



HAUTE ÉCOLE  
D'INGÉNIERIE ET DE GESTION  
DU CANTON DE VAUD  
[www.heig-vd.ch](http://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](mailto:antoine.friant@gmail.com)

16 juillet 2018

# Table des matières

<b>1 Cahier des charges</b>	<b>iii</b>
1.1 Résumé du problème . . . . .	iii
1.2 Objectifs . . . . .	iii
1.3 Limitations . . . . .	iv
1.4 Description fonctionnelle . . . . .	iv
1.5 Délais . . . . .	iv
<b>2 Introduction</b>	<b>1</b>
<b>3 Gleam</b>	<b>2</b>
<b>4 Exploration des données</b>	<b>3</b>
4.1 Jeux de données . . . . .	3
4.1.1 Images satellite . . . . .	3
4.1.2 Grilles de population . . . . .	6
4.1.3 Pays . . . . .	6
4.2 Recherche de corrélation . . . . .	8
4.2.1 Pays . . . . .	8
4.2.2 Grille de population . . . . .	13
4.3 Données à explorer . . . . .	16
<b>5 Modèle</b>	<b>17</b>
5.1 Environnement de développement . . . . .	17
5.2 Réseau de neurones . . . . .	17

5.3	Première topologie . . . . .	18
5.4	Premiers résultats . . . . .	18
5.5	Améliorations . . . . .	20
5.5.1	Pré-traitement . . . . .	20
5.5.2	Topologie . . . . .	20
5.6	Réseau final . . . . .	20
5.7	Résultats . . . . .	20
<b>6</b>	<b>Conclusion</b>	<b>22</b>
<b>7</b>	<b>Authentification</b>	<b>24</b>
<b>8</b>	<b>Symboles et abréviations</b>	<b>25</b>

# 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 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 C0<sub>2</sub>, 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.

## 3 Gleam

Le projet Gleam ("lueur") se présente comme un ensemble de notebooks Jupyter.

# 4 Exploration des données

## 4.1 Jeux de données

### 4.1.1 Images satellite

#### NASA Worldview

La première source de données explorée est l'application "Worldview" de la NASA [8]. Elle permet de visionner un grand nombre d'images satellite composites sur un globe en trois dimensions (voir figure 4.1). Parmi les jeux de données disponibles sont trois jeux d'images nocturnes.



FIGURE 4.1 – Outil de visualisation NASA Worldview [8].

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.

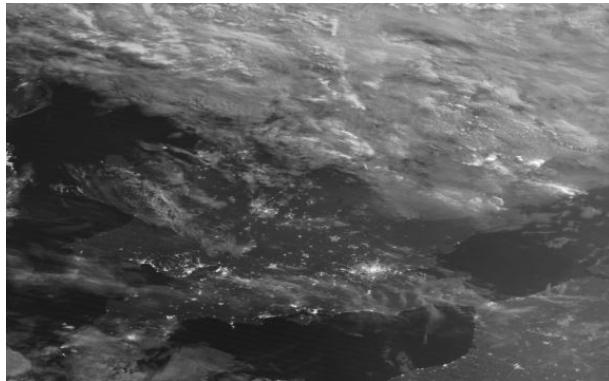


FIGURE 4.2 – Image satellite quotidienne servie par NASA Worldview [8], représentant la Grande Bretagne et son climat nuageux.

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 4.3) pour reconstituer une image complète du globe de plus de 800 millions de pixels (figure 4.4).

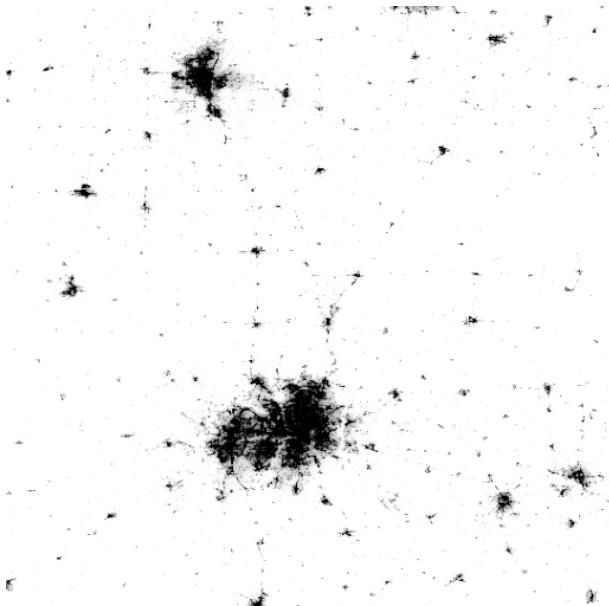


FIGURE 4.3 – Une tuile de l'image de 2016 montrant la ville de Dallas (USA) après avoir été mise en couleurs négatives.

