

ÉCOLE NATIONALE DES CHARTES

Jade Norindr

licenciée ès lettres

Le traitement des sources historiques par la vision artificielle

L'exemple des manuscrits d'astronomie de
tradition ptoléméenne

Mémoire pour le diplôme de master
« Technologies numériques appliquées à l'histoire »

2023

Résumé

Résumé

Mots-clés : histoire des sciences ; diagrammes astronomiques ; images ; *machine learning* ; vision artificielle ; apprentissage profond ; YOLOv5 ; réseau de neurones artificiels ; automatisation ; Python ; programmation modulaire ; IIIF ; API ; Flask.

Informations bibliographiques : Jade Norindr, *Le traitement des sources historiques par la vision artificielle. L'exemple des manuscrits d'astronomie de tradition ptoléméenne*, mémoire de master « Technologies numériques appliquées à l'histoire », dir. Maxime Challon, École nationale des chartes, 2023.

Remerciements

Bibliographie

- API de La RMN-Grand Palais*, URL : <https://api.art.rmnngp.fr/> (visité le 10/08/2023).
- ARSLAN (Görkem), *Create a Machine Learning API With Django Rest Framework / Better Programming*, juill. 2021, URL : <https://betterprogramming.pub/create-a-machine-learning-api-with-django-rest-framework-967571640c46> (visité le 27/08/2023).
- AZENCOTT (Chloé-Agathe), *Introduction au Machine Learning*, Malakoff, 2018.
- BAUJARD (Corinne), « Numérisation du patrimoine culturel et stratégie managériale des musées », *Management des technologies organisationnelles*-7 (févr. 2017), p. 69-78, URL : <https://www.cairn.info/revue-management-des-technologies-organisationnelles-2017-2-page-69.htm> (visité le 06/08/2023).
- BENHAMOU (Yaniv), « Droit d'auteur et musées numériques », *Magazine de l'OMPI*-3 (juin 2016), URL : https://www.wipo.int/wipo_magazine/fr/2016/03/article_0005.html (visité le 06/08/2023).
- BERGSTROM (Austin C.) et MESSINGER (David W.), « Image Quality and Computer Vision Performance : Assessing the Effects of Image Distortions and Modeling Performance Relationships Using the General Image Quality Equation », *Journal of Electronic Imaging*, 32 (mars 2023), p. 023018, DOI : [10.1117/1.JEI.32.2.023018](https://doi.org/10.1117/1.JEI.32.2.023018).
- BESSE (Camille), *Numérisation de Masse : Vers La Création d'un Nouvel Acteur de l'information. Le Projet Time Machine*, Paris, 2019.
- BÜTTNER (Jochen), MARTINETZ (Julius), EL-HAJJ (Hassan) et VALLERIANI (Matteo), « CorDeep and the Sacrobosco Dataset : Detection of Visual Elements in Historical Documents », *Journal of Imaging*, 8-10 (oct. 2022), p. 285, DOI : [10.3390/jimaging8100285](https://doi.org/10.3390/jimaging8100285).
- CARREMANS (Bert), *Handling Overfitting in Deep Learning Models*, janv. 2019, URL : <https://towardsdatascience.com/handling-overfitting-in-deep-learning-models-c760ee047c6e> (visité le 22/08/2023).
- Category :Google Art Project Works by Collection - Wikimedia Commons*, URL : https://commons.wikimedia.org/wiki/Category:Google_Art_Project_works_by_collection (visité le 10/08/2023).
- Celery - Distributed Task Queue — Celery 5.3.1 Documentation*, URL : <https://docs.celeryq.dev/en/stable/> (visité le 29/08/2023).

- CHAGUÉ (Alix) et CLÉRICE (Thibault), *HTR-United*, URL : <https://htr-united.github.io/index.html> (visité le 10/08/2023).
- CHOLLET (François), *L'apprentissage profond avec Python*, Saint-Cyr-sur-Loire, 2020.
- CLAERR (Thierry) et WESTEEL (Isabelle), « Mode d'emploi », dans *Numériser et mettre en ligne*, Villeurbanne, 2017 (La Boîte à outils), p. 7-18, DOI : 10.4000/books.pressesenssib.419.
- COCO - *Common Objects in Context*, URL : <https://cocodataset.org/#home> (visité le 15/08/2023).
- CorDeep, URL : <https://cordeep.mpiwg-berlin.mpg.de/> (visité le 10/08/2023).
- Corpus, URL : <https://vhs.hypotheses.org/corpus> (visité le 09/08/2023).
- CRAMER (Tom), *The International Image Interoperability Framework (IIIF) : Laying the Foundation for Common Services, Integrated Resources and a Marketplace of Tools for Scholars Worldwide*, déc. 2011, URL : <https://www.cni.org/topics/information-access-retrieval/international-image-interoperability-framework> (visité le 08/08/2023).
- « cron », *Wikipédia* (, juill. 2023), URL : <https://fr.wikipedia.org/w/index.php?title=Cron&oldid=206272232> (visité le 29/08/2023).
- Darknet : *Open Source Neural Networks in C*, URL : <https://pjreddie.com/darknet/> (visité le 18/08/2023).
- DENOYELLE (Martine), *A propos du coût des images d'œuvres patrimoniales*, Billet, juin 2021, URL : <https://numrha.hypotheses.org/2142> (visité le 06/08/2023).
- *En savoir plus sur la licence Etalab et les autres licences*, Billet, juin 2021, URL : <https://numrha.hypotheses.org/2134> (visité le 06/08/2023).
- Digital Images, 2020, URL : <https://livecode.byu.edu/images/DigitalImages.php> (visité le 07/08/2023).
- Django, URL : <https://www.djangoproject.com/> (visité le 27/08/2023).
- Django REST Framework, URL : <https://www.django-rest-framework.org/> (visité le 27/08/2023).
- DOWLING (Jim), *Guide to File Formats for Machine Learning : Columnar, Training, Inferencing, and the Feature Store*, oct. 2019, URL : <https://towardsdatascience.com/guide-to-file-formats-for-machine-learning-columnar-training-inferencing-and-the-feature-store-2e0c3d18d4f9> (visité le 29/08/2023).
- EdIter et analyser les Diagrammes astronomiques historiques avec l'intelligence Artificielle, URL : <https://anr.fr/Projet-ANR-22-CE38-0014> (visité le 10/08/2023).
- Enhancing Heritage Image Databases, URL : <https://anr.fr/Project-ANR-17-CE23-0008> (visité le 10/08/2023).
- Europeana, URL : <https://www.europeana.eu/fr> (visité le 10/08/2023).
- EVANS (James), « History of Astronomy », dans *Encyclopedia Britannica*, URL : <https://www.britannica.com/science/astronomy/Ancient-Greece> (visité le 03/08/2023).

- FastAPI*, URL : <https://fastapi.tiangolo.com/> (visité le 27/08/2023).
- FastAPI, le framework python alternatif à Django et Flask ?*, juin 2023, URL : <https://www.agencedebord.com/blog/fastapi-le-framework-python-alternatif-a-django-et-flask> (visité le 27/08/2023).
- Flask Documentation (2.3.x)*, URL : <https://flask.palletsprojects.com/en/2.3.x/> (visité le 27/08/2023).
- Gallica*, URL : <https://gallica.bnf.fr/> (visité le 10/08/2023).
- GOODFELLOW (Ian), BENGIO (Yoshua) et COURVILLE (Aaron), *Deep Learning*, 2016, URL : <http://www.deeplearningbook.org>.
- Google Arts & Culture*, URL : <https://artsandculture.google.com/> (visité le 10/08/2023).
- GRINBERG (Miguel), *RESTful Authentication with Flask*, URL : <http://blog.miguelgrinberg.com/post/restful-authentication-with-flask> (visité le 27/08/2023).
- Jeffrey F. Hamburger, David J. Roxburgh et Linda Safran (éd.), *The Diagram as Paradigm : Cross-Cultural Approaches*, Cambridge, MA, 2022 (Dumbarton Oaks Byzantine Symposia and Colloquia).
- Historical Navigation — DISHAS Project — Astronomical Table in Database*, URL : <https://dishes.observatoire.fr/historical-navigation> (visité le 10/08/2023).
- Home — DISHAS Project — Astronomical Table in Database*, URL : <https://dishes.observatoire.fr/> (visité le 10/08/2023).
- How It Works*, URL : <https://iiif.io/get-started/how-iiif-works/> (visité le 08/08/2023).
- IIIF for Museums in France*, juill. 2023, URL : <https://www.culture.gouv.fr/en/Thematic/Museums/Pour-les-professionnels/Network/IIIF-for-museums-in-France> (visité le 09/08/2023).
- IIIF Viewers*, URL : <https://iiif.io/get-started/iiif-viewers/> (visité le 08/08/2023).
- Image API 3.0*, URL : <https://iiif.io/api/image/3.0/> (visité le 08/08/2023).
- ImageNet*, URL : <https://www.image-net.org/> (visité le 10/08/2023).
- INSTITUT NATIONAL D'HISTOIRE DE L'ART, *Guide Pratique Pour La Recherche et La Réutilisation Des Images d'œuvres d'art*, URL : https://www.inha.fr/_resources/PDF/2021/2021_LIVRET_GUIDE-PRATIQUE_web_15062021.pdf.
- JACQUOT (Olivier), *Décrire, transcrire et diffuser un corpus documentaire hétérogène : méthodes, formats, outils*, Billet, nov. 2017, URL : <https://bnf.hypotheses.org/2214> (visité le 06/08/2023).
- JARDINE (Boris) et JARDINE (Nicholas), « Critical Editing of Early-Modern Astronomical Diagrams », *Journal for the History of Astronomy*, 41–3 (août 2010), p. 393-414, DOI : [10.1177/002182861004100307](https://doi.org/10.1177/002182861004100307).
- JONES (Alexander Raymond), « Ptolemy », dans *Encyclopedia Britannica*, URL : <https://www.britannica.com/biography/Ptolemy> (visité le 03/08/2023).

- KLINKE (Harald), « Big Image Data within the Big Picture of Art History », *International Journal for Digital Art History*–2 (oct. 2016), DOI : 10.11588/dah.2016.2.33527.
- LIN (Xinyu), ZHOU (Yingjie), LIU (Yipeng) et ZHU (Ce), *A Comprehensive Review of Image Line Segment Detection and Description : Taxonomies, Comparisons, and Challenges*, avr. 2023, DOI : 10.48550/arXiv.2305.00264, arXiv : 2305.00264 [cs].
- MALLORY (Gavin), *IIIF for Museums, Explained*, juill. 2019, URL : <https://blog.cogapp.com/iiif-for-museums-explained-49fd0560e1ba> (visité le 09/08/2023).
- MANCA (Isabelle), « Les nouveaux défis des agences photographiques des musées », *Le Journal Des Arts* (, juin 2018), URL : <https://www.lejournaldesarts.fr/patrimoine/les-nouveaux-defis-des-agences-photographiques-des-musees-138076> (visité le 06/08/2023).
- MOIRAGHI (Eleonora) et MOREUX (Jean-Philippe), *Explorer des corpus d'images. L'IA au service du patrimoine*, Billet, avr. 2018, URL : <https://bnf.hypotheses.org/2809> (visité le 06/08/2023).
- MONNIER (Tom) et AUBRY (Mathieu), « docExtractor : An off-the-Shelf Historical Document Element Extraction », dans *2020 17th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 2020, p. 91-96, DOI : 10.1109/ICFHR2020.2020.00027, arXiv : 2012.08191 [cs].
- MONNIER (Tom), VINCENT (Elliot), PONCE (Jean) et AUBRY (Mathieu), *Unsupervised Layered Image Decomposition into Object Prototypes*, août 2021, DOI : 10.48550/arXiv.2104.14575, arXiv : 2104.14575 [cs].
- NORINDR (Jade), *Element Extraction on a GPU*, juin 2023, URL : <https://github.com/jnorindr/extractorAPI> (visité le 26/08/2023).
- Presentation API 3.0*, URL : <https://iiif.io/api/presentation/3.0/> (visité le 08/08/2023).
- PyTorch*, URL : <https://pytorch.org/> (visité le 18/08/2023).
- RICHARD (Michel), « Le programme de numérisation de la Bibliothèque de France », *Bulletin des bibliothèques de France*–3 (1993), p. 53-63, URL : <https://bbf.enssib.fr/consulter/bbf-1993-03-0053-007> (visité le 06/08/2023).
- RONSE (Christian), *Invariance Par Translations*, URL : <https://dpt-info.u-strasbg.fr/~cronse/TIDOC/SYM/invtr.html> (visité le 15/08/2023).
- ROUSSEAU (Jean-Jacques), *Epicycles de Ptolémée*, URL : <http://ressources.univ-lemans.fr/AccesLibre/UM/Pedago/physique/02/divers/ptolemee.html> (visité le 05/08/2023).
- SAKAI (Akira), *Digitizing Disparity in the Museum. The Object-Based Research in the Tokyo National Museum*, août 2021, URL : <https://universityheritage.eu/en/digitizing-disparity-in-the-museum-the-object-based-research-in-the-tokyo-national-museum/> (visité le 06/08/2023).

- SANDY (James), *Choosing between Django, Flask, and FastAPI*, URL : <https://www.section.io/engineering-education/choosing-between-django-flask-and-fastapi/> (visité le 27/08/2023).
- SEPETJAN (Sophie), « Respecter le droit de la propriété littéraire et artistique », dans *Numériser et mettre en ligne*, dir. Thierry Claerr et Isabelle Westeel, Villeurbanne, 2017 (La Boîte à outils), p. 46-64, DOI : 10.4000/books.pressesensib.426.
- SHARABOK (German), *Why Deep Learning Uses GPUs ?*, juill. 2020, URL : <https://towardsdatascience.com/why-deep-learning-uses-gpus-c61b399e93a0> (visité le 26/08/2023).
- SHARMA (Aditya), *Training the YOLOv5 Object Detector on a Custom Dataset*, juin 2022, URL : <https://pyimagesearch.com/2022/06/20/training-the-yolov5-object-detector-on-a-custom-dataset/> (visité le 18/08/2023).
- STRIEN (Daniel van), BEELEN (Kaspar), WEVERS (Melvin), SMITS (Thomas) et McDONOUGH (Katherine), « Computer Vision for the Humanities : An Introduction to Deep Learning for Image Classification (Part 1) », *Programming Historian* (, août 2022), URL : <https://programminghistorian.org/en/lessons/computer-vision-deep-learning-pt1> (visité le 10/08/2023).
- ULTRALYTICS, *Ultralytics YOLOv8 Docs*, URL : <https://docs.ultralytics.com/> (visité le 15/08/2023).
- Une API REST, c'est quoi ?*, URL : <https://www.redhat.com/fr/topics/api/what-is-a-rest-api> (visité le 27/08/2023).

