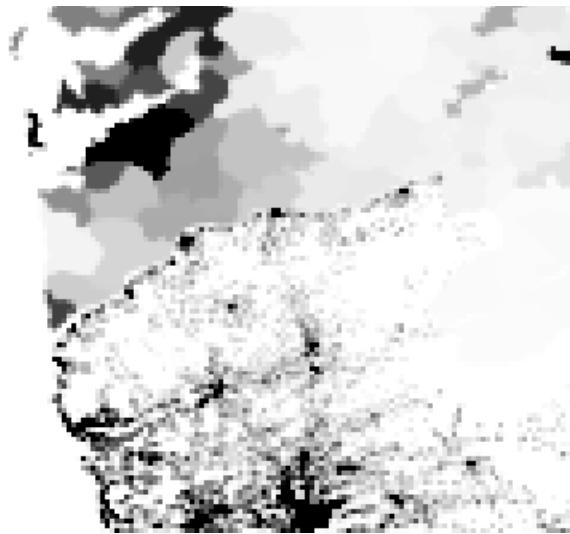


FIGURE 6.4 – Grille de population pour la frontière nord entre l’Espagne et le Portugal illustrant les différences de qualité des données de population d’un pays à l’autre.

6.5.2 Topologie et paramètres

Plus de 50 modèles ont été générés dans le but de tester différents ajustements sur le réseau de neurones et ses paramètres. Ces tests ont été faits en utilisant la validation croisée 4-fold sur au moins 150 itérations. Le jeu de données d’entraînement est la Colombie en 2015 dont les tuiles ne se superposent pas (décalage de 32 pixels entre chaque observation de 32 sur 32). Voici un résumé non exhaustif de ces tests.

- Ajustements de la taille des filtres des convolutions,
- Ajouts/suppressions de couches de convolution et de max pooling,
- Ajouts/suppressions de couches denses accompagnées de dropout,
- Ajustements de la force du max pooling,
- Ajustements du learning rate,
- Implémentation de learning rate dégressif,
- Pondération des observations selon la population présente (non concluant),
- Ajustements du nombre de neurones dans la couche dense,
- Ajustements des valeurs de dropout,
- Passage du max pooling à l’average pooling, car lorsque l’on réduit la résolution des grilles on veut sommer les valeurs plutôt que de garder la valeur maximale des pixels. L’average pooling remplit ce rôle car la moyenne des pixels est la somme divisée par une constante dans le cas du pooling.

La section 6.6 détaille la topologie et les paramètres retenus après ces tests.

6.5.3 Prédiction

Les prédictions ne peuvent être faites que sur des images satellite de même échelle (mètres par pixel) que celle qui a servi à entraîner le réseau. Le résultat de la prédiction est une estimation de la population présente sur l’observation en entrée du modèle, ce qui réduit fortement la résolution de l’image en sortie. Cependant, grâce à l’article Equitable development through deep learning [9] et l’exploration préalable des données, on sait que la répartition de population est en relation

exponentielle avec la répartition de la luminosité.

Cette constatation nous permet de répartir approximativement la population prédictive par le modèle pour recréer une tuile de dimension identique à l'observation en entrée du modèle, c'est-à-dire 1024 pixels. On peut valider cette répartition en calculant le coefficient de corrélation de la prédiction et de la donnée d'entraînement. Avec le modèle entraîné sur le Brésil pour 2015, en comparant la grille de population à la prédiction de la population du Brésil en 2015, on obtient les coefficients de corrélation :

- 0.60771883 avec une distribution pondérée linéairement à la luminosité des pixels de la tuile,
- 0.69339848 avec une distribution exponentielle,
- 0.63450044 avec une distribution logarithmique,
- 0.65867300 avec une distribution quadratique.

6.6 Réseau final

La topologie du réseau retenu est la suivante :

Couche (nb de neurones)	Dimensions de la sortie	Nb de paramètres
Convolution (64)	$32 \times 32 \times 64$	640
Average pooling	$16 \times 16 \times 64$	0
Convolution (128)	$16 \times 16 \times 128$	73856
Average pooling	$8 \times 8 \times 128$	0
Convolution (256)	$8 \times 8 \times 256$	295168
Average pooling	$4 \times 4 \times 256$	0
Flatten	4096	0
Dropout 50%	4096	0
Dense (128)	128	524416
Dense (1)	1	129
Total		894209

Paramètres supplémentaires :