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 exécution peut demander plus d'une heure pour le téléchargement (vitesse limitée par le serveur),



FIGURE 4.4 – Image globale annuelle (2016) reconstituée à partir de tuiles téléchargées, puis mise en couleurs négatives.

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 [3] (1km par pixel, unités arbitraire de 0 à 63) et 2012 à 2018 [2] (mensuelles, 500m par pixel, radiance en nanoWatts/cm<sup>2</sup>/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.

### 4.1.2 Grilles de population

Sedac [1] met à disposition des grilles de populations pour le monde entier, sous forme de fichier GeoTIFF. Chaque "case" de 1 km<sup>2</sup> 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 4.5).

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.

### 4.1.3 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 : 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 [5] contient les frontières des pays actuelles, ainsi que des méta-données sur chacun de ces pays (population estimée, indice



FIGURE 4.5 – Extrait de la grille de population [1] rendu avec QGIS. Le blanc indique une absence d'habitants, le noir indique au moins 1000 habitants par kilomètre carré.

de développement, différentes appellations et abréviations, etc.). En cas de conflits ou ambiguïté politique sur les frontières, c'est le pays qui contrôle le terrain qui est marqué comme souverain. Voici à quoi ressemblent les métadonnées 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 CO<sub>2</sub>, 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 [7], 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 [6] et téléchargée depuis [http://data.un.org/Data.aspx?q=gdp&d=WDI&f=Indicator\\_Code%3aNY.GDP.MKTP.CD](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.

## 4.2 Recherche de corrélation

### 4.2.1 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 travail

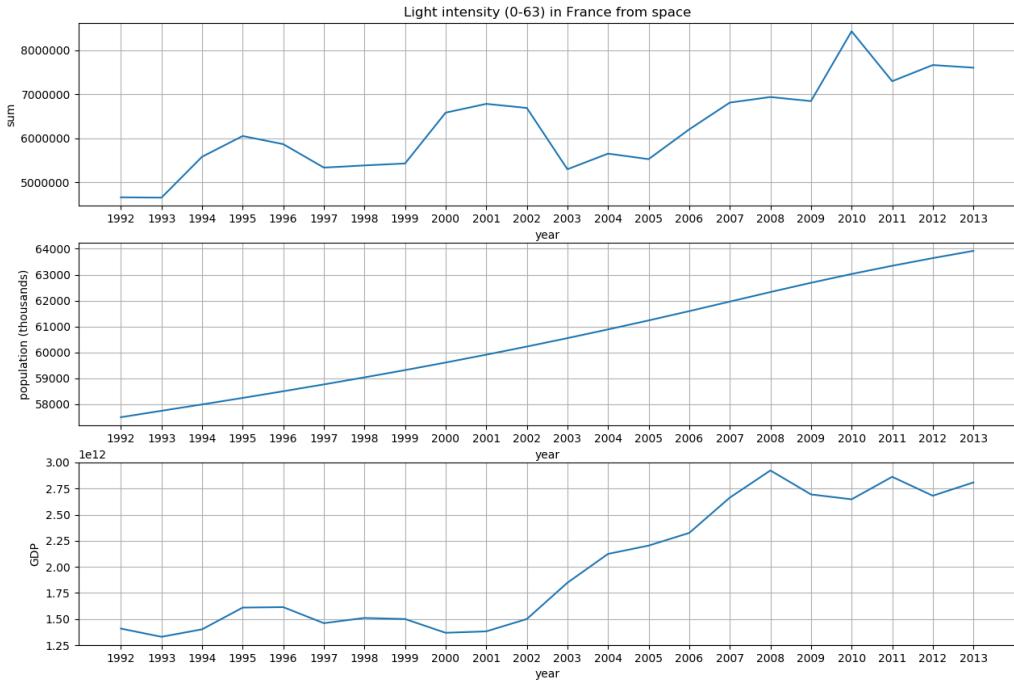


FIGURE 4.6 – Quantité de lumière perçue depuis l'espace, population et PIB de la France entre 1992 et 2013.

sont la population par pays [7], le produit intérieur brut [6], 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 [5] et des images de la NOAA [3]).

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 4.6, 4.7, 4.8 et 4.9). 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 4.6 est typique des résultats obtenus. Elle laisse présager une forte corrélation entre les trois données observées. Cependant, la figure 4.7 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 4.8) semblent indiquer que la luminosité suit le PIB également lorsque ce dernier diminue. Enfin le cas de l'Arménie (figure 4.9) 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 [5]). Ces graphes sont