Introduction

Première partie

Construire un corpus de numérisations pour le traitement par vision artificielle

Chapitre 1

Le projet EiDA

Cette partie a pour objectif de revenir sur le contexte institutionnel du projet EIDA (*Editing and analysing hIstorical astronomical Diagrams with Artificial intelligence*), projet de recherche mené à l’Observatoire de Paris par l’équipe d’histoire des sciences du laboratoire SYRTE. Ce projet a pour sujet d’étude les diagrammes astronomiques de tradition ptoléméenne : cette partie vise ainsi à contextualiser d’un point de vue historique le corpus du projet, et à en expliciter les bornes chronologiques et géographiques.

1.1 Contexte et objectifs du projet

1.1.1 Un projet de recherche en humanités numériques

Cadre du projet

Les sciences astronomiques se développent, à travers les siècles, à l’intersection de besoins et de questionnements divers auxquels elles tentent de répondre. Des besoins politiques aux besoins religieux, ces disciplines existent au cœur de contextes culturels, géographiques, historiques variés et interconnectés, et produisent, dans le cadre de leur pratique, des supports éclectiques qui témoignent de l’évolution des idées, et des conceptions du monde et de l’univers. Les manuscrits et imprimés produits dans le cadre des sciences astronomiques, pour répondre aux besoins de ceux qui les emploient, comportent des tables, des textes et des diagrammes d’une grande richesse, à l’image de la diversité de leurs contextes de production et de leurs utilisations.

Le projet EIDA¹, ayant pour vocation d’analyser ces sources dans le cadre de l’histoire des sciences, s’intéresse plus spécifiquement aux diagrammes astronomiques, en tant que patrimoine visuel et supports pour la circulation des idées et théories astronomiques et mathématiques. L’étude de ces circulations implique l’exploitation d’un corpus large,

1. *EdIter et analyser les Diagrammes astronomiques historiques avec l'intelligence Artificielle*, URL : <https://anr.fr/Projet-ANR-22-CE38-0014> (visité le 10/08/2023).

représentatif des bornes chronologiques et géographiques du projet, qui s'étendent de l'Europe à l'Asie, du VIII^e au XVIII^e siècle. Le projet EIDA a pour objectif d'étudier la diversité des diagrammes produits, du point de vue de l'utilisation mais aussi du support et de la diffusion. Les diagrammes sont étudiés d'une point de vue épistémique, par la définition de différentes typologies de diagrammes, l'analyse de leurs utilisations dans divers contextes et l'étude des sujets mathématiques et astronomiques qui les implique ; ainsi que d'un point de vue documentaire, avec un intérêt pour leurs contextes de production, le rapport texte-diagramme dans les manuscrits, et la circulation des représentations à une échelle locale ou globale.

Le projet s'appuie sur des sources latines, byzantines, arabes, persanes, hébraïques, sanskrites et chinoises, qui partagent des éléments communs permettant ainsi de retracer les évolutions des sciences astronomiques, et d'étudier leur continuité, leur diversité, et leurs connections à travers l'histoire de l'astronomie.

Le choix de la vision artificielle

Profondément ancré dans les méthodes des humanités numériques, le projet EIDA a pour objectif le développement d'une approche reposant sur la vision artificielle, qui permettrait d'appliquer au corpus une série de traitements en prévision de leur analyse et de leur exploitation. Le développement d'outils basés sur la vision artificielle a pour but de limiter les étapes manuelles de fouille et d'annotation dans le corpus de manuscrits, en automatisant les étapes d'extraction des diagrammes dans les numérisations, et leur transformation en objets aisément manipulables pour leur analyse, comme pour l'édition. La projet EIDA repose ainsi, dans ses principes fondateurs, sur une approche interdisciplinaire mêlant histoire des sciences et vision artificielle, avec pour objectif l'analyse de sources historiques dans le domaine des sciences astronomiques et le développement d'outils pour l'étude des diagrammes.

Dans une perspective de science ouverte, le projet EIDA prévoit le développement d'une application Web dédiée à l'extraction, la visualisation, l'étude et l'édition de diagrammes ; une plateforme dédiée à la communauté de la recherche qui permettra une utilisation des données produites et des outils développés par EIDA via une interface web. Cette plateforme permettra à un public de chercheurs, d'étudiants ou d'amateurs d'exploiter les algorithmes développés dans le cadre du programme pour traiter leurs propres sources ; à l'image de la plateforme développée dans le cadre du projet DISHAS (*Digital Information System for the History of Astral Sciences*), dédié aux tables astronomiques.

Annotations et données ouvertes

Le projet EIDA ayant pour objectif de produire un modèle de vision par ordinateur performant pour la détection de diagrammes dans des numérisations de manuscrits, un

jeu de données d’entraînement – ou vérité de terrain – est produit à partir d’annotations réalisées par les chercheurs du programme. Ce jeu de données est composé d’un ensemble d’images et de leurs fichiers d’annotation où sont localisés manuellement les diagrammes, selon les résultats espérés de la détection automatique. Cette vérité de terrain se veut, pour des meilleurs résultats, la plus exhaustive possible quant au matériau source, et constitue ainsi un jeu de données diversifié en termes d’images et d’annotations, aussi bien du point de vue de la période que de la provenance géographique, ou simplement du style de représentation.

EIDA prévoit ainsi de mettre à disposition ce jeu de données, pour permettre son exploitation par d’autres projets qui souhaiteraient entraîner un modèle de vision artificielle pour la détection d’objets dans des images. Il est envisagé, pour le bon référencement de ce jeu de données, de les référencer dans un catalogue à l’image de l’initiative HTR-United² pour la reconnaissance automatique des écritures manuscrites. L’annotation étant une étape cruciale des projets en *machine learning*, l’ouverture de ces données permettra ainsi à d’autres projets de bénéficier du travail effectué dans le cadre d’EIDA.

1.1.2 Avant EiDA : le projet DISHAS

Étudier les tables astronomiques

Précédant le projet EIDA, DISHAS s’intéresse au développement des sciences astronomiques, à leur circulation, à leur transmission à travers les cultures et les époques, par le prisme des tables. De la même manière que les diagrammes sont un objet d’étude permettant de retracer des traditions scientifiques à travers les cultures eurasiennes, les tables sont un objet caractéristique des sciences astronomiques, dont le contenu numérique rend possible l’alignement et la comparaison entre différentes sources pour une étude de la circulation des idées entre les cultures et les pratiques scientifiques.

Le projet DISHAS emploie les méthodologies des humanités numériques et les applique au traitement des sources en histoire de l’astronomie. Dans cette optique, une application dédiée à l’édition et à l’analyse des tables astronomiques a été mise en ligne³ : cette application propose une contextualisation historique, chronologique et géographique des tables⁴, ainsi qu’une série d’outils plus spécifiques, pour leur étude par un public divers, plus ou moins spécialisé. Parmi ces outils dédiés au traitement des tables, DTI (*DISHAS Table Interface*) est destiné à la saisie de tables numériques et DIPS (*DISHAS Interactive*

2. HTR-United ne propose pas de cataloguer des jeux de données pour la *computer vision*, il est cependant envisagé de prendre contact avec les porteurs de l’initiative pour discuter de la possible intégration des données du projet EIDA. Alix Chagué et Thibault Clérice, *HTR-United*, URL : <https://htr-united.github.io/index.html> (visité le 10/08/2023)

3. *Home — DISHAS Project — Astronomical Table in Database*, URL : <https://dishes.obspm.fr/> (visité le 10/08/2023).

4. *Historical Navigation — DISHAS Project — Astronomical Table in Database*, URL : <https://dishes.obspm.fr/historical-navigation> (visité le 10/08/2023).

Parameter Squeezer) propose l’analyse mathématique de tables astronomiques anciennes ; mettant ainsi à disposition de la communauté de la recherche les résultats du projet, pour que les données comme les outils puissent être utilisés dans le cadre de recherches dans le domaine de l’histoire des sciences.

Approche par la vision artificielle

L’approche des données visuelles par l’intelligence artificielle est un élément clé du projet DISHAS, ayant donné lieu au développement de TableTranscriber⁵, un outil dédié à la détection par un algorithme de vision artificielle⁶ de la structure des tables, afin d’en transcrire le contenu manuscrit à l’aide d’un algorithme de HTR. L’objectif de TableTranscriber est ainsi d’automatiser la chaîne de traitement des numérisations de tables astronomiques manuscrites, afin d’obtenir des données au format XML, CSV ou HTML, aisément manipulables pour une analyse des sources.

Cette volonté collaborative dans le développement d’outils numériques est centrale au projet EIDA qui, comme son prédecesseur DISHAS, vise à produire une recherche communautaire, aux acteurs scientifiques divers, à l’intersection des humanités numériques et de l’histoire des sciences.

1.1.3 Acteurs du projet

SYRTE-PSL-Observatoire de Paris, CNRS

EIDA est un projet interdisciplinaire mené en collaboration avec de nombreux acteurs et institutions, pour en faire un projet réellement collaboratif dans la réalisation de ses objectifs. Au sein de l’Observatoire de Paris, l’équipe est composée d’historiens et d’ingénieurs affiliés au SYRTE (Systèmes de Référence Temps-Espace), ayant pour la plupart collaboré au projet DISHAS, et possédant donc une expérience de la recherche en histoire des sciences, et en humanités numériques. L’équipe compte également des chercheurs indépendants et collaborateurs scientifiques externes, qui assurent la diversité du corpus en contribuant à EIDA par leur expertise dans des sources issues d’aires géographiques et chronologiques qui s’ajoutent à celles représentées dans l’équipe d’histoire des sciences de l’Observatoire de Paris.

Du point de vue des institutions, le groupe de recherche en vision artificielle IMAGINE (Laboratoire d’Informatique Gaspard Monge) basé à l’École des Ponts ParisTech, co-coordonne le projet EIDA, et assure notamment les recherches en vision artificielle et apprentissage machine nécessaires à la réalisation des objectifs du projet. Le projet VHS (Vision artificielle et analyse Historique de la circulation de l’illustration Scientifique), coordonné par l’Institut des sciences du calcul et des données, est également partenaire

5. [tristandotTableTranscriberAutomaticPipeline2022](#).

6. Basé sur [monnieriDocExtractor2023](#).

d'EIDA pour le développement des algorithmes de détection et de la plateforme dédiée aux chercheurs.

Algorithmes de vision par ordinateur : le groupe de recherche IMAGINE

L'équipe IMAGINE— avec laquelle l'équipe d'histoire des sciences avait déjà collaboré pour le développement de tableTranscriber dans le projet DISHAS— apporte au projet EIDA son expertise dans le domaine de l'apprentissage profond et de la vision artificielle appliqués aux humanités numériques. Les recherches menées entre 2018 et 2022 dans le cadre du projet EnHerit (*Enhancing Heritage Image Databases*)⁷, dédié à l'application d'algorithmes de vision à l'histoire de l'art pour la reconnaissance de motifs dans les œuvres d'art, sont le socle des développements menés pour les projets EIDA et VHS.

Dans le cadre de leur participation à EIDA, l'objectif des chercheurs d'IMAGINE est le développement de nouvelles approches⁸ pour permettre la décomposition et la vectorisation de dessins techniques. Ces nouvelles approches par vision par ordinateur devraient permettre l'analyse et l'édition de diagrammes astronomiques sans intervention humaine, avec une méthode plus flexible dans son apprentissage, qui allègerait ainsi les étapes d'annotation nécessaires à l'apprentissage profond.

Ces recherches en apprentissage faiblement supervisé sont également menées dans le cadre du projet VHS, auquel l'équipe IMAGINE contribue également, avec un accent mis sur la détection de similarité dans les images et dans le texte, qui devraient permettre une navigation facilitée des corpus étudiés, et qui seront également exploitées dans EIDA.

Le projet ANR VHS

Le projet VHS réunit des chercheurs en histoire des sciences et en vision artificielle pour l'étude de la circulation des savoirs scientifiques au Moyen-Âge et à la période moderne. Coordonné par Alexandre Guilbaud, VHS a pour objectif de développer des méthodes d'analyse de corpus illustrés par apprentissage profond, afin d'étudier la circulation, l'évolution et la réutilisation des images scientifiques dans le cadre de la transmission des savoirs à travers les époques et les zones géographiques, et d'analyser le rapport des images au contenu textuel des ouvrages dans lesquels elles figurent. Les méthodes développées par le projet ont pour objectif de limiter la nécessité des annotations pour l'obtention de résultats dans la détection de similarité, qui permettraient ainsi de faciliter la navigation de corpus scientifiques par des regroupements et rapprochements basés sur les images et sur le texte.

7. *Enhancing Heritage Image Databases*, URL : <https://anr.fr/Project-ANR-17-CE23-0008> (visité le 10/08/2023).

8. Basées sur les progrès récents de l'analyse par synthèse. Tom Monnier, Elliot Vincent, Jean Ponce et Mathieu Aubry, *Unsupervised Layered Image Decomposition into Object Prototypes*, août 2021, DOI : 10.48550/arXiv.2104.14575, arXiv : 2104.14575 [cs]

Ces recherches s'accompagnent du développement d'une application Web, qui mettra à disposition de la communauté scientifique les outils et données produites dans le cadre du projet VHS, afin que ceux-ci puissent être exploités par d'autres équipes travaillant dans le domaine des études visuelles, ou souhaitant apporter une analyse sur des corpus illustrés par le prisme de la circulation des motifs et des idées.

Les membres de l'équipe IMAGINE sont sollicités lors des prises de décision relatives aux travaux menées par les chercheurs⁹ et participent aux discussions relatives au développement de la plateforme. De plus, pour un développement plus fluide des applications VHS et EIDA, l'ingénieur chargé de ce développement pour le projet VHS mène régulièrement ses travaux à l'Observatoire, pour faciliter les échanges entre les deux équipes. Ces collaborations permettent une mutualisation des recherches et expertises des différents projets, ainsi que le développement commun d'outils réutilisables qui bénéficient des regards variés et complémentaires des différents acteurs.