- La taille des filtres est 3 sur 3 pixels pour toutes les couches de convolution.
- Chaque average pooling divise la largeur et la hauteur de l'image en entrée de la couche par 2.
- Le dropout abandonne 50% des neurones lors des étapes d'entraînement pour éviter le surapprentissage.
- L'ajustement des paramètres (rétropropagation) utilise l'algorithme Adam, et ne s'effectue que toutes les 1024 mesures. Cela permet de paralléliser les calculs efficacement pour un processeur graphique et réduit le nombre de rétropropagations à effectuer.
- La fonction objectif est l'erreur quadratique moyenne. Cela permet de réduire la variance des erreurs ainsi que l'erreur absolue.
- Le learning rate commence à 0.02. Il est divisé par 2 lorsqu'il n'y a pas eu d'amélioration de la fonction objectif dans les 20 dernières itérations. Il ne descend pas plus bas que 0.0000001.
- Si la fonction objectif ne s'améliore pas pendant 200 itérations, l'entraînement s'arrête.
- Le nombre maximal d'itérations est fixé à une valeur très élevée de sorte à ne pas interrompre l'entraînement tant que la fonction objectif s'améliore.
- L'ordre de traitement des observations est randomisé à chaque itération pour ne pas

grouper les observations par région.

6.7 Résultats

A coups de rastercomparator et rasterstats

Aperçu : Brésil réalité 2015 vs prédiction 2017 Pearson correlation : 0.7499682639089719 Sum of values : 204777100.0, 272478200.0 Sum increase : 67701100.0 (33.060875790098095 %) Mean absolute error (by pixel) : 2.044411

Brésil réalité 2015 vs prédiction 2015 Pearson correlation : 0.693398475250684 Sum of values : 204777100.0, 204874460.0 Sum increase : 97360.0 (0.047544377812863296 %) Mean absolute error (by pixel) : 1.5557544

italie réalité vs italie prédiction, entraîné *partiellement* sur USA : Pearson correlation : 0.5192059500037354 Sum of values : 58895904.0, 39399708.0 Sum increase : -19496196.0 (-33.102804568548606 %) Mean absolute error (by pixel) : 5.861093

6.8 Discussion

7 Conclusion

7.1 Discussion finale

Nous avons exploré une partie des données à disposition. Nous savons désormais où chercher, quel format utiliser et comment exploiter les données géographiques. Il devrait être relativement facile de substituer la grille de population à une autre, plus utile, quand on aura démontré l'efficacité du réseau de neurones.

Le travail effectué, bien que ne fournissant pas de résultat exploitable, a permis de très clairement définir les priorités à court et moyen terme :

- Faire en sorte que le modèle neural produise une estimation par pixel plutôt que par région (autant de pixels en entrée qu'en sortie).
- Adapter les méta-paramètres du modèle neural afin d'obtenir de meilleurs résultats.
- Réduire la consommation en mémoire des scripts d'entraînement et de test.
- Tenter de faire des parallèles avec d'autres données que la population, notamment écologiques.
- Entraîner un réseau à prédire l'évolution d'une image satellite au fil des années, afin de créer des images satellites futures sur lesquelles faire d'autres prédictions (génération d'une grille de population dans 20 ans par exemple).
- Rendre l'utilisation des scripts d'entraînement et de test plus résistante aux changements de formats des données. Il faut qu'un nouvel utilisateur n'ait pas besoin de plonger dans le code source pour adapter la résolution de l'image satellite, par exemple.
- Optimiser la performance du modèle, que ce soit par la transformation ou l'enrichissement des données d'entraînement, ou par la modification des méta-paramètres.

7.2 Pour aller plus loin

Plus de pays

ajouter les images de jour

prédirer la carte des luminosités des années prochaines

évolution par mois sur une année "typique"

utiliser des données de densité de population plus spécialisées et récentes (étude sur un seul pays p. ex.)