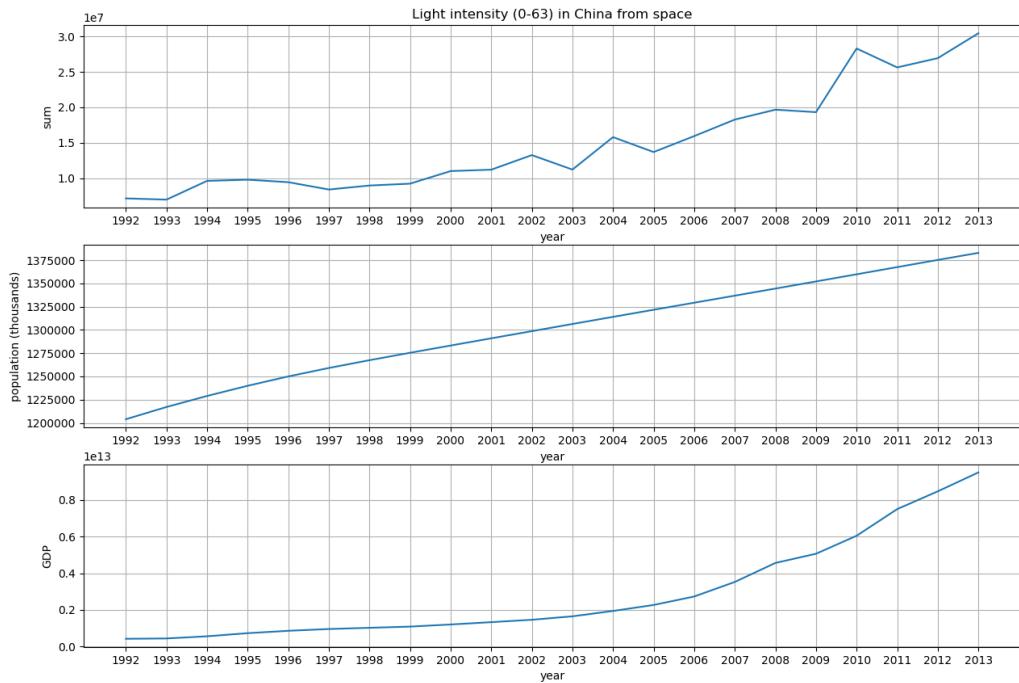


FIGURE 4.7 – Quantité de lumière perçue depuis l'espace, population et PIB de la Chine entre 1992 et 2013.

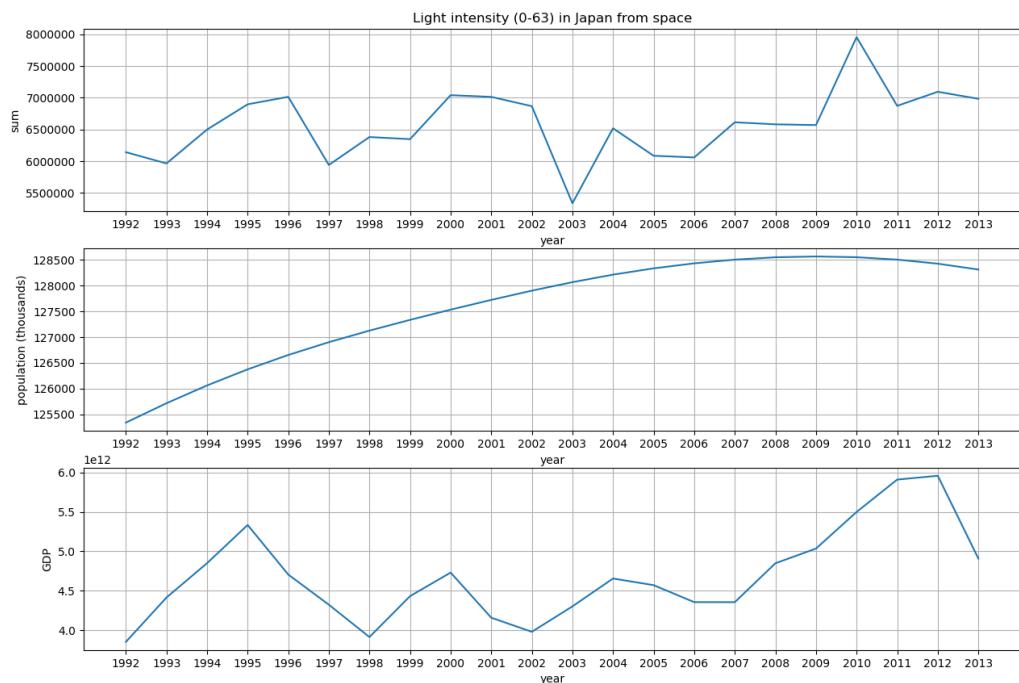


FIGURE 4.8 – Quantité de lumière perçue depuis l'espace, population et PIB du Japon entre 1992 et 2013.