1.2 Sources primaires

1.2.1 L'astronomie ptoléméenne : naissance et diffusion

L'Almageste et le modèle ptoléméen

L'astronomie et les disciplines qui y sont liées sont, dès l'Antiquité, cultivées dans un grand nombre de cultures où se développent des pratiques variées, appuyées sur l'observation des objets célestes et le développement de systèmes mathématiques pour en prédire le comportement. Jusqu'à l'arrivée des théories de Nicolas Copernic au XV^e siècle, les travaux de Ptolémée¹⁰ influencent, à travers l'Eurasie, les productions des astronomes. L'œuvre de Ptolémée devient en effet dès le II^e siècle une référence en matière d'astronomie : développées sur les travaux de ses prédecesseurs grecs, les théories de Ptolémée approfondissent les productions de son temps dont il fait la synthèse dans ses travaux, et qu'il enrichit de ses propres observations. Si le modèle géocentrique est déjà établi avant l'achèvement de *L'Almageste*, les travaux de Ptolémée perfectionnent la théorie des épicycles¹¹, et parviennent à l'affiner avec une précision suffisante pour la prédiction des

9. On considère l'annotation des diagrammes dans les manuscrits comme un travail de recherche.

10. Claude Ptolémée (v. 100-v. 170) est un astronome, mathématicien et géographe actif à Alexandrie au II^e siècle de notre ère. *L'Almageste*, complété vers 150, comprend un quart de siècle d'observations des mouvements des objets célestes, expliqués par des systèmes mathématiques qu'il développe dans son œuvre. Le système ptoléméen désigne le modèle géocentrique de l'univers qu'il développe dans ses travaux. Alexander Raymond Jones, « Ptolemy », dans *Encyclopedia Britannica*, URL : <https://www.britannica.com/biography/Ptolemy> (visité le 03/08/2023).

11. Introduite par les astronomes de la Grèce antique, la théorie des épicycles permet, dans un modèle géocentrique, d'expliquer les changements de vitesse et de direction dans le mouvement observé des planètes, du Soleil et de la Lune. Selon cette théorie, les objets célestes se déplacent à vitesse uniforme sur un cercle appelé épicycle, dont le centre est lui-même en rotation sur un cercle centré sur la Terre appelé déférant. Jean-Jacques Rousseau, *Epicycles de Ptolémée*, URL : <http://ressources.univ->

positions des planètes, notamment par l'introduction du point équant¹².

L'Almageste est constitué d'observations et descriptions des procédures mathématiques appliquées par Ptolémée pour établir les paramètres de son modèle géométrique¹³. À ses explications textuelles s'ajoutent des tables de valeurs numériques calculées par Ptolémée qui permettent, en suivant ses explications, de calculer la position des planètes, en tant qu'application de ses théories. Dans les siècles qui suivent le développement du modèle ptoléméen, peu d'innovations ont lieu dans les sciences astronomiques de tradition hellénique, et les travaux produits prennent essentiellement la forme de commentaires des théories de Ptolémée, sans remise en cause de son modèle, qui reste la norme pendant plusieurs siècles et est diffusé à travers l'Eurasie par le biais de traductions.

Diffusion du modèle ptoléméen

Avant les Grecs, les Babyloniens pratiquent, dès le I^e millénaire av. J.-C., une astronomie basée sur des calculs arithmétiques permettant de prédire la position des planètes. L'astronomie indienne hérite de ces théories des sciences astronomiques babylonniennes qu'elle mêle avec des méthodes locales, qui s'enrichissent plus tard des travaux de Ptolémée et de son prédecesseur Hipparque, dont les idées voyagent vers l'Est¹⁴ par le biais de la Perse.

À partir du VIII^e siècle, dans le monde islamique, se développe une pratique de l'astronomie à l'intersection des théories grecques, babylonniennes, perses et indiennes. Suivant le modèle des tables de Ptolémée, les astronomes arabes développent au IX^e siècle des ouvrages de tables, appelés *zīj*, dédiés au calcul des positions du Soleil, de la Lune et des planètes mêlant ces diverses traditions. La traduction en arabe – pendant le califat abbasside¹⁵ – des écrits de Ptolémée permet aux astronomes du monde islamique d'enrichir leurs pratiques en l'inscrivant dans cette tradition, tout en améliorant certains paramètres calculés dans le cadre de son modèle pour en préciser les prédictions¹⁶. Si des critiques des travaux de Ptolémée émergent, notamment au XI^e siècle dans les écrits d'Ibn al-Haytham,

lemans.fr/AccesLibre/UM/Pedago/physique/02/divers/ptolemee.html (visité le 05/08/2023)

12. En effet, la théorie des épicycles telle qu'elle existe avant l'intervention de Ptolémée ne permet pas de justifier précisément des mouvements observés des planètes. Ptolémée introduit ainsi la notion de point équant, point excentré par rapport au centre de la Terre, à partir duquel la vitesse de rotation des corps célestes est constante. Ce modèle géométrique permet de calculer avec précision le mouvement longitudinal des planètes. James Evans, « History of Astronomy », dans *Encyclopedia Britannica*, URL : <https://www.britannica.com/science/astronomy/Ancient-Greece> (visité le 03/08/2023)

13. *Ibid.*

14. *Ibid.*

15. La dynastie sunite des Abbassides gouverne le monde musulman de 750 à 1258. La capitale du califat, Bagdad, est le siège d'entreprises de traduction vers l'arabe d'écrits scientifiques, notamment depuis le grec.

16. L'écart temporel entre les calculs de Ptolémée et ceux des astronomes arabes du IX^e siècle permettent également de mettre en évidence certains déplacements observables seulement sur un temps plus long, tels que la diminution de l'obliquité de l'écliptique.

son modèle reste prédominant jusqu'aux travaux de Copernic¹⁷.

Dans le monde latin, les théories grecques se sont peu transmises, et l'enseignement de l'astronomie s'appuie essentiellement sur les écrits de Pline l'Ancien. Au XII^e et XIII^e siècles, les textes grecs sont rendus disponibles par des initiatives de traduction : Gérard de Crémone, notamment, traduit *L'Almageste*¹⁸ de l'arabe vers le latin, participant activement à la redécouverte du modèle ptoléméen par les scientifiques du monde latin. La transmission des écrits et théories astronomiques ne se fait donc pas directement des Grecs au Latin, mais a pour intermédiaires les traductions effectuées aux siècles précédents dans le monde arabo-musulman. En parallèle, le développement des universités réintègre l'astronomie à l'enseignement des arts libéraux : dans ce contexte d'émulation, de nouveaux ouvrages sont publiés, et notamment des tables astronomiques, toujours basés sur le modèle ptoléméen.

Les sources chinoises témoignent d'une pratique de l'astronomie aussi ancienne que le II^e millénaire av. J.-C. sous la forme de prédiction des éclipses du Soleil et de la Lune. Cette pratique est profondément liée à la culture impériale, et les éclipses – puis plus tardivement, les planètes – sont un outil d'analyse du règne d'un empereur. Plus proche des procédures babylonniennes que du modèle géométrique grec, l'astronomie chinoise s'appuie avant tout sur l'arithmétique pour ses prédictions, et s'intéresse principalement aux événements ponctuels – tels que les comètes ou les éclipses – plutôt qu'aux mouvements des corps célestes¹⁹.

Jusqu'à l'émergence du modèle héliocentrique, et la mise au centre de la pratique des théories de Copernic, le modèle ptoléméen représente, dans de nombreux contextes culturels, le socle des méthodes des sciences astronomiques. Qu'il soit amélioré, contesté, ou enrichi de pratiques locales, il reste un standard de la conception de l'astronomie en Eurasie, et permet ainsi de retracer une tradition des pratiques à travers les siècles.

1.2.2 Les diagrammes

Une culture visuelle des sciences astronomiques

Dans le contexte eurasiatique, les supports détaillant les pratiques des astronomes circulent entre les siècles et les cultures, sont enrichis ou adaptés aux pratiques autochtones de l'astronomie, en réponse aux besoins spécifiques de chaque contexte culturel. Les manuscrits et imprimés, supports de ces pratiques, sont les témoins de ces échanges intellectuels et permettent d'établir une histoire des idées et des méthodes héritées du modèle ptoléméen (fig. 1.1). Ces œuvres comprennent, en support des pratiques, des textes, des tables et des diagrammes, tous porteurs des méthodes et théories des sciences astrono-

17. *Ibid.*

18. Ainsi que d'autres versions arabes de textes grecs, tels que *Du ciel* d'Aristote, ou les *Éléments* d'Euclide.

19. *Ibid.*

miques : le projet DISHAS ayant mené une étude sur les tables astronomiques, il semble naturel qu'EIDA, dans la continuité de celui-ci, ait pour vocation de s'intéresser spécifiquement aux diagrammes. Les astronomes, dans leur pratique, n'hésitent pas à employer des modes d'expression visuels, au-delà du texte, qu'il convient d'étudier en parallèle de celui-ci pour en comprendre les interactions. Ainsi, il est nécessaire de développer des outils permettant d'appréhender les formes variées que peuvent prendre les sources en histoire de l'astronomie ; à l'image des récentes recherches en histoire des sciences, qui s'appliquent notamment à une étude des sources visuelles, non-discursives, de la pratique scientifique, et en considérant les illustrations comme des objets d'étude à part entière, au-delà d'un simple accompagnement du contenu textuel²⁰.

En tant que vecteurs d'une culture visuelle de l'astronomie, les diagrammes sont, pour les astronomes, le support d'une pratique scientifique, et sont ainsi révélateurs de leurs méthodes, de leur contexte d'exercice et de leur conception de ces disciplines. Le projet vise à proposer une analyse de ces diagrammes qui souligne la variété de leurs fonctions et des traditions auxquelles ils appartiennent, tout en étudiant leurs modes de circulation à travers l'histoire de l'astronomie. Dans la diversité de leurs sphères de création, ces diagrammes reçoivent « une légitimité en tant qu'instruments pédagogiques et vecteurs de la pensée²¹ », et occupent ainsi dans le contexte d'une culture visuelle globale une place particulière.

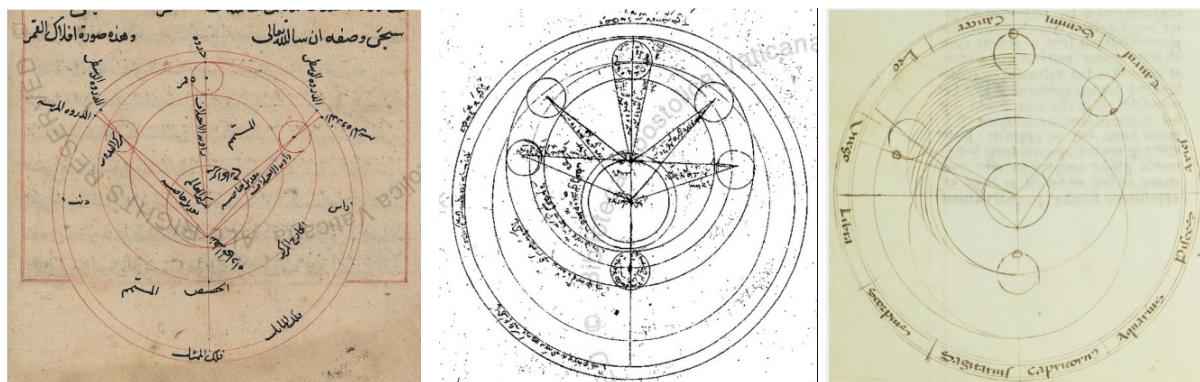


FIGURE 1.1 – Modèles lunaires de tradition arabe, byzantine et latine

Bornes géographiques et chronologiques

Ayant la volonté de produire une étude des circulations et évolutions des diagrammes astronomiques, EIDA s'attache à l'étude de sources provenant de sphères géographiques

20. Boris Jardine et Nicholas Jardine, « Critical Editing of Early-Modern Astronomical Diagrams », *Journal for the History of Astronomy*, 41–3 (août 2010), p. 393-414, DOI : 10.1177/002182861004100307, p. 394

21. « [...] legitimacy as instruments of pedagogy and as vehicle of thought ». Jeffrey F. Hamburger, David J. Roxburgh et Linda Safran (éd.), *The Diagram as Paradigm : Cross-Cultural Approaches*, Cambridge, MA, 2022 (Dumbarton Oaks Byzantine Symposia and Colloquia), p. 1.

et temporelles larges : les pratiques eurasiatiques des sciences astronomiques partagent des éléments communs qui justifient une étude au cadre vaste, pour proposer une analyse transculturelle des continuités et divergences dans l'histoire de l'astronomie comme dans la culture visuelle. Pour répondre à cette volonté, le projet EIDA s'appuie sur cinq corpus aux provenances temporelles et géographiques diverses, provenant de traditions diverses également, dans une volonté de représentativité.

Les manuscrits astronomiques arabo-persans, produits du VIII^e au XVIII^e siècle, représentent à eux seuls un corpus d'une grande diversité, aux traditions diverses aussi bien du point de vue des pratiques que de la provenance géographique. Les travaux produits dans ces contextes circulent à travers l'Eurasie du Moyen-Âge au début de la période moderne, et influencent de nombreuses pratiques. Les manuscrits d'astronomie latin médiévaux sont produits majoritairement entre le XIII^e et le XVI^e siècle dans un contexte universitaire, et sont influencés par les traductions faites de l'arabe au latin de textes de tradition arabe ou persane. Les manuscrits byzantins, produits entre les IX^e et XV^e siècles sont à l'intersection des traditions helléniques, latines et arabo-persanes, et influencent considérablement les manuscrits astronomiques latins du XV^e et du XVI^e siècle. Les manuscrits sanskrits, dont la diversité reflète les nombreuses traditions établies sur près de deux millénaires, témoignent à partir du XI^e siècle d'influences helléniques puis arabo-persanes. Les sources chinoises, dont le support n'est pas nécessairement le manuscrit, prennent souvent la forme d'imprimés par blocs xylographiques dont les matrices sont réemployées à travers plusieurs témoins. Ces documents témoignent des influences arabo-persanes puis latines des pratiques astronomiques chinoises à partir de la dynastie Ming²². Les sources étudiées dans le cadre d'EIDA proviennent essentiellement des dynasties Ming et Qing²³.