Bibliographie

- [1] Jason BROWNLEE. *Crash Course in Convolutional Neural Networks for Machine Learning*. 24 juin 2016. URL : <https://machinelearningmastery.com/crash-course-convolutional-neural-networks/> (visité le 23/04/2018).
- [2] Jason BROWNLEE. *Evaluate the Performance Of Deep Learning Models in Keras*. 26 mai 2016. URL : <https://machinelearningmastery.com/evaluate-performance-deep-learning-models-keras/> (visité le 23/04/2018).
- [3] Jason BROWNLEE. *Handwritten Digit Recognition using Convolutional Neural Networks in Python with Keras*. 27 juin 2016. URL : <https://machinelearningmastery.com/handwritten-digit-recognition-using-convolutional-neural-networks-python-keras/> (visité le 23/04/2018).
- [4] Jason BROWNLEE. *Object Recognition with Convolutional Neural Networks in the Keras Deep Learning Library*. 1^{er} juil. 2016. URL : <https://machinelearningmastery.com/object-recognition-convolutional-neural-networks-keras-deep-learning-library/> (visité le 23/04/2018).
- [5] Jason BROWNLEE. *Regression Tutorial with the Keras Deep Learning Library in Python*. 9 juin 2016. URL : <https://machinelearningmastery.com/regression-tutorial-keras-deep-learning-library-python/> (visité le 23/04/2018).
- [6] CENTER FOR INTERNATIONAL EARTH SCIENCE INFORMATION NETWORK-CIESIN-COLUMBIA UNIVERSITY. *Gridded Population of the World, Version 4 (GPWv4) : Population Count Adjusted to Match 2015 Revision of UN WPP Country Totals, Revision 10*. 2017. DOI : [10.7927/h4jq0xzw](https://doi.org/10.7927/h4jq0xzw).
- [7] NOAA's National Geophysical Data CENTER et US Air Force Weather AGENCY. *Version 1 VIIRS Day/Night Band Nighttime Lights*. URL : https://ngdc.noaa.gov/eog/viirs/download_dnb_composites.html (visité le 15/07/2018).
- [8] NOAA's National Geophysical Data CENTER et US Air Force Weather AGENCY. *Version 4 DMSP-OLS Nighttime Lights Time Series*. URL : <https://ngdc.noaa.gov/eog/dmsp/downloadV4composites.html> (visité le 14/06/2018).
- [9] Patrick DOUPE et al. "Equitable development through deep learning". In : *Proceedings of the 7th Annual Symposium on Computing for Development - ACM DEV '16*. ACM Press, 2016. DOI : [10.1145/3001913.3001921](https://doi.org/10.1145/3001913.3001921). URL : <https://doi.org/10.1145/3001913.3001921>.
- [10] Christoph GOHLKE. *Unofficial Windows Binaries for Python Extension Packages*. URL : <https://www.lfd.uci.edu/~gohlke/pythonlibs/> (visité le 09/07/2018).
- [11] Natural Earth. *Admin 0 – Countries*. Version 4.0.0. Natural Earth. 21 mar. 2018. URL : <https://www.naturalearthdata.com/downloads/10m-cultural-vectors/10m-admin-0-countries/> (visité le 07/06/2018).
- [12] World Development Indicators. Version version. note. The World Bank. 12 oct. 2016. URL : <https://data.worldbank.org/> (visité le 07/06/2018).

- [13] *World Population Prospects 2017*. Version Révision de 2017. United Nations Department of Economic and Social Affairs, Population Division. 7 déc. 2017. URL : <https://esa.un.org/unpd/wpp/> (visité le 23/03/2018).
- [14] *Worldview*. NASA EOSDIS. URL : <https://worldview.earthdata.nasa.gov/> (visité le 06/06/2018).

8 Authentification

Par la présente, je soussigné, Antoine Friant, déclare avoir réalisé seul ce travail et ne pas avoir utilisé d'autres sources que celles citées dans la bibliographie.

Date

Signature

Nom complet

9 Symboles et abréviations

Table des figures

5.1	Outil de visualisation NASA Worldview [14].	4
5.2	Image satellite quotidienne servie par NASA Worldview [14], représentant la Grande Bretagne et son climat nuageux.	5
5.3	Une tuile de l'image de 2016 montrant la ville de Dallas (USA) après avoir été mise en couleurs négatives.	5
5.4	Image globale annuelle (2016) reconstituée à partir de tuiles téléchargées, puis mise en couleurs négatives.	6
5.5	Extrait de la grille de population [6] rendu avec QGIS. Le blanc indique une absence d'habitants, le noir indique au moins 1000 habitants par kilomètre carré.	8
5.6	Quantité de lumière perçue depuis l'espace, population et PIB de la France entre 1992 et 2013.	10
5.7	Quantité de lumière perçue depuis l'espace, population et PIB de la Chine entre 1992 et 2013.	11
5.8	Quantité de lumière perçue depuis l'espace, population et PIB du Japon entre 1992 et 2013.	11
5.9	Quantité de lumière perçue depuis l'espace, population et PIB de l'Arménie entre 1992 et 2013.	12
5.10	Pays placés par population et luminosité totale émise sur une échelle logarithmique en 2013, colorés par indice économique.	13
5.11	Pays placés par PIB en USD et luminosité totale émise sur une échelle logarithmique en 2013, colorés par indice économique.	13
5.12	Pays placés par consommation en électricité en millions de kWh et luminosité totale émise sur une échelle logarithmique en 2013, colorés par indice économique.	14
5.13	Superposition de l'image satellite nocturne (bleu ciel) et de la grille de population (rose).	15
5.14	Effet de l'augmentation de la résolution de la grille de population lors de la fusion	16
5.15	Valeur de luminosité et de population mondiale pour chaque km ² (année 2000)	16

5.16 Valeur de luminosité et de population pour chaque 0.25km^2 du Brésil (année 2015)	17
6.1 Topologie de la toute première version du réseau de neurones.	20
6.2 Moyennes des erreurs absolues sur 100 itérations.	20
6.3 Moyennes des erreurs au carré (fonction objectif) sur 100 itérations.	21
6.4 Grille de population pour la frontière nord entre l'Espagne et le Portugal illustrant les différences de qualité des données de population d'un pays à l'autre.	22