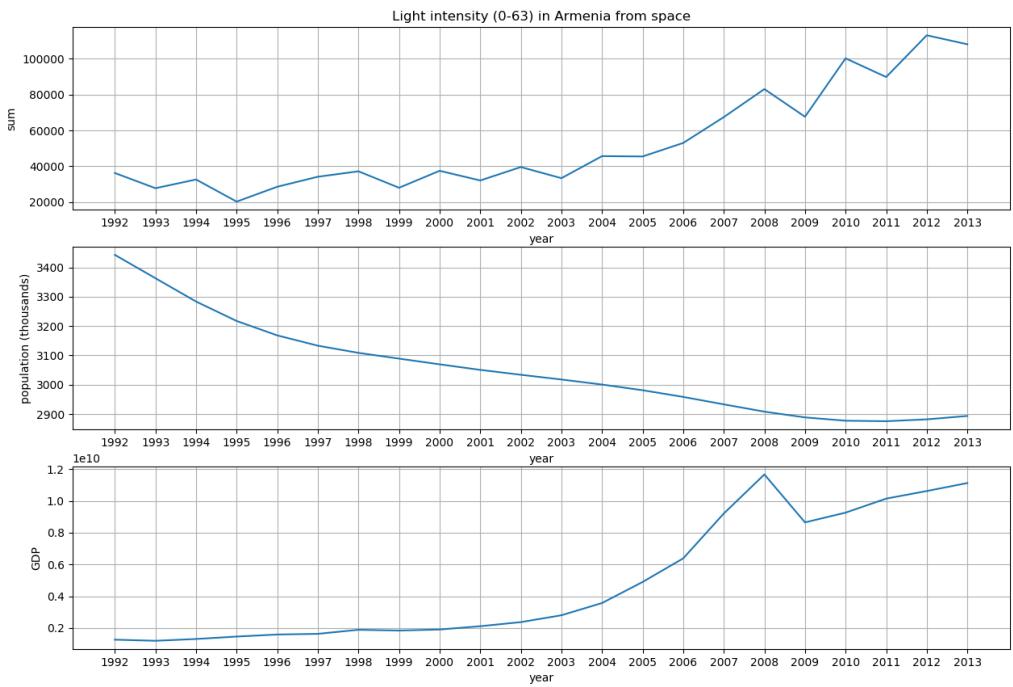


FIGURE 4.9 – Quantité de lumière perçue depuis l'espace, population et PIB de l'Arménie entre 1992 et 2013.

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 4.10). 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 4.11). 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 4.12. 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.

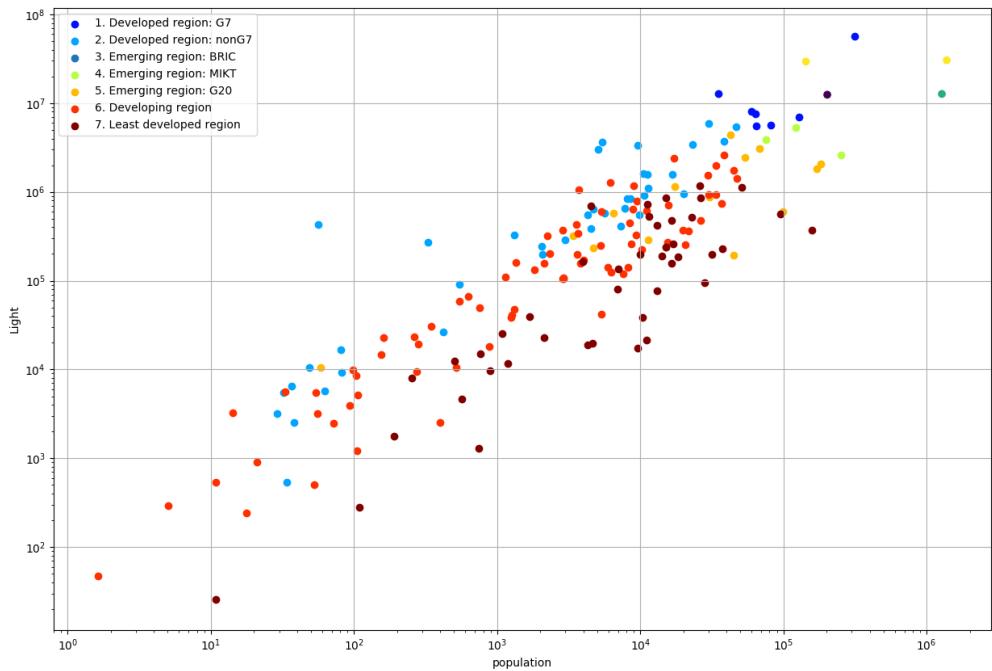


FIGURE 4.10 – Pays placés par population et luminosité totale émise sur une échelle logarithmique, colorés par indice économique.