Les bornes de ces cinq corpus représentent les bornes chronologiques et géographiques du projet. S'étendant du VIII^e au XVIII^e siècle, les périodes et aides étudiées se veulent représentatives du contexte afro-eurasien de la circulation des idées et des images. Ces corpus représentent, dans chaque cas, plusieurs milliers de manuscrits. Plusieurs centaines de manuscrits sont numérisés pour chaque tradition représentée dans le projet, permettant ainsi la représentativité espérée dans le cadre du projet EIDA. Ces numérisations, mises à disposition par les institutions patrimoniales qui conservent ces témoins, représentent les sources primaires du projet, sur lesquelles seront appliquées les traitements en prévision de l'analyse par les chercheurs.

EIDA est un projet aux bornes chronologiques, géographiques et thématiques vastes, pour permettre une étude sur un temps long et dans un contexte global de la circulation des diagrammes astronomiques et des théories scientifiques qui les accompagnent. Dans cette démarche, proposer une étude quantitative, traitant un grand nombre de sources,

22. La dynastie Ming règne sur la Chine de 1368 à 1644.

23. La dynastie Qing, dernière dynastie impériale chinoise, règne de 1644 à 1912.

permet de mettre en avant des motifs, connexions et évolutions en accord avec la diffusion afro-eurasienne des idées développées par Ptolémée. Les bornes définies dans cette partie permettent de délimiter un corpus d'images – numérisations d'ouvrages manuscrits ou imprimés – qui en tant d'objets numériques présentent également leurs propres problématiques, sur lesquelles revient la partie suivante.

Chapitre 2

Images et interopérabilité

Les sources étudiées dans ce mémoire sont des sources iconographiques : cette partie vise à revenir sur les formats, ressources et méthodes pour le traitement des images en ligne, de la production de la ressource à sa publication. Les images digitales présentent des enjeux spécifiques, du point de vue de la technique, du droit et de la disponibilité. La mise en ligne et la diffusion d'une image fait suite à une longue chaîne de traitement qui a pour point de départ un objet matériel, et soulève des questionnements divers. Les spécifications IIIF (*International Image Interoperability Framework*) tente de répondre à un certain nombre de problématiques liées à la présence en ligne de ressources iconographiques, et cette partie vise donc à présenter les solutions, possibilités et limites offertes par ce standard.

2.1 L'image comme source

2.1.1 Construire un corpus d'images : enjeux et méthodes

La numérisation, enjeu patrimonial

L'application à un corpus de sources historiques de méthodes numériques repose sur une première étape cruciale dans le traitement de ces sources : la numérisation des collections patrimoniales. De la bibliothèque au musée, la numérisation du patrimoine culturel est un enjeu crucial depuis le début des années 1990¹. Pensée initialement comme un outil de préservation, de valorisation et d'accessibilité aux collections², la mise à disposition des collections au format numérique ouvre également la voie à de nouvelles méthodologies

1. Corinne Baujard, « Numérisation du patrimoine culturel et stratégie managériale des musées », *Management des technologies organisationnelles*-7 (févr. 2017), p. 69-78, URL : <https://www.cairn.info/revue-management-des-technologies-organisationnelles-2017-2-page-69.htm> (visité le 06/08/2023).

2. Michel Richard, « Le programme de numérisation de la Bibliothèque de France », *Bulletin des bibliothèques de France*-3 (1993), p. 53-63, URL : <https://bbf.enssib.fr/consulter/bbf-1993-03-0053-007> (visité le 06/08/2023).

de la recherche, notamment dans le champs disciplinaire des humanités numériques : dans le cadre d'un projet tel qu'EIDA, qui prévoit des traitements automatiques des sources par des algorithmes de vision par ordinateur, la disponibilité des sources numérisées est un principe fondateur de la démarche de recherche.

La numérisation des données culturelles fait désormais partie des pratiques courantes des institutions culturelles, et plus particulièrement des bibliothèques, représentant une part intégrante du travail de conservation et de diffusion du patrimoine³. Dès le milieu 1990, des initiatives telles que Gallica⁴, bibliothèque numérique de la BnF (Bibliothèque nationale de France), voient le jour, et mettent à disposition du public, au format numérique, les collections de la bibliothèque. Dans les années qui suivent, des initiatives telles qu'Europeana⁵, permettent le déploiement de projets de plus large envergure⁶, portant toujours cette responsabilité de préservation, diffusion et valorisation des collections dans un contexte international. La notion de patrimoine culturel numérique prend tout son sens, et Internet devient un espace permettant à des utilisateurs aux profils variés d'exploiter ces ressources numériques, et d'accéder à un patrimoine vaste, au-delà des limites physiques de la consultation des documents. Cependant, malgré les initiatives de soutien à la numérisation portées par les collectivités, ou même par l'État⁷, il est nécessaire de souligner les disparités, aussi bien sur le plan national qu'international, de ces entreprises de numérisation, souvent coûteuses : ainsi, il n'existe pas de réelle égalité entre les institutions dans la numérisation et la mise en ligne des collections, un biais qu'il est nécessaire de prendre en compte, et particulièrement dans des projets de recherche appuyés sur des sources aux bornes géographiques internationales.

Dans le cas des corpus étudiés par le projet EIDA, on compte, pour des sources qui représentent plusieurs centaines de milliers de documents, une somme de plusieurs centaines de documents numérisés, mis à disposition par les institutions patrimoniales européennes, chinoises et indiennes : nous constatons ainsi que les documents numérisés ne représentent qu'un fragment de la réalité des sources existantes, cependant, ces centaines de documents numérisés sont à la fois suffisantes et suffisamment représentatives pour le bon déroulé du projet.

Les collections patrimoniales des musées présentent un tout autre enjeu en termes de mise en ligne : les œuvres conservées, allant bien au-delà du format livre, ont une diversité qui redéfinit entièrement la notion de numérisation⁸. Tout comme les bibliothèques, les institutions muséales font face, avec le développement des pratiques numériques, à une

3. Thierry Claerr et Isabelle Westeel, « Mode d'emploi », dans *Numériser et mettre en ligne*, Villeurbanne, 2017 (La Boîte à outils), p. 7-18, DOI : 10.4000/books.pressesenssib.419.

4. *Gallica*, URL : <https://gallica.bnf.fr/> (visité le 10/08/2023).

5. *Europeana*, URL : <https://www.europeana.eu/fr> (visité le 10/08/2023).

6. Id., « Mode d'emploi »...

7. *Ibid.*

8. Par numérisation, dans le contexte muséal, nous entendons essentiellement la prise de photographies de haute qualité des œuvres d'arts.

transformation des pratiques de conservation pour inclure ces méthodes. Outre les aspects documentaires de la numérisation, le Web est un lieu d'exposition à part entière, et la numérisation des œuvres présentent ainsi un intérêt stratégique en termes de valorisation et de visibilité⁹. Dans le monde muséal, néanmoins, l'accès libre aux images n'est pas aussi systématique qu'en bibliothèque¹⁰ : il existe bel et bien des projets de numérisation de grande envergure, tels que Google Arts & Culture¹¹, publié en 2011. La vocation de ce type de projets, cependant, semble plutôt être la création d'espaces d'exposition dématérialisés permettant de mettre en valeur les œuvres pour des spectateurs, plutôt que la véritable mise à disposition de ces images pour un usage par des utilisateurs. Nous soulignons, notamment, dans le cas de Google Arts & Culture, l'impossibilité de copier les images avec un clic droit, et l'absence de bouton pour le téléchargement des numérisations sur un certain nombre de pages : les numérisations sont faites pour être vues, et non utilisées¹².

Une étude menée entre 2019 et 2020 par le Musée National de Tokyo souligne, en parallèle, les disparités dans la qualité des numérisations, qui ne permet ainsi pas de considérer qu'un objet numérisé ou photographié l'est nécessairement dans une qualité exploitable¹³, notamment parce que les besoins en termes de qualité ne sont pas les mêmes pour des œuvres en 2D et des œuvres en 3D, qui souffrent particulièrement de la mauvaise qualité des numérisations. Ainsi, en prenant en compte ces critères, il est nécessaire de soulever la question de la réelle exploitabilité des images mises en ligne, et particulièrement dans le cadre de projets en vision artificielle qui reposent, dans leurs fondations, sur la disponibilité et la qualité des images du corpus choisi. Des initiatives récentes tentent cependant de pallier à ces lacunes des institutions patrimoniales quant à la mise à disposition des images, comme la création en 2015 d'une API (*Application Programming Interface*) donnant accès aux reproductions de la base photographie de la Rmn-Grand Palais¹⁴, mais la norme, pour ces images d'œuvres d'art, reste loin de l'*open access*¹⁵.

9. C. Baujard, « Numérisation du patrimoine culturel et stratégie managériale des musées »...

10. Pour comparaison, Gallica représente aujourd'hui près de 10 millions de documents, et la base de la Rmn-Grand Palais (Réunion des musées nationaux-Grand Palais) compte 600 000 œuvres numérisées.

11. *Google Arts & Culture*, URL : <https://artsandculture.google.com/> (visité le 10/08/2023).

12. Certaines œuvres du projet sont disponibles sur Wikimedia, mais ces images en accès libre ne représentent qu'un fragment des œuvres numérisées dans le cadre du projet. De plus, les musées français sont très peu représentés dans cette sélection. *Category :Google Art Project Works by Collection - Wikimedia Commons*, URL : https://commons.wikimedia.org/wiki/Category:Google_Art_Project_works_by_collection (visité le 10/08/2023)

13. Akira Sakai, *Digitizing Disparity in the Museum. The Object-Based Research in the Tokyo National Museum*, août 2021, URL : <https://universityheritage.eu/en/digitizing-disparity-in-the-museum-the-object-based-research-in-the-tokyo-national-museum/> (visité le 06/08/2023).

14. *API de La RMN-Grand Palais*, URL : <https://api.art.rmnrgp.fr/> (visité le 10/08/2023).

15. Isabelle Manca, « Les nouveaux défis des agences photographiques des musées », *Le Journal Des Arts* (, juin 2018), URL : <https://www.lejournaldesarts.fr/patrimoine/les-nouveaux-defis-des-agences-photographiques-des-musees-138076> (visité le 06/08/2023).

Droit d'auteur et coût des images

La numérisation et la mise en ligne de documents patrimoniaux implique également de prendre en compte la question des droits d'auteur, qui régit souvent les possibilités des institutions en termes de libre accès, et impacte par conséquent les projets, aussi bien du point de vue des exploitations possibles que de la publication de leurs résultats¹⁶.

Une œuvre originale est protégée au titre du droit d'auteur : ce droit comprend le droit moral, perpétuel et inaliénable, qui donne droit au respect de l'œuvre et du nom de son auteur, ainsi que le droit patrimonial, qui donne à l'auteur et ses ayants droit le droit d'autoriser ou non la reproduction et la diffusion de l'œuvre¹⁷. Les droits patrimoniaux s'éteignent soixante-dix ans après la mort de l'auteur, et l'œuvre entre alors dans le domaine public¹⁸. Il n'existe pas d'exception au droit d'auteur pour une numérisation mise en ligne sur Internet¹⁹. Ces droits, qui s'appliquent à l'œuvre originale, ne sont pas nécessairement ceux qui nous intéressent dans le cadre d'un projet de recherche en histoire, dont les œuvres du corpus ont souvent rejoint le domaine public. La photographie d'une œuvre, cependant, peut faire l'objet du droit d'auteur, si celle-ci n'est pas considérée comme une reproduction servile.

Outre le droit d'auteur et les droits du photographe, les institutions peuvent exiger une redevance de réutilisation ou de prestation d'un service pour l'utilisation de leurs images, qui sert à couvrir, par exemple, les frais engagés pour la production du cliché, le traitement de la demande, ou la recherche iconographique²⁰. Beaucoup d'institution, telles que la BnF et l'INHA (Institut national d'histoire de l'art)²¹, proposent des tarifications particulières pour des projets académiques, scientifiques ou non commerciaux, et n'appliquent un coût qu'en cas de demande de numérisation d'œuvres qui ne sont pas déjà numérisées. Les projets de recherche peuvent ainsi faire le choix de construire leurs corpus en prenant en compte les documents déjà disponibles, et mis en ligne sous licence libre.

16. Olivier Jacquot, *Décrire, transcrire et diffuser un corpus documentaire hétérogène : méthodes, formats, outils*, Billet, nov. 2017, URL : <https://bnf.hypotheses.org/2214> (visité le 06/08/2023).

17. Sophie Sepetjan, « Respecter le droit de la propriété littéraire et artistique », dans *Numériser et mettre en ligne*, dir. Thierry Claerr et Isabelle Westeel, Villeurbanne, 2017 (La Boîte à outils), p. 46-64, DOI : 10.4000/books.pressesenssib.426.

18. Il existe des cas particuliers, notamment pour les œuvres collectives ou les journaux, ainsi que pour les publications posthumes. Institut national d'histoire de l'art, *Guide Pratique Pour La Recherche et La Réutilisation Des Images d'œuvres d'art*, URL : https://www.inha.fr/_resources/PDF/2021/2021_LIVRET_GUIDE_PRATIQUE_web_15062021.pdf.

19. S. Sepetjan, « Respecter le droit de la propriété littéraire et artistique »...

20. Institut national d'histoire de l'art, *Guide Pratique Pour La Recherche et La Réutilisation Des Images d'œuvres d'art...*

21. Martine Denoyelle, *A propos du coût des images d'œuvres patrimoniales*, Billet, juin 2021, URL : <https://numrha.hypotheses.org/2142> (visité le 06/08/2023).

2.1.2 Images et données ouvertes

Licences Creative Commons

Pour favoriser la libre circulation des images sur le Web, des licences ont été créées pour encadrer la réutilisation des numérisations. Dès 2001, les licences Creative Commons²² sont créées avec la vocation de favoriser la diffusion et le parage des images dans un contexte numérique, en assouplissant notamment les droits de propriété intellectuelle pour s'adapter à ce nouveau cadre. Ces licences se basent sur un système de quatre options qu'il est possible de combiner pour créer six licences différentes. Ces options sont attribution (BY), non commercial (NC), sans modification (ND) et partage à l'identique (SD)²³.

À partir de 2011, l'État français crée ses propres licences ouvertes, comme la licence unique Etalab, qui vise à « permettre et encourager la réutilisation des données publiques²⁴ ». Une deuxième version, publiée en 2017, se veut compatible avec la licence CC BY, c'est-à-dire une licence ouverte avec attribution de l'œuvre à son auteur.