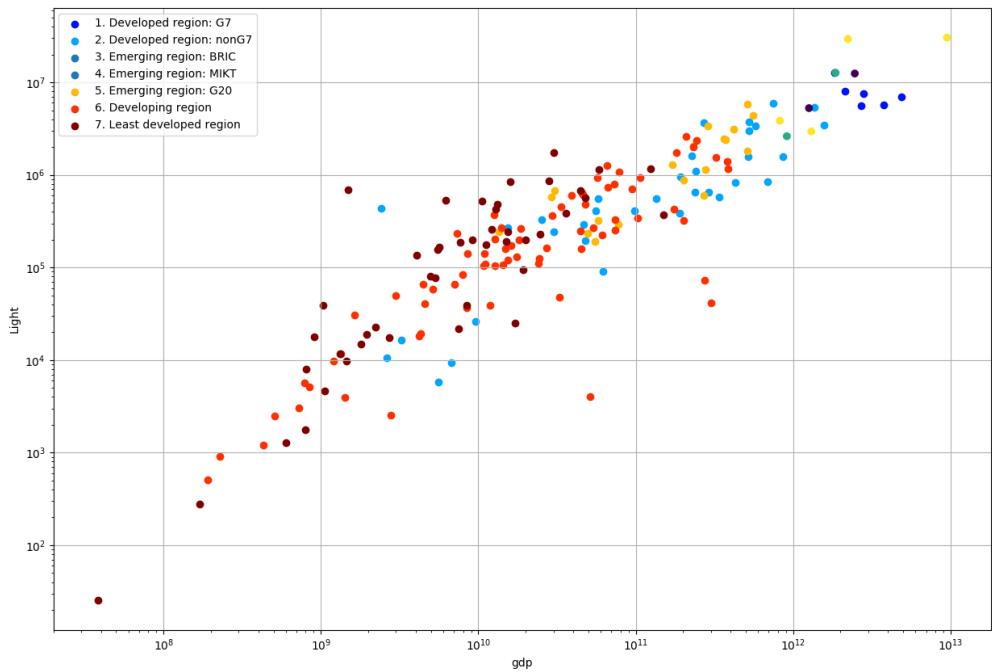


FIGURE 4.11 – Pays placés par PIB en USD et luminosité totale émise sur une échelle logarithmique, colorés par indice économique.

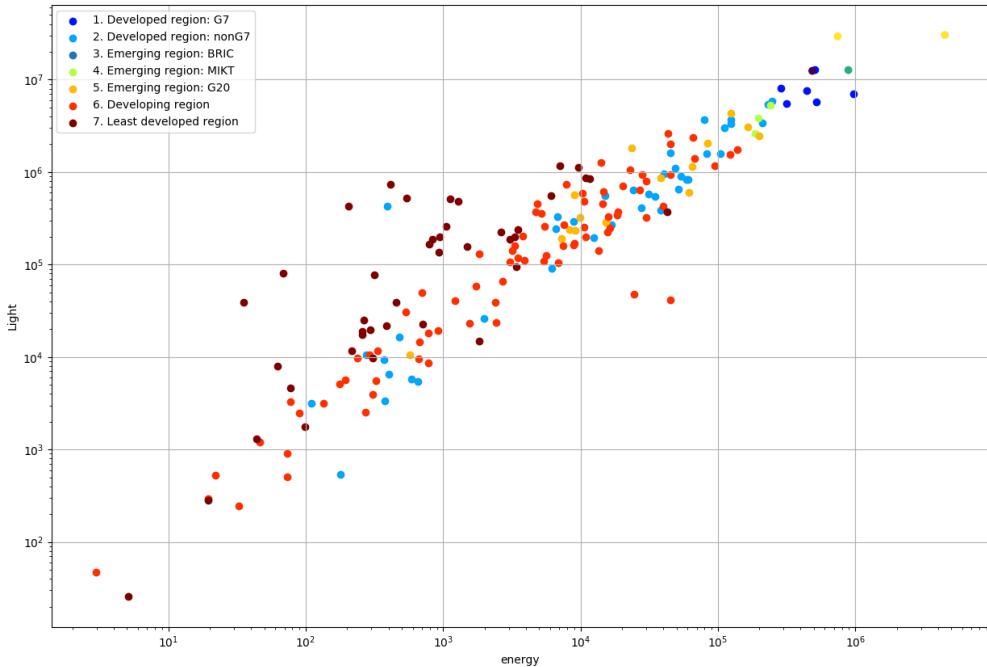


FIGURE 4.12 – Pays placés par consommation en électricité en millions de kWh et luminosité totale émise sur une échelle logarithmique, colorés par indice économique.

#### 4.2.2 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 [3]). Après avoir ajusté ses propriétés d'affichage et inversé les couleurs, on obtient la figure 4.13. Le bleu ciel correspond à la lumière visible depuis l'espace, le rose la population (sur une échelle de 1 à 1000 habitants par km<sup>2</sup>). 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 [2] 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 4.13 – Superposition de l'image satellite nocturne (bleu ciel) et de la grille de population (rose).

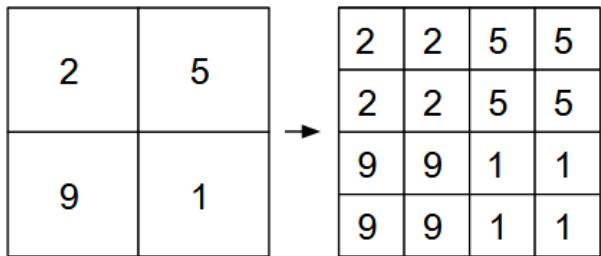


FIGURE 4.14 – Effet de l'augmentation de la résolution de la grille de population lors de la fusion

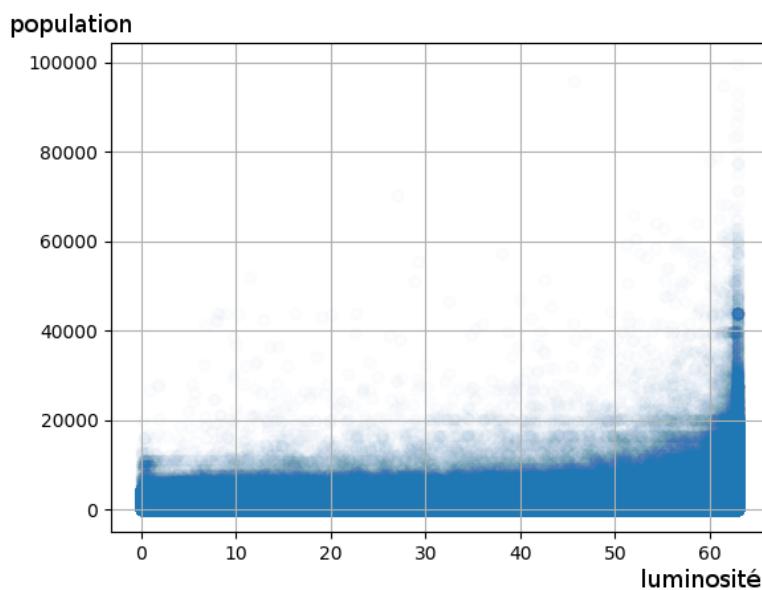


FIGURE 4.15 – Valeur de luminosité et de population mondiale pour chaque km<sup>2</sup> (année 2000)

Chaque pixel donne donc correctement la densité de population par kilomètre carré, mais la population totale est multipliée par 4. La figure 4.14 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 4.15. La figure 4.16 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 [5] 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".

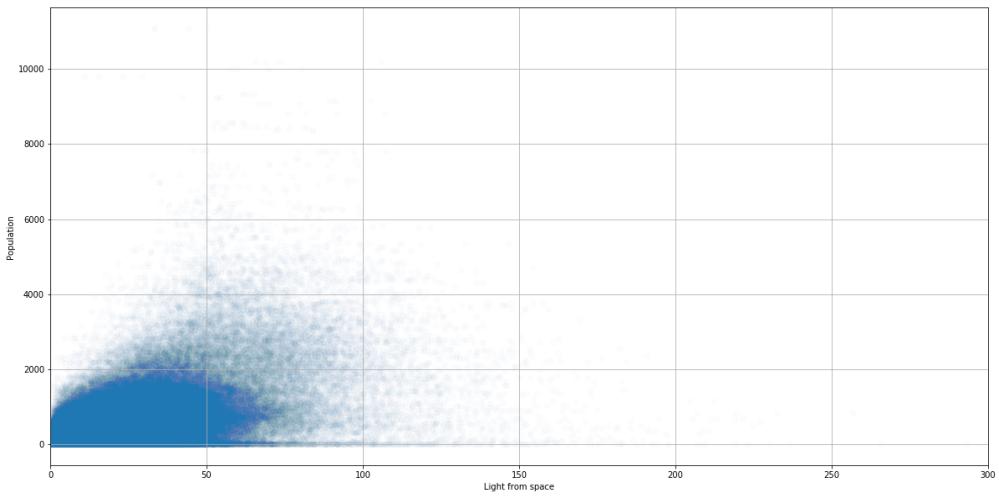


FIGURE 4.16 – Valeur de luminosité et de population pour chaque  $0.25\text{km}^2$  du Brésil (année 2015)

### 4.3 Données à explorer

Parmi les données potentiellement intéressantes qui n'ont pas encore été explorées dans ce travail, il y a notamment des grilles pour :

- l'impact humain sur l'environnement,
- l'estimation des populations haute résolution de Facebook (digital globe),
- les changements de luminosité d'une année à l'autre (à calculer à partir des images satellites),
- la couverture du territoire (vert ou bétonné),

Toutes les sources de données satellite n'ont par ailleurs pas été explorées. NOAA dispose d'autres images composites, plus récentes et exposant d'autres fréquences du spectre lumineux.

# 5 Modèle

## 5.1 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](https://www.tensorflow.org/install/install_linux)).

## 5.2 Réseau de neurones

Pour commencer à développer le réseau de neurones, on commencera par tenter de prédire la population d'une petite région (une tuile) à partir de l'image satellite de l'année 2005, en entraînant la machine sur celle de 2000 et la grille de population correspondante [1].