Les données ouvertes sont un enjeu central des politiques culturelles : dans un contexte patrimonial, ces licences libres, conçues pour adapter les droits de propriété intellectuelle au contexte particulier du numérique, permettent ainsi la circulation, la diffusion et le partage de l'information. Il n'existe pas encore de cadre juridique international régissant l'utilisation des images produites par numérisation de collections patrimoniales, ainsi, les institutions sont soumises au droit spécifique du pays qui les abrite. En l'absence d'un encadrement global des pratiques, les utilisateurs – et notamment la communauté de la recherche – font face aux exceptions propres à chaque contexte national, qui représentent encore un obstacle. Une réflexion autour de ce cadre juridique international permettrait, en parallèle, d'harmosniser à une échelle globale les formats des données ouvertes²⁵, pour en faciliter la réutilisation par les institutions et par les projets.

Enjeux techniques : images et métadonnées

Le document numérique obtenu à la suite d'une processus de numérisation est, en réalité, constitué de plusieurs fichiers répondant à des besoins différents pour la mise en ligne. Chacun de ces éléments représentent leurs propres enjeux techniques, dont la gestion impacte la qualité et l'exploitabilité des images mises en ligne à la suite d'entreprises de numérisation.

22. Ces licences ne sont plus reconnues par l'État français depuis 2016. Id., *En savoir plus sur la licence Etalab et les autres licences*, Billet, juin 2021, URL : <https://numrha.hypotheses.org/2134> (visité le 06/08/2023).

23. Institut national d'histoire de l'art, *Guide Pratique Pour La Recherche et La Réutilisation Des Images d'œuvres d'art...*

24. M. Denoyelle, *En savoir plus sur la licence Etalab et les autres licences...*

25. Yaniv Benhamou, « Droit d'auteur et musées numériques », *Magazine de l'OMPI-3* (juin 2016), URL : https://www.wipo.int/wipo_magazine/fr/2016/03/article_0005.html (visité le 06/08/2023).

Les fichiers image issus d'un processus de numérisation peuvent avoir plusieurs formats²⁶ aux propriétés similaires mais différentes. En premier lieu, il est de bonne pratique de privilégier un format ouvert²⁷, par opposition à un format propriétaire qui nécessite un logiciel spécifique pour être lu. Le choix d'un format ouvert permet d'assurer l'accessibilité des données produites. Les formats .tiff et .jpeg sont favorisés dans le contexte de numérisation des collections patrimoniales, parce qu'ils permettent notamment la compression des fichiers²⁸ tout en maintenant la haute qualité des images²⁹. Les choix effectués par les institutions impactent la qualité des numérisations produites qui, sur un plus long terme, impacte aussi bien la préservation des données, l'expérience des utilisateurs qui consultent les images, et l'exploitabilité des fichiers par des projets. La question de la qualité des images est cruciale dans le cadre d'un projet de recherche en vision artificielle, puisque celle-ci a la possibilité d'impacter les performances de l'algorithme utilisé, et d'affecter négativement les interprétations du modèle³⁰.

Les fichiers images sont systématiquement accompagnés de métadonnées de diverses catégories, qui visent à décrire et identifier le document. Bien qu'il n'existe pas de standard international obligatoire concernant les métadonnées de numérisation, la standardisation des fichiers de métadonnées selon certains standards spécifiques permet de fluidifier la communication et les échanges entre institutions, dans une perspective interopérable qui permettrait de sortir des silos de données. Les métadonnées sont de plusieurs types : descriptives³¹, administratives³², structurelles³³; elles ont pour vocation de gérer tous ces aspects du document numérique. Ces métadonnées sont cruciales pour l'identification et la classification des documents numérisés, mais aussi pour permettre la recherche et la navigation à travers les collections. Malgré leur importance, les différentes institutions ont une gestion profondément hétérogène de leurs métadonnées, sans uniformisation nationale ou internationale, et avec une disparité profonde entre les différents types d'institutions patrimoniales.

Pour pallier à cette disparité dans la gestion des métadonnées et au manque de standards pour le partage des images en ligne, des initiatives internationales ont vu le

26. Un format décrit la manière dont les informations sont organisées dans un fichier.

27. Camille Besse, *Numérisation de Masse : Vers La Création d'un Nouvel Acteur de l'information. Le Projet Time Machine*, Paris, 2019.

28. La compression sans perte s'effectue par identification et suppression des redondances, c'est-à-dire des pixels identiques. Cette méthode n'est pas irréversible, il est possible de décompresser le fichier.

29. *Digital Images*, 2020, URL : <https://livecode.byu.edu/images/DigitalImages.php> (visité le 07/08/2023).

30. Austin C. Bergstrom et David W. Messinger, « Image Quality and Computer Vision Performance : Assessing the Effects of Image Distortions and Modeling Performance Relationships Using the General Image Quality Equation », *Journal of Electronic Imaging*, 32 (mars 2023), p. 023018, doi : 10.1117/1.JEI.32.2.023018.

31. Pour identifier le contenu du document numérique, le rattacher au document original.

32. Pour gérer les droits d'accès, préserver les informations techniques, suivre les modifications du fichier et garantir son intégrité.

33. Pour rattacher les fichiers entre eux et reconstituer la structure du document.

jour, visant à mettre fin à cette gestion en silo des données des institutions patrimoniales pour faciliter les échanges et la mise à disposition des données dans une optique de libre accès.

2.2 Le standard IIIF

2.2.1 IIIF et les données ouvertes

Principes et objectifs

Le standard IIIF³⁴ est une initiative collaborative lancée en 2011 par un groupement d'institutions³⁵ attachées à faciliter la diffusion sur le Web des ressources iconographiques des collections patrimoniales³⁶. La volonté de développement de ce standard provient du constat que, malgré la nécessité d'un accès global aux images numériques pour la transmission du patrimoine culturel, un grand nombre de ces ressources restent, sur Internet, isolées dans des silos d'informations dont l'accès n'est possible que par des applications spécifiques à chaque institution³⁷, rendant complexe le partage et la mise en commun des ressources.

De cette observation est née la volonté de créer un outil proposant des méthodes standardisées de partage sur le Web des images et de leurs métadonnées, pour toute institution qui souhaiterait partager numériquement ses collections. IIIF propose ainsi un standard dont les fonctionnalités vont au-delà de celles d'un navigateur, capable d'afficher des formats libres d'image³⁸, sans proposer plus d'interaction avec les images servies³⁹. IIIF propose ainsi des outils interopérables, libres d'accès, permettant notamment de zoomer en profondeur sur les images, d'y appliquer des traitements visuels simples, de reconstruire la structure du document ou de les annoter.

Pour assurer ces fonctionnalités, IIIF s'appuie sur deux API : l'API Image et l'API Présentation. Ces API sont le socle du standard, elles en assurent la cohérence afin que les images puissent être partagées de manière fluide entre les institutions et projets. Le cœur du standard IIIF, qui fait tout son intérêt dans le contexte décrit précédemment du partage des images sur le Web, est en effet l'interopérabilité : IIIF assure ainsi la portabilité

34. IIIF

35. Les collaborateurs à l'initiative du projet sont la British Library, la BnF, Stanford University, les Bodleian Libraries, la Bibliothèque nationale de Norvège, la Bibliothèque nationale de Los Alamos et Cornell University.

36. Tom Cramer, *The International Image Interoperability Framework (IIIF) : Laying the Foundation for Common Services, Integrated Resources and a Marketplace of Tools for Scholars Worldwide*, déc. 2011, URL : <https://www.cni.org/topics/information-access-retrieval/international-image-interoperability-framework> (visité le 08/08/2023).

37. *Ibid.*

38. IIIF propose également, depuis 2020, un service dédié aux fichiers vidéo, que nous n'abordons pas dans ce mémoire car les projets étudiés portent tous sur des images fixes.

39. *How It Works*, URL : <https://iiif.io/get-started/how-iiif-works/> (visité le 08/08/2023).

entre les visualiseurs, pour permettre une transmission réellement fluide de l'information.

Les API IIIF

Le fonctionnement du standard IIIF s'appuie sur deux éléments essentiels : l'envoi des images depuis les serveurs, et leur visualisation.

L'API Image définit les spécifications pour un service renvoyant une image en réponse à une requête HTTP (*Hypertext Transfer Protocol*) ou HTTPS (*Hypertext Transfer Protocol Secure*) : à partir d'une URI (*Uniform Resource Identifier*) valide, l'image peut être affichée dans un visualiseur, ou dans un navigateur. L'URI peut spécifier, dans sa requête, la région, la taille, la rotation, la qualité ou le format de l'image demandée, suivant des instructions spécifiques prédéfinies par le standard⁴⁰. En ajustant les paramètres, l'utilisateur peut ainsi obtenir une image qu'il peut modifier en accord avec ses besoins sans passer par un éditeur spécifique. L'API Image, développée pour des clients tels que les applications d'institutions patrimoniales, peut être implémentée sans l'API Présentation : seule, elle permet notamment un zoom rapide et profond sur des images en très haute résolution.

L'API Présentation permet d'introduire des fonctionnalités plus vastes en termes d'informations liées aux images : par le biais d'un manifeste au format JSON (*JavaScript Object Notation*), elle attache aux images des métadonnées descriptives, administratives et structurelles qui définissent la manière dont elles seront présentées dans un visualiseur. Le manifeste permet notamment de rassembler en un fichier les différents éléments d'un objet IIIF⁴¹, et notamment les séries d'images⁴², tout en gérant également leur ordre dans le document numérique. Le manifeste contient généralement des informations d'identification de l'objet original⁴³, et peut être enrichi d'annotations sur l'objet ou son contenu⁴⁴.

Pour afficher les images et les informations liées, il existe une multitude de visualiseurs⁴⁵ aux fonctionnalités différentes, tous compatibles avec tous les manifestes IIIF, quelle que soit leur institution d'origine. Ainsi, les ressources produites par les institutions peuvent être consultées par le biais d'outils libres, accessibles à tous. Pour aller au-delà des fonctionnalités proposées par les deux API centrales du standard, quatre autres API ont été développées, pour gérer d'autres besoins liés à mise à disposition d'images tout en respectant des principes standardisés⁴⁶.

40. *Image API 3.0*, URL : <https://iiif.io/api/image/3.0/> (visité le 08/08/2023).

41. *How It Works...*

42. Cette fonctionnalité est particulièrement intéressante pour les numérisations d'ouvrages, puisqu'elle permet de regrouper en un fichier toutes les pages numérisées.

43. On retrouve notamment le titre, l'auteur, l'institution de conservation, le numéro d'inventaire ou la cote, et les informations sur les droits de diffusion.

44. *Presentation API 3.0*, URL : <https://iiif.io/api/presentation/3.0/> (visité le 08/08/2023).

45. *IIIF Viewers*, URL : <https://iiif.io/get-started/iiif-viewers/> (visité le 08/08/2023).

46. *How It Works...*

Anatomie d'un manifeste

Pour constituer les éléments retournés par les API IIIF, il existe un modèle de données spécifique, qui assure la caractérisation des objets selon des normes communes, qui permettent de conserver l'interopérabilité du standard. Il existe ainsi des types de ressources, ou classes, précisément définies, avec une hiérarchie fixe (fig. 2.1), qui permettent de décrire les objets numérisés au format JSON.

Un manifeste est un fichier au format JSON. Du point de vue de la donnée, la classe *Manifest* permet de décrire un objet composé, une œuvre dans sa totalité : elle contient ainsi un ensemble de métadonnées descriptives la concernant (titre, auteur, institution de conservation, licence), et permet, si nécessaire, de reconstituer la structure de l'objet à partir des *Canvases*.

Le *Canvas* est un conteneur de niveau inférieur au *Manifest*, qui permet de représenter une vue spécifique d'un objet, et les métadonnées qui y sont associées. Si nécessaire, un élément de niveau supérieur peut exister entre les *Canvases* et le *Manifest* : il s'agit des *Ranges*, qui permettent de décrire une séquence de *Canvases*.

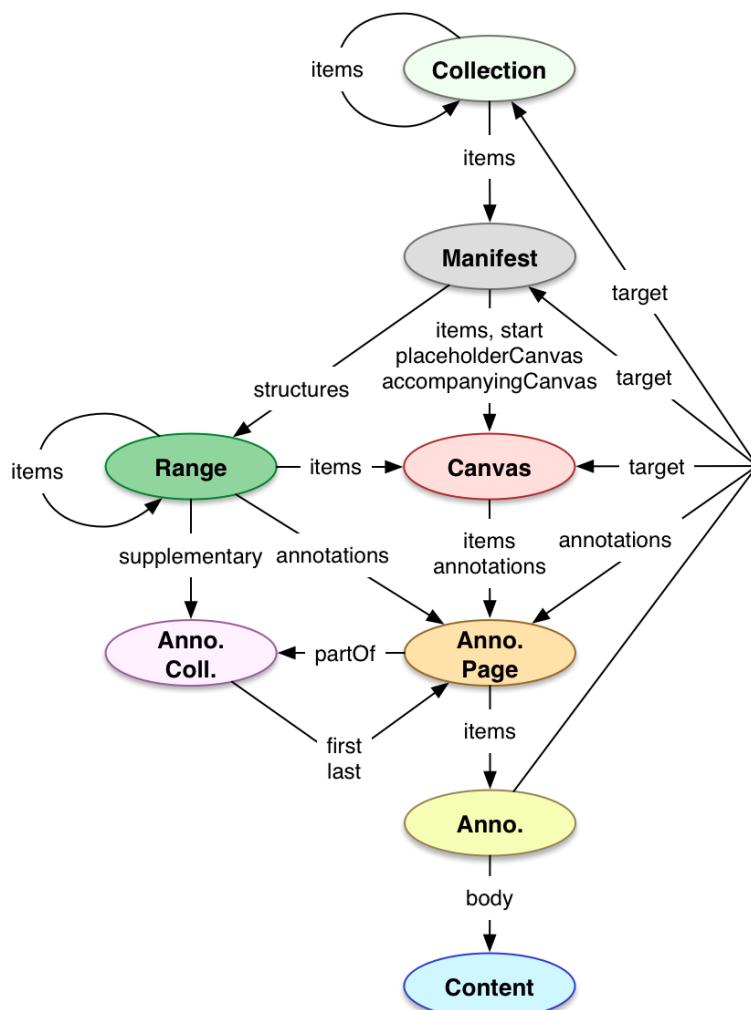


FIGURE 2.1 – Modèle de données IIIF pour la description d'un objet

Il existe, à un niveau supérieur au *Manifest*, une classe *Collection*, qui assure une plus grande flexibilité du modèle pour les projets qui souhaiteraient réunir plusieurs objets dans une catégorie plus large. Une *Collection* peut représenter une liste de *Manifests*, ou une liste de *Collections*, permettant ainsi de regrouper des objets ou groupes d'objets selon un cadre défini, qui facilite la navigation dans un ensemble restreint de ressources liées⁴⁷.