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.

Avant de commencer l'entraînement, on découpe l'image satellite nocturne en tuiles de 32 sur 32 pixels ( $1024 \text{ km}^2$ ). Chacune de ces tuiles sera considérée 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.

### 5.3 Première topologie

La topologie du réseau (figure 5.1) s'inspire grossièrement du travail effectué par l'Arnold Institute for Global Health [4], 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  $3 \times 3$  pixels, fonction d'activation ReLU,
- MaxPooling 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.

### 5.4 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 5.2 et 5.3 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  $1024 \text{ km}^2$ , la prédiction du modèle se trompe en moyenne de 26599.64 habitants. C'est moins bon que tous les résultats obtenus pendant l'entraînement. Cela signifie que le modèle apprend (car l'erreur diminue lors de l'entraînement), mais il n'apprend rien de généralisable.

Améliorations possibles :

- Réduction de la taille des tuiles ( $1024 \text{ km}^2 \rightarrow 256 \text{ km}^2$ ),
- Ajout d'une couche cachée supplémentaire,
- Pondérer les tuiles par nombre d'habitants durant l'entraînement (et supprimer les tuiles qui sont entièrement dans l'océan) [4],
- Pondérer les tuiles selon la qualité des données de population (qui est également une grille),
- Augmenter le nombre de neurones de sortie pour correspondre à une grille de population plutôt qu'une somme sur tous les pixels de la région.

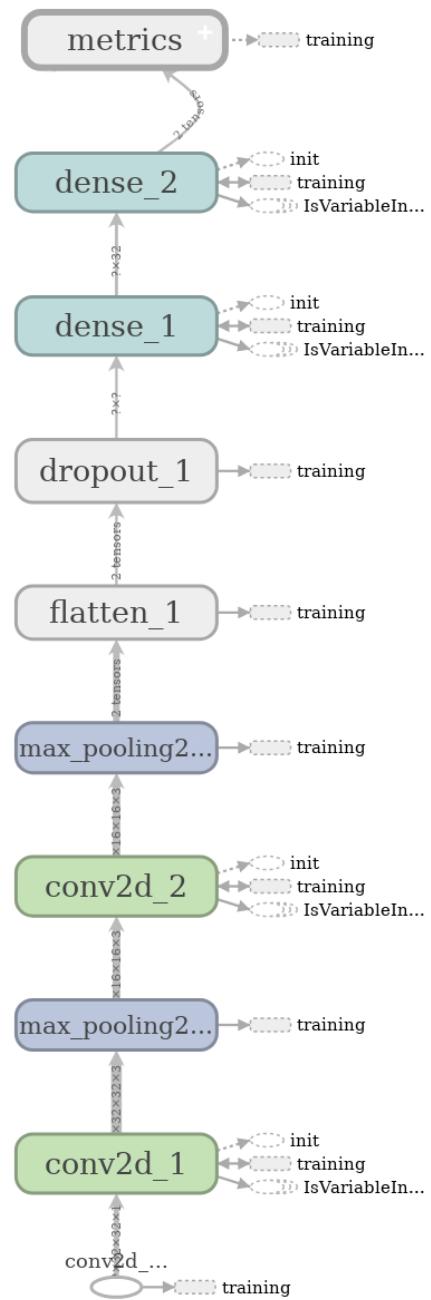


FIGURE 5.1 – Topologie de la toute première version du réseau de neurones.

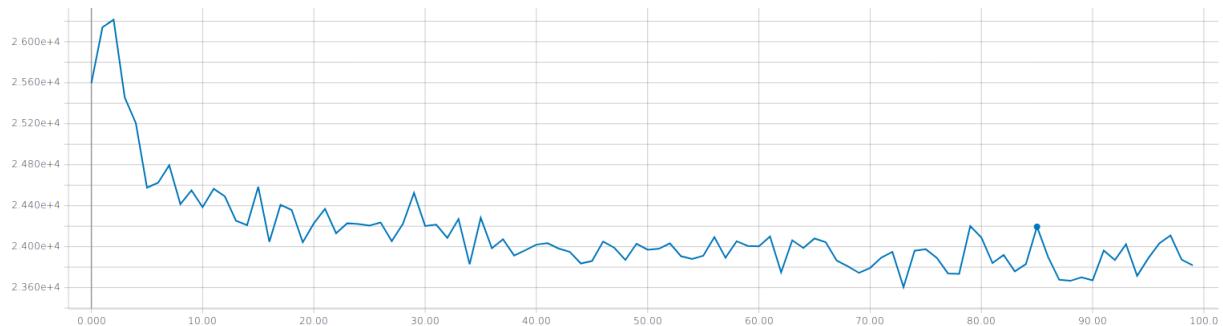


FIGURE 5.2 – Moyennes des erreurs absolues sur 100 itérations.