Pour un manifeste ne contenant que des images et leurs métadonnées, ces éléments suffisent à construire un fichier cohérent et fonctionnel, qui pourra être lu par un visualiseur IIIF.

Pour annoter les images d'un manifeste, il existe une série de types de ressource additionnels qui contiennent les annotations selon une hiérarchie précise, et les rattache à un *Canvas* spécifique. L'*Annotation Page* contient une liste ordonnée d'*Annotations* pour un *Canvas* : pour annoter un manifeste, il existe donc autant d'*Annotation Pages* qu'il existe de *Canvases* annotés.

La classe *Annotation* fait le lien entre le contenu de l'annotation (*Content*) et le *Canvas*⁴⁸ : ce type de ressource permet d'assurer l'interopérabilité du modèle, puisqu'elle permet l'alignement des annotations et des images, y compris dans le contexte d'annotations produites par un utilisateur différent du créateur du manifeste. Ainsi, ce modèle de données permet la collaboration entre plusieurs utilisateurs à partir d'un manifeste produit par une institutions, et rend possible la réutilisation des images comme des manifestes dans des projets ultérieurs, avec la fluidité d'un modèle partagé, standard, qui ne nécessite aucun effort d'alignement entre institutions.

2.2.2 IIIF, un modèle universel ?

Limites de l'interopérabilité du standard

IIIF est ainsi pensé comme un standard universel permettant aux institutions et aux projets qui le souhaitent de partager leurs ressources, de les mettre à disposition en assurant la possibilité de leur réutilisation : dans une optique de science ouverte, IIIF permet de partager aisément les résultats de projets de recherche impliquant des images.

Cependant, malgré l'existence d'un modèle de données standard pour la description des objets, IIIF présente des limites dans son universalité. En effet, malgré son cadre construit pour assurer l'interopérabilité des manifestes, le modèle n'est en pratique pas employé de la même manière par toutes les institutions, et nécessite donc une adaptation des développements faits autour de IIIF pour prendre en compte ces exceptions. Nous constatons, par exemple, que si les ressources images sont supposées être intégrées au manifeste dans un conteneur *Canvas*, certaines institutions – comme la bibliothèque univeristaire de

47. *Presentation API 3.0...*

48. *Ibid.*

Yale – utilisent plutôt des *Items* (fig. 2.2), modifiant ainsi profondément la structure du manifeste IIIF tout en conservant un fichier lisible par les visualiseurs. L’application du standard varie ainsi d’une institution à l’autre, sans altérer le fonctionnement des outils de base de IIIF, mais en provoquant cependant une déperdition de son universalité pour des projets qui souhaiteraient développer des outils autour de ces manifestes.

Dans un projet reposant sur l’exploitation d’images d’objets patrimoniaux, tels que les projets décrits dans ce mémoire, il est ainsi nécessaire de prendre en compte ces exceptions techniques : dans le cadre du projet EIDA, la récupération des images est effectuée par un algorithme spécifique⁴⁹, qui parse un manifeste IIIF déposé par un utilisateur dans l’application du projet et en extrait des fichiers images pour les enregistrer. Dans le développement de cet algorithme, il a ainsi été nécessaire de prendre en compte les exceptions possibles, pour ne pas créer d’erreur en cas de dépôt d’un manifeste à la structure différente.



```

    {
      "canvases": [
        {
          "0": {
            "@id": "https://iiif.slub-dresden.de/3548814/canvas/PHYS_0001",
            "@type": "sc:Canvas",
            "height": 2690,
            "width": 2000,
            "label": "Seite 1",
            "thumbnail": (...),
            "images": [
              {
                "0": {
                  "@type": "oa:Annotation",
                  "motivation": "sc:painting",
                  "on": "https://iiif.slub-dresden.de/3548814/canvas/PHYS_0001",
                  "resource": [
                    {
                      "0": {
                        "@id": "https://images.iiif.slub-dresden.de/iiif/2/data/2Fkito0s2Ftracvaa3_323548814_2Ftracvansa_323548814_tif2Fipas2F00000001.tif.large.jpg/full/full/0/default.jpg",
                        "gtypes": "dctypes:Image",
                        "format": "image/jpeg",
                        "height": 2690,
                        "width": 2000
                      }
                    }
                  ]
                }
              }
            ]
          }
        ]
      }
    }
  
```

(a) Manifeste IIIF de la bibliothèque universitaire de Dresde avec *Canvases*



```

    {
      "items": [
        {
          "0": {
            "type": "Canvas",
            "items": [...],
            "id": "https://collections.library.yale.edu/iiif/2/16969241/canvas/16969241",
            "label": (...),
            "thumbnail": [...],
            "rendering": [
              {
                "0": {
                  "id": "https://collections.library.yale.edu/iiif/2/16969241/full/full/0/default.jpg",
                  "label": (...),
                  "type": "Image",
                  "format": "image/jpeg"
                },
                "1": {
                  "label": (...),
                  "height": 3000,
                  "width": 2026
                }
              ],
              "metadata": [...]
            ]
          }
        ]
      }
    }
  
```

(b) Manifeste IIIF de la bibliothèque universitaire de Yale avec *Items*

FIGURE 2.2 – Comparatif des ressources contenant les images de deux manifestes IIIF

Implémentation de IIIF : un réflexe global ?

Pour les projets de recherche exploitant des images respectant le standard IIIF, il se pose la question des institutions ayant fait le choix de gérer leurs images différemment. En effet, si de nombreuses institutions ont adopté le standard et mis en ligne des centaines de

49. [Iiifdownloader](#).

millions d’images compatibles depuis la naissance de l’initiative⁵⁰, il n’est ni obligatoire ni systématique, et il reste ainsi un nombre certain d’établissements qui gèrent leurs images avec des outils techniques différents, et restent ainsi exclus du standard. Les musées, notamment, et particulièrement les musées hors Amérique du Nord, exploitent encore peu les possibilités offertes par IIIF en termes de partage des images⁵¹, et restent en marge de cette initiative de facilitation de la mise en commun et de l’exploitation des ressources iconographiques.

Il n’est cependant pas envisageable, dans le cadre d’un projet de recherche, d’exclure totalement les sources provenant d’institutions n’utilisant pas le standard IIIF : il est donc nécessaire, dans le développement d’outils numériques pour la gestion des images, de prendre en compte ces sources aux formats différents, qu’il faut traiter pour les rendre exploitable de la même manière que le sont les images extraites de manifestes IIIF. Ainsi, dans la chaîne de traitement des images, une réflexion doit être menée pour l’intégration d’autres formats, tels que les PDF pour les numérisations d’ouvrages, ou les fichiers image indépendants. Malgré la fluidité du traitement des sources permise par l’existence d’un standard, il est ainsi nécessaire de réfléchir à des solutions techniques au-delà des limites de ce dernier, pour ne pas exclure des corpus les sources numérisées présentant un intérêt pour la recherche.

Dans un projet de recherche en vision artificielle, l’utilisation du standard IIIF permet, d’une part, la constitution aisée d’un corpus d’images, qu’il est possible de récupérer directement sur les sites des institutions à l’aide des API : le standard IIIF fluidifie fortement les échanges et la réutilisation des documents numérisés, et permet de construire un outil générique pour le traitement de sources numériques provenant d’institutions diverses. Il reste cependant de nombreux cas où une chaîne de traitement parallèle doit être développée, pour pallier à l’absence d’implémentation du standard dans certaines institutions. D’autre part, l’utilisation du standard IIIF pour la publication en ligne des données du projet permet également d’envisager leur réutilisation future par d’autres initiatives, en s’assurant de l’interopérabilité des résultats produits, dans une démarche de science ouverte.

Les sources iconographiques sont soumises à un ensemble de restrictions, du point de vue du format, des métadonnées ou des droits, qui manquent encore d’une uniformité internationale et entre institutions qui rendrait fluide le partage de ces ressources sur Internet et entre les projets de recherche. Dans un projet impliquant l’utilisation d’algorithmes de vision artificielle, qui repose alors sur le traitement d’un volume important d’image, la mise en ligne de ces documents est un enjeux crucial, sur lequel repose d’une part la

50. Gavin Mallory, *IIIF for Museums, Explained*, juill. 2019, URL : <https://blog.cogapp.com/iiif-for-museums-explained-49fd0560e1ba> (visité le 09/08/2023).

51. *IIIF for Museums in France*, juill. 2023, URL : <https://www.culture.gouv.fr/en/Thematic/Museums/Pour-les-professionnels/Network/IIIF-for-museums-in-France> (visité le 09/08/2023).

possibilité de constituer un corpus exploitable, ainsi que la publication des résultats du projet, qui peuvent également prendre la forme d'images numériques. Dans une optique de science ouverte, l'utilisation de standards et d'outils tels que ceux développés par le consortium IIIF permet d'assurer le partage d'images et de données respectant les mêmes formats, exploitables avec des outils libres, et d'avancer vers une abolition des silos de données, qui faciliterait notamment la construction de corpus massifs pour l'apprentissage machine.

Chapitre 3

Corpus historiques et jeux de données pour l'apprentissage machine

La vision artificielle et l'apprentissage machine permet le traitement de corpus d'images massifs par des méthodes quantitatives qui permettent aux historiens de traiter un volume de données bien plus important qu'une approche manuelle, ouvrant ainsi la voie à de nouvelles approches. Cette partie revient sur les bonnes pratiques à mettre en place afin d'assurer la pertinence de l'utilisation de ces outils, et d'en faire des traitements efficaces en accord avec les ambitions des projets.

3.1 Dimensions et cadre

3.1.1 La tentative de l'exhaustivité

Représentativité et pertinence

L'approche computationnelle de corpus de sources historiques permet le traitement d'un volume d'images inenvisageable sans automatisation, qui redéfinit ainsi le rôle des chercheurs comme les méthodes de constitution d'un corpus, pour prendre en compte cette quantité de données qu'il est possible de considérer à l'aide d'outils numériques¹. L'apprentissage profond permet d'envisager une redéfinition des méthodes des historiens et historiens de l'art, en intégrant une part d'automatisation dans le traitement des sources iconographiques, permettant ainsi d'envisager l'exploration de corpus bien plus vastes²

1. Harald Klinke, « Big Image Data within the Big Picture of Art History », *International Journal for Digital Art History*-2 (oct. 2016), DOI : 10.11588/dah.2016.2.33527.

2. Eleonora Moiraghi et Jean-Philippe Moreux, *Explorer des corpus d'images. L'IA au service du patrimoine*, Billet, avr. 2018, URL : <https://bnf.hypotheses.org/2809> (visité le 06/08/2023).

par des méthodes quantitatives³.

Dans le cadre du projet EIDA, ces possibilités ouvertes par l'intégration de techniques de vision artificielle dans le traitement du corpus permet ainsi d'envisager des bornes larges, définies précédemment, sur le plan géographique comme chronologique, tout en assurant un traitement d'un grand nombre de sources – plusieurs centaines ou milliers d'images – pour chaque contexte étudié.

Le projet VHS s'appuie également grandement sur la vision artificielle appliquée à la détection de similarité, dans le cadre d'une étude de la circulation des savoirs scientifiques par le biais de l'illustration : les corpus constitués pour le projet ont donc été pensés pour offrir un regard pertinent sur ces questions, et sélectionnés avec une volonté de représentativité géographique, chronologique et thématique⁴. Leur analyse portera ainsi sur un ensemble de plus de 10000 images d'animaux, de plantes et de minéraux.

Ainsi, l'intégration au cadre des projets de méthodes de vision artificielle permet d'envisager l'étude de corpus larges, vastes, et de porter un regard sur des zones géographiques larges, sur des périodes étendues, avec la possibilité de traiter des corpus massifs par des approches quantitatives développées en collaboration avec les historiens.

Données d'entraînement

Au-delà de la pertinence du corpus pour un projet de recherche donné, il est nécessaire, pour l'entraînement d'un modèle de vision artificielle, de prévoir un fragment du corpus dédié à la constitution d'un ou de plusieurs jeux de données d'entraînement⁵. La représentativité est un élément clé de ces corpus restreints, qui doivent à leur échelle comporter suffisamment de cas d'études différents pour englober les situations rencontrées dans le corpus global, et ainsi produire un modèle apte à appréhender toutes les situations rencontrées dans un contexte d'inférence. Un dialogue entre les chercheurs et les ingénieurs est alors nécessaire, pour établir les besoins d'un point de vue technique comme d'un point de vue scientifique, afin d'obtenir en finalité un outil performant, apte à traiter le corpus du projet. Les chercheurs ayant une connaissance scientifique de la typologie des sources rencontrées dans le cadre du projet, ils sont ainsi à même d'estimer la diversité des images que rencontrera le modèle, et donc de construire un jeu de données d'entraînement représentatif et pertinent, en accord avec les bornes définies du projet.

Le projet EIDA, dont le cadre géographique s'étend de l'Europe à l'Asie, du VIII^e au XVIII^e siècle, est ainsi vaste en termes de sources historiques et de diagrammes repré-

3. H. Klinke, « Big Image Data within the Big Picture of Art History »...

4. *Corpus*, URL : <https://vhs.hypotheses.org/corpus> (visité le 09/08/2023).

5. Un jeu de données d'entraînement comptant en général plusieurs centaines d'exemples, il est possible, à cette étape, d'évaluer la pertinence véritable du développement d'un modèle dédié au traitement du corpus global du projet. Daniel van Strien, Kaspar Beelen, Melvin Wevers, Thomas Smits et Katherine McDonough, « Computer Vision for the Humanities : An Introduction to Deep Learning for Image Classification (Part 1) », *Programming Historian* (, août 2022), URL : <https://programminghistorian.org/en/lessons/computer-vision-deep-learning-pt1> (visité le 10/08/2023)

sentés. Les jeux de données d’entraînement doivent être représentatifs d’un point de vue thématique, en proposant des diagrammes aux formats et apparences diverses, en accord avec les typologies variées retrouvées dans le corpus ; mais ceux-ci doivent aussi prendre en compte la diversité des supports. Le corpus d’EIDA comporte en effet des sources manuscrites comme des sources imprimées, qu’il est donc nécessaire de traiter conjointement. Il a été envisagé, pour des questions de performance, de produire un modèle de vision entraîné spécifiquement sur des sources manuscrites, et un second modèle entraîné sur des sources imprimées, afin de dissocier totalement le traitement de ces deux supports dans l’application finale⁶. Dans un corpus aussi varié linguistiquement que celui du projet EIDA, il est également crucial de sélectionner un jeu de données d’entraînement diversifié du point de vue des langues, pour s’assurer de l’efficacité du modèle sur des sources aux provenances diverses : nous constatons en effet que les premières détections faites avec des modèles de détection pré-entraînés⁷ sont satisfaisantes sur des sources latines ou grecques, mais qu’ils performent peu efficacement sur des sources chinoises, où chaque idéogramme est détecté comme une image.

Ainsi, l’automatisation d’une partie de la chaîne de traitement des sources permet, de construire un corpus de recherche vaste en offrant l’opportunité d’analyser un volume massif de sources historiques : ces possibilités sont particulièrement intéressante dans le cadre de projets de recherches inscrits dans l’histoire de la circulation des idées, des images et des théories, puisque les corpus construits peuvent être représentatifs de multiples cadres géographiques, temporels et thématiques, sans craindre un surplus de sources à traiter. Pour construire des outils pertinents pour le traitement de ces corpus massifs, il est cependant nécessaire d’avoir un regard global sur leur contenu, pour reproduire à moindre échelle leur représentativité dans des jeux de données d’entraînement, nécessaires au développement de modèles de vision artificielle pertinents pour des sources étudiées.

3.1.2 Automatiser le traitement de corpus massifs

Possibilités de la *computer vision*

La vision artificielle offre, pour le traitement de ces larges corpus, la possibilité d’automatiser certaines étapes spécifiques qui interviennent en parallèle d’étapes d’analyse par les chercheurs. Il n’existe ainsi pas de *workflow* totalement automatisée, où l’historien n’interviendrait pas dans le traitement des sources étudiées. En effet, l’intégration de la vision artificielle aux méthodes des historiens n’existe que conjointement à des interventions humaines, qui assurent la pertinence des traitements effectués et apportent une analyse nécessaire.

6. L’entraînement des modèles de détection d’objet n’ayant pas encore été effectué, il est actuellement impossible de tirer des conclusions quant à la pertinence de ce choix.

7. Nous traitons de ces modèles dans la partie II.1.2.

L'une des premières applications de la vision par ordinateur exploitée dans les projets étudiés est la détection d'objets dans les images : pour le traitement de sources numérisées, la détection d'objets présente des applications diverses. Il s'agit, le plus souvent, de la première étape d'une chaîne de traitement ; permettant, par exemple, dans l'étude de sources manuscrites ou imprimées contenant du texte et des illustrations, d'extraire les images présentes dans ces sources⁸. Dans les projets EIDA et VHS, cette étape de détection des images dans les ouvrages est faite dans une optique de segmentation des pages, pour en extraire les illustrations qui seront ensuite traitées par d'autres algorithmes de vision, tels que des algorithmes de détection de similarité ou de vectorisation. D'autres projets, cependant, proposent dès l'étape de détection une classification des illustrations par typologie, pour permettre une première analyse et un premier regard sur la place de ces objets détectés dans les sources étudiées : l'application *CorDeep*⁹, développée par le Max Planck Institute for the History of Science en partenariat avec BIFOLD pour l'extraction d'éléments visuels dans les sources historiques, propose de classifier les illustrations détectées en quatre catégories¹⁰, appliquant dès la détection un premier traitement pour l'analyse des illustrations par les chercheurs, qui peuvent ainsi plus aisément naviguer des corpus de numérisations d'ouvrages par le biais des images.

La détection de similarité compte parmi les applications les plus directes du *deep learning*¹¹ : il est possible, à l'aide d'un modèle entraîné sur un jeu de données restreint, d'effectuer sans supervision des comparaisons entre toutes les images du corpus, pour les réunir en séries ou groupes qui seront ensuite analysés par les historiens. Le développement d'un tel outil exemplifie les possibilités offertes par l'intelligence artificielle pour la navigation de gros corpus : elle permet de classifier les images en vue de leur étude en constituant des séries iconographiques aux caractéristiques visuelles similaires, selon un score de similarité calculé par l'algorithme. Dans le cas d'une étude des circulations des illustrations scientifiques, la constitution de ces séries permet aux chercheurs d'effectuer une analyse sur l'évolution des théories scientifiques, par le regroupement d'images de traditions différentes présentant des éléments similaires. L'automatisation de cette étape permet ainsi de tracer des parallèles entre un grand nombre d'image, en lançant la détection sur des corpus très larges qu'il serait difficile de traiter manuellement.

La *computer vision* permet également d'envisager des traitements pour l'édition, et particulièrement dans le cas d'objets scientifiques tels que les diagrammes, dont la mise en forme est un questionnement à part entière pour l'édition des textes qu'ils illustrent. La détection de contours (ou *edge detection*) est une étape fondamentale pour le traitement

8. Jochen Büttner, Julius Martinetz, Hassan El-Hajj et Matteo Valleriani, « CorDeep and the Sacrobosco Dataset : Detection of Visual Elements in Historical Documents », *Journal of Imaging*, 8–10 (oct. 2022), p. 285, DOI : 10.3390/jimaging8100285.

9. *CorDeep*, URL : <https://cordeep.mpiwg-berlin.mpg.de/> (visité le 10/08/2023).

10. *Content Illustrations, Initials, Decorations et Printer's Marks*.

11. E. Moiraghi et J.P. Moreux, *Explorer des corpus d'images. L'IA au service du patrimoine...*

des images en vision artificielle : combinée à des méthodes de détection des lignes¹², il devient envisageable d'automatiser la transformation d'images au format .tiff ou .jpeg en objets plus aisément manipulables, et notamment la transformation vers le SVG (*Scalable Vector Graphics*), qui fait alors des diagrammes des objets édités et éditables, exploitables par les chercheurs en tant qu'objets numériques.

Diversité des traitements, diversité des données

Pour ces divers traitements exploitant la vision artificielle, l'étape d'entraînement est nécessaire pour l'obtention d'un modèle efficace sur les données du projet, et pour que celui-ci performe en accord avec les tâches qui lui sont demandées. L'entraînement d'un modèle nécessite généralement plusieurs étapes¹³, et par conséquent autant de jeux de données qu'il y a d'étapes : il est de bonne pratique de prévoir un jeu de données initial, et d'évaluer les performances du modèle après ce premier entraînement. À partir de ces résultats, il devient possible d'adapter les jeux de données suivants, pour pallier aux faiblesses constatées. De plus, si le format des données d'entraînement nécessaires varie en fonction des tâches effectuées par le modèle¹⁴, les exigences en termes de volume et de représentativité varient également.

Pour des tâches classiques de vision artificielle, telles que la détection d'objets, des modèles pré-entraînés performants existent en libre accès, qui ne nécessitent donc pas autant de travail d'entraînement qu'un modèle créé dans son entiereté pour un projet. Des jeux de données en accès libre sont également disponibles pour l'entraînement de modèles pour la détection automatique, tels que ImageNet¹⁵, qui bien qu'ils ne soient pas adaptés à des sources historiques, permettent d'effectuer un premier entraînement d'un modèle sans mobiliser les moyens nécessaires à la création d'un jeu de données. Les algorithmes de détection de similarité ou de vectorisation, cependant, ne sont pas aussi développés que ces algorithmes de détection d'objets, et nécessitent donc un volume de données plus important, que les projets se doivent de fournir pour la création de modèles performants.

Ainsi, pour un projet d'humanités numériques faisant appel aux méthodes de la vision artificielle, il est nécessaire d'apporter une réflexion, en premier lieu, sur la pertinence de l'usage de l'intelligence artificielle pour le traitement du corpus – pertinence souvent relative au volume de données à traiter. Il est ensuite nécessaire, au-delà des sources du projet, de penser les données pour le *deep learning*, en prenant en compte l'importance

12. Xinyu Lin, Yingjie Zhou, Yipeng Liu et Ce Zhu, *A Comprehensive Review of Image Line Segment Detection and Description : Taxonomies, Comparisons, and Challenges*, avr. 2023, DOI : 10.48550/arXiv.2305.00264, arXiv : 2305.00264 [cs].

13. D. van Strien, K. Beelen, M. Wevers, *et al.*, « Computer Vision for the Humanities... ».

14. Les cas varient en fonction du format des données d'entrée et de sortie : images au format .jpeg et annotations au format .txt pour la détection, images au format .jpeg et images au format .svg pour la vectorisation, etc.

15. *ImageNet*, URL : <https://www.image-net.org/> (visité le 10/08/2023).

d'avoir des jeux de données qualitatifs et représentatifs pour la création d'un modèle efficace pour l'automatisation des tâches souhaitées. Les formats, typologies, volumes de ces jeux de données d'entraînement, souvent multiples, est ainsi à considérer en collaboration entre les équipes d'ingénierie et de recherche, pour prendre en considération les besoins vis-à-vis du contenu du corpus, tout en ne perdant pas de vue les besoins techniques liés à l'entraînement d'un modèle de vision par ordinateur.

3.2 Objectifs scientifiques et possibilités numériques

3.2.1 Dialogue entre chercheurs et ingénieurs

Concilier les besoins

L'intégration de traitements employant l'intelligence artificielle a des projets de recherche en humanités implique une redéfinition des méthodes, pour adapter les pratiques des chercheurs aux nouveaux besoins soulevés par ces techniques. Dans la section précédentes, nous avons établi les nécessités et possibilités liées aux corpus de recherches, mais il est également crucial de prendre en compte les échanges entre les acteurs divers impliqués dans de telles initiatives. Les projets étudiés dans le cadre de ce mémoire – notamment les projets EIDA et VHS– présentent une architecture similaire en termes d'équipes impliquées : il est, d'une part, une équipe d'historiens¹⁶ menant des recherches sur les thèmes étudiés, travaillant en collaboration avec une équipe de chercheurs en vision artificielle¹⁷. Les équipes d'histoire des sciences comptent, dans le cas de VHS et EIDA, un ou plusieurs ingénieurs, chargés de développement spécifiques et d'établir la communication avec les équipes de vision.

Les deux équipes de recherche impliquées dans les projets font ainsi face à des besoins variés, qui ne sont pas nécessairement alignés. La communication établie entre ces deux pôles permet de clarifier les besoins en termes de traitement des sources historiques, par opposition aux besoins en termes de techniques, et particulièrement de définir les limites de ces méthodes et techniques numériques appliquées à l'histoire. Il se dessine alors une tension entre les attentes vis-à-vis des modèles de vision, et les possibilités réelles de la technique, qu'il est crucial de prendre en compte pour le développement d'outils à la fois performants et pertinents.

16. Composée de chercheurs permanents, de chercheurs affiliés, de chercheurs postdoctorants et de doctorants.

17. Dans les deux cas cités, il s'agit de l'équipe dirigée par Mathieu Aubry, appartenant au groupe de recherche IMAGINE.

Utilité pour le *machine learning* et intérêt historique

Ces tensions entre les besoins se manifestent dès les premières étapes de développement d'un modèle de vision artificielle : dans le cas du projet EIDA, porté sur la détection de diagrammes dans les manuscrits, la confrontation entre l'aspect binaire de la détection et les nuances que doivent présenter l'approche historique se présente dès l'étape de constitution d'un premier jeu de données d'entraînement. En effet, cet aspect binaire du modèle, qui s'appuie sur l'opposition entre ce qui est un diagramme et ce qui n'en est pas un – et plus largement, ce qui est une objet à détecter et ce qui n'en est pas un – ne peut prendre en compte les cas limites qui se présentent systématiquement avec des sources historiques provenant d'un corpus aussi large. Se pose ainsi la question de la conciliation de ces deux exigences visiblement opposées : il est nécessaire de décider, par une discussion entre l'équipe de vision artificielle et l'équipe d'histoire des sciences, de ce qui est bel et bien un diagramme, selon les possibilités techniques et les besoins historiques.

Une distinction se crée entre les pratiques utiles pour le *deep learning* et les éléments intéressants pour les historiens. L'objectif, pour les deux équipes, est en effet de construire un outil utile aux résultats exploitables ; les projets ont donc pour devoir de prendre en compte ces besoins divergents, et d'établir un juste milieu entre les possibilités offertes et les objectifs initiaux.

Ainsi, dans cette situation, un équilibre est défini entre performances du modèle et pertinence historique, en prenant en compte le fait qu'il existe, après la détection, une intervention humaine : aucune solution n'étant pleinement satisfaisante, il convient de décider de celle qui permet de concilier les possibilités techniques et les objectifs scientifiques, afin de créer un outil qui produit des données exploitables, et qui automatise de manière pertinente le traitement des sources. Dans cette démarche, il est nécessaire de donner un cadre strict aux pratiques pour assurer une uniformisation des données, premier critère pour le développement d'outils numériques efficaces aux résultats cohérents.

3.2.2 Définir les pratiques

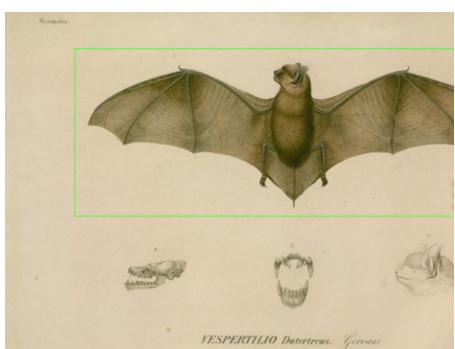
Traiter les cas limites

Il existe autant de manière de gérer ces situations qu'il existe de projet : les pratiques sont à définir en accord avec les besoins de chaque projet, sans qu'il n'existe de solution universelle. Les normes à appliquer sont à établir par un dialogue prenant en compte les ambitions techniques et scientifiques de chaque projet, ainsi que les spécificités des sources étudiées, qui présenteront des caractéristiques spécifiques propres à chaque corpus.

En amont de la constitution du jeu de données pour l'entraînement du modèle de détection des diagrammes astronomiques du projet EIDA, cette discussion entre l'équipe de l'Observatoire et celle du laboratoire IMAGINE a mené à la décision d'annoter tout type

de diagramme, ainsi que les illustrations retrouvées dans les manuscrits, même lorsqu'elles ne correspondent pas aux objets recherchés. En effet, l'exclusion d'illustrations telles que les miniatures dans les données d'entraînement – même si ces dernières ne correspondent pas aux objectifs scientifiques du projet – pourrait produire un modèle aux détections lacunaires. La solution choisie l'a été en prenant en considération le fait qu'il est plus aisé pour un historien qui analyserait les images détectées d'exclure les illustrations qui ne sont pas des diagrammes, plutôt que de rechercher dans la numérisation des diagrammes manqués par un modèle aux performances insatisfaisantes.

La création d'un premier jeu de données pour l'entraînement d'un modèle dédié à la détection des illustrations dans le corpus de VHS a permis de mettre en lumière un certain nombre de cas limites, ou *edge cases*, qui correspondent techniquement aux pratiques d'annotation définies par le projet, mais qui néanmoins impacteraient négativement les performances du modèle. Ainsi, nous voyons dans les annotations réalisées par les chercheurs de VHS des pratiques à modifier (fig. 3.1). Dans le premier cas, toutes les illustrations de la page n'ont pas été annotées : seule l'image principale de la planche zoologique est annotée comme un objet à détecter, tandis que les plus petites images sont délaissées. Comme mentionné précédemment, l'aspect manichéen de la détection par vision artificielle de permet pas de hiérarchiser ainsi les images considérées intéressantes pour la recherche, et celles qui ne le sont pas, il est donc nécessaire d'annoter l'ensemble des illustrations pour ne pas impacter négativement les performances du modèle, et risquer une détection lacunaire. Dans les deuxièmes et troisièmes cas, les images annotées n'auraient pas dû l'être, car si la mauvaise qualité de la numérisation ou l'aspect fragmenté du document n'empêche pas un observateur humain d'identifier les illustrations comme telles, il est une part de reconstitution des pans manquants de l'image qui tient de l'interprétation, et qui ne peut donc être attendue d'un modèle de vision artificielle. L'annotation de ce type d'objets pour l'entraînement du modèle risquerait ainsi d'altérer ses performances, en le poussant à détecter comme des illustrations des éléments qui n'en seraient pas.



(a) Annotation incomplète des illustrations



(b) Annotation d'un document mal numérisé



(c) Annotation d'un document lacunaire

FIGURE 3.1 – Cas limites rencontrés lors de l'annotation des images du projet VHS

Ces décisions quant à l'annotation sont ainsi prises suite à un dialogue entre les équipes de chercheurs en vision et d'historiens, pour trouver un juste équilibre entre attentes et possibilités réelles. Des choix sont faits pour tirer parti des méthodes de vision artificielle, tout en acceptant que leurs résultats ne seront jamais identiques à ceux d'un travail manuel : les décisions doivent ainsi être prises pour exploiter de manière pertinentes ces techniques. Les pratiques définies par ces discussions entre les équipes mènent à des normes d'annotation : des spécifications sont alors rédigées pour les établir, et s'assurer de leur application uniforme, pour la création de données cohérentes.

Guides des pratiques : établir les normes

Pour s'assurer de l'application uniforme des normes d'annotation, il est de bonne pratique de rédiger une documentation à destination des chercheurs, à laquelle chacun peut se référer lors des étapes de création des données d'entraînement, qui constituent en elles-mêmes un travail considéré comme travail de recherche, puisqu'il requiert une connaissance et une compréhension des sources. Ainsi, les données produites suivent les mêmes règles, et cette cohérence assure une uniformité de jeu de données fourni au modèle de vision : cette uniformité permet, s'il une nouvelle étape d'entraînement est nécessaire, de corriger plus facilement les choix ayant mené à des résultats non-satisfaisants.

Les chercheurs du projet EIDA se sont vus fournir un *Annotation Tutorial* en anglais, rédigé par Ségolène Albouy, cheffe de projet numérique, et Jil Le Bois, stagiaire de licence. Cette documentation fait suite à la mise en place d'une chaîne de traitement automatique des sources déposées par les chercheurs sur l'application du programme, qui produit une première détection que les chercheurs ont ensuite pour devoir de corriger. Le tutoriel qui a été communiqué contient des explications sur l'annotation dans l'application, avec un pas-à-pas détaillant chaque étape de ce travail, ainsi qu'une liste de cas pratiques établie en communication avec les historiens et l'équipe du laboratoire IMAGINE.

Les cas pratiques établis répondent – le plus souvent – à des questions posées par les historiens à l'équipe d'ingénierie à l'occasion d'ateliers ou de séminaires, qui permettent de souligner au préalable les possibles doutes qui seront rencontrés lors de l'annotation. L'*Annotation Tutorial* permet de garder trace de ces questionnements, et d'y apporter une réponse dont l'application sera systématique. La production de cette documentation, dont la responsabilité incombe aux ingénieurs d'étude, en tant que vecteurs de la communication entre les équipes d'histoire et de vision, pour donner à chaque parti impliqué la possibilité de se référer à un document qui rappelle les décisions prises. Cette possibilité de se référer à un document exhaustif est importante pour les étapes d'annotation, puisqu'elle garantit des données uniformes, mais aussi lors des étapes postérieures, notamment si d'autres entraînements sont nécessaires pour affiner les performances du modèle.

Le *deep learning* et la vision artificielle permettent aux projets de recherche en

histoire et en histoire de l'art d'envisager de nouvelles approches des sources, à l'aide de traitement automatisés qui offrent de nouvelles méthodes de navigation de corpus d'images massifs : de la détection à l'édition, les outils produits redéfinissent les étapes de traitement des sources, et il est ainsi nécessaire d'intégrer aux pratiques des chercheurs des méthodes spécifiques à ce type d'approche, et notamment pour la création de jeu de données d'entraînement, à la base du développement de tout modèle de *machine learning*. Cette intégration passe notamment par la rédaction de documentation, ainsi qu'un dialogue entre les équipes de recherche en vision artificielle et les équipes d'historiens. Ce dialogue permet d'établir les besoins de chacun, et de développer des outils techniques qui répondent aux besoins scientifiques de manière pertinente, en développant des modèles qui, malgré leurs limites, rejoignent aux mieux les attentes des sciences historiques.

Deuxième partie

**De l'image à l'objet : intégrer
l'apprentissage profond au
traitement des sources historiques**

Chapitre 4

Principes et utilisation de l'apprentissage profond

L'apprentissage profond est un sous-domaine de l'apprentissage automatique basé sur l'apprentissage de couches successives de représentation. Le nombre de couches définit la profondeur du modèle : de nos jours, l'apprentissage profond compte plusieurs dizaines à plusieurs centaines de couches, qui apprennent toutes automatiquement à l'aide de données d'apprentissage¹. Cette approche est au cœur des modèles de vision artificielle dont nous parlons dans ce mémoire, qui reposent sur des réseaux de neurones qui constituent ces couches superposées permettant un apprentissage des représentations à partir de données fournies.

4.1 Réseaux de neurones et *computer vision*

4.1.1 Des « neurones » pour le *deep learning*

Les modèles de vision artificielle pour des tâches telles que la détection d'objets, la détection de similarités ou la détection de lignes reposent souvent, en leur cœur, sur des réseaux de neurones plus ou moins profonds. Un réseau de neurones artificiels est un modèle paramétrique plus ou moins complexe² composé d'une couche d'entrée recevant les données brutes, puis d'une ou de plusieurs couches de calcul traitant les données en se corrigent mutuellement, et d'une couche de sortie proposant une prédiction à partir des données d'entrée et des calculs effectués. Un réseau de neurones est dit « profond » lorsqu'il compte un nombre suffisant, variable, de couches³. La prédiction se fait à l'aide d'un système de poids, qui correspondent aux paramètres d'une couche pour la transformation qu'elle applique aux données d'entrée⁴, et qui a pour objectif de calculer la prédiction la

1. François Chollet, *L'apprentissage profond avec Python*, Saint-Cyr-sur-Loire, 2020.

2. Chloé-Agathe Azencott, *Introduction au Machine Learning*, Malakoff, 2018.

3. *Ibid.*

4. F. Chollet, *L'apprentissage profond avec Python...*

plus juste, c'est-à-dire d'associer l'entrée avec une cible⁵ par une série de transformations. L'algorithme corrige ses poids en cas de prédiction fausse : il s'agit de l'apprentissage, qui signifie donc la recherche d'un ensemble de valeurs pour les poids de toutes les couches du réseau de sorte que les résultats obtenus soient satisfaisants.

4.1.2 Réseaux de neurones à convolution

Les modèles mis en avant dans ce mémoire de recherche reposent tous plus précisément sur des réseaux de neurones à convolution, ou CNN (Convolutional Neural Nets), développés pour reconnaître des motifs visuels dans des images avec un minimum de traitements appliqués. Ces derniers sont favorisés pour les tâches de vision par ordinateur, et pour les tâches de perception en général⁶. Ces réseaux sont composés de deux types de neurones agencés en plusieurs couches : les neurones de traitement, dédiés à traiter chacun une portion de l'image, et les neurones de mise en commun des sorties, dits de *pooling*⁷. Les couches de convolution⁸ ont pour spécificité d'être basées sur des fragments qui ne représentent que quelques pixels des images d'origine⁹ : elles apprennent des motifs locaux, à l'inverse des couches entièrement connectées qui apprennent des motifs globaux. Les CNN ont besoin de moins d'exemples d'apprentissage, parce qu'ils ont la capacité d'apprendre un motif et de le reconnaître quelle que soit sa position. Les CNN requièrent donc un volume de mémoire moins important, pour une efficacité supérieure¹⁰.

Les CNN ont la capacité d'apprendre des motifs locaux, puis d'apprendre dans les couches suivantes des motifs plus grands qui en découlent, apprenant ainsi des concepts visuels de plus en plus abstraits, et de plus en plus complexes¹¹ : l'intérêt dans le cadre d'un apprentissage pour la vision artificielle réside donc dans cette possibilité d'apprendre des motifs invariants par translation¹² et spatialement hiérarchiques, deux caractéristiques fondamentales du monde visuel¹³.

5. Pour un algorithme de détection d'images dans les manuscrits, l'entrée est l'image de la page, et la cible est l'étiquette « Illustration » qui sera attribuée à la zone de l'image où une illustration est détectée.

6. *Ibid.*

7. Ian Goodfellow, Yoshua Bengio et Aaron Courville, *Deep Learning*, 2016, URL : <http://www.deeplearningbook.org>.

8. Une couche de convolution est une couche constituée de copies d'un même neurone qui ne prend en compte qu'une partie de l'entrée.

9. Ces motifs locaux peuvent être les bords, les textures, et d'autres éléments. F. Chollet, *L'apprentissage profond avec Python...*

10. I. Goodfellow, Y. Bengio et A. Courville, *Deep Learning...*

11. F. Chollet, *L'apprentissage profond avec Python...*

12. « L'invariance par translations est une propriété fondamentale partagée par la quasi-totalité des opérations de traitement d'images. Elle exprime le fait qu'une information visuelle sera traitée de la même façon, quelle que soit sa localisation dans l'espace. En effet, si on bouge le capteur, chaque objet sera déplacé dans l'image, mais devra être traité de la même façon que précédemment. » Christian Ronse, *Invariance Par Translations*, URL : <https://dpt-info.u-strasbg.fr/~cronse/TIDOC/SYM/invtr.html> (visité le 15/08/2023)

13. F. Chollet, *L'apprentissage profond avec Python...*

4.2 Modèles de détection *off-the-shelf* : outils libres pour l'extraction d'objets

4.2.1 YOLOv5, docExtractor et autres modèles

Utilisation d'un modèle pré-entraîné

Il est courant, pour la vision artificielle, d'utiliser un modèle pré-entraîné sur un grand jeu de données peu spécifique, et de l'affiner à partir d'un plus petit jeu de données correspondant précisément aux attentes et applications du modèle dans le projet¹⁴. En effet, un modèle entraîné sur un jeu de données de base suffisamment large et généraliste apprend des caractéristiques qui peuvent être appliquées à l'ensemble du monde visuel, et peut ainsi servir de modèle générique pour de nombreux problèmes de vision par ordinateur¹⁵, même si ces derniers sont éloignés de la tâche initiale. La portabilité des caractéristiques apprises rend ainsi l'usage du *deep learning* pertinent dans le cadre de projets de recherche sur des sources historiques, puisqu'elle assure ainsi l'efficacité des modèles créés malgré les possibles limites des jeux de données disponibles : il est ainsi possible d'entraîner un modèle sur un jeu de données limité, qui correspond à la réalité matérielle des sources étudiées.

ImageNet

ImageNet¹⁶ est une initiative ayant pour objectif de fournir en accès libre un vaste jeu de données images pour la recherche en vision artificielle. Cette initiative est née du besoin crucial de données d'entraînement et de validation pour le *machine learning*, et particulièrement pour la classification, tâche de base de la vision par ordinateur, qui requiert un volume important d'images pour l'obtention de bonnes performances. Le jeu de données ImageNet compte plus d'un million d'images, catégorisées selon des concepts récupérés du projet WordNet qui vise à répertorier et classifier le contenu sémantique et lexical de la langue anglaise. Chaque image du jeu de données est annoté manuellement pour indiquer les objets présent dans l'image : ImageNet est un projet participatif, ce qui permet d'assurer la continuité de son enrichissement. Le projet ne possède pas les droits des images du jeu de données, formulant simplement une liste d'images disponibles sur le Web pour chaque concept WordNet : elles sont mises à disposition des projets à visée non-commerciale, pour assurer aux chercheurs la possibilité de mener des recherches en ayant accès à un jeu de données d'entraînement riche et vaste, permettant de développer des modèles performants avant un entraînement plus spécialisé.

14. Nous nous focaliserons spécifiquement sur les modèles de détection et de classification, dont la question de l'entraînement a été abordée lors du stage.

15. *Ibid.*

16. *ImageNet...*

yolov5 : un modèle de pointe pour l'extraction d'objets

YOLOv5¹⁷ est la cinquième version du modèle de détection d'objets et de segmentation d'images YOLO (*You Only Look Once*), développé à l'Université de Washington Joseph Redmon et Ali Farhadi et lancé en 2015. YOLOv5, publié en 2020, est un modèle réputé pour sa rapidité d'exécution et sa précision¹⁸, pré-entraîné sur le jeu de données MS COCO¹⁹, dont l'implémentation se veut aisée : YOLOv5, contrairement à ses prédecesseurs, est directement implémenté dans PyTorch²⁰, permettant une intégration facile à un environnement de développement, puisqu'il nécessite moins d'adaptation que les versions précédentes fonctionnant avec *framework* Darknet²¹, basé sur le langage C²². Le réseau neuronal de YOLO