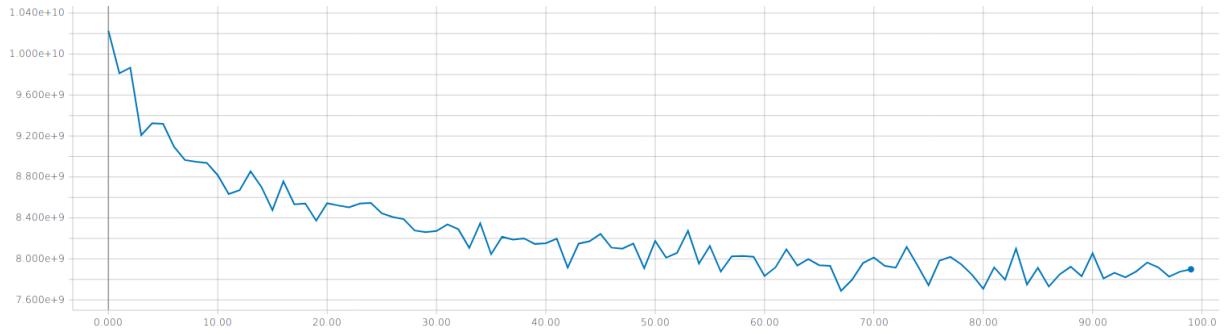


FIGURE 5.3 – Moyennes des erreurs au carré (fonction objectif) sur 100 itérations.

## 5.5 Améliorations

### 5.5.1 Pré-traitement

Dataset : trop grand, divers

Mauvaise qualité de la résolution pour la plupart des régions

=> Restriction à 1 pays ayant de bonnes données => peu de données => moving window in preprocess => trop de données => ignore empty tiles

plus petites tuiles => mauvais résultats

### 5.5.2 Topologie

inputs pondérés par la population => mauvais résultats

runs\_history pour les tests de topologies

## 5.6 Réseau final

description du réseau final

## 5.7 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 prédition, entrainé *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 Conclusion

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.

# Bibliographie

- [1] 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).
- [2] 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](https://ngdc.noaa.gov/eog/viirs/download_dnb_composites.html) (visité le 15/07/2018).
- [3] 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).
- [4] 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>.
- [5] *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).
- [6] *World Development Indicators.* Version version. note. The World Bank. 12 oct. 2016. URL : <https://data.worldbank.org/> (visité le 07/06/2018).
- [7] *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).
- [8] *Worldview.* NASA EOSDIS. URL : <https://worldview.earthdata.nasa.gov/> (visité le 06/06/2018).

## 7 Authentication

## 8 Symboles et abréviations

# Table des figures

4.1	Outil de visualisation NASA Worldview [8]. . . . .	3
4.2	Image satellite quotidienne servie par NASA Worldview [8], représentant la Grande Bretagne et son climat nuageux. . . . .	4
4.3	Une tuile de l'image de 2016 montrant la ville de Dallas (USA) après avoir été mise en couleurs négatives. . . . .	4
4.4	Image globale annuelle (2016) reconstituée à partir de tuiles téléchargées, puis mise en couleurs négatives. . . . .	5
4.5	Extrait de la grille de population [1] rendu avec QGIS. Le blanc indique une absence d'habitants, le noir indique au moins 1000 habitants par kilomètre carré.	7
4.6	Quantité de lumière perçue depuis l'espace, population et PIB de la France entre 1992 et 2013. . . . .	9
4.7	Quantité de lumière perçue depuis l'espace, population et PIB de la Chine entre 1992 et 2013. . . . .	10
4.8	Quantité de lumière perçue depuis l'espace, population et PIB du Japon entre 1992 et 2013. . . . .	10
4.9	Quantité de lumière perçue depuis l'espace, population et PIB de l'Arménie entre 1992 et 2013. . . . .	11
4.10	Pays placés par population et luminosité totale émise sur une échelle logarithmique, colorés par indice économique. . . . .	12
4.11	Pays placés par PIB en USD et luminosité totale émise sur une échelle logarithmique, colorés par indice économique. . . . .	12
4.12	Pays placés par consommation en électricité en millions de kWh et luminosité totale émise sur une échelle logarithmique, colorés par indice économique. . . . .	13
4.13	Superposition de l'image satellite nocturne (bleu ciel) et de la grille de population (rose). . . . .	14
4.14	Effet de l'augmentation de la résolution de la grille de population lors de la fusion	15
4.15	Valeur de luminosité et de population mondiale pour chaque km <sup>2</sup> (année 2000) . .	15

4.16 Valeur de luminosité et de population pour chaque $0.25\text{km}^2$ du Brésil (année 2015)	16
5.1 Topologie de la toute première version du réseau de neurones.	19
5.2 Moyennes des erreurs absolues sur 100 itérations.	19
5.3 Moyennes des erreurs au carré (fonction objectif) sur 100 itérations.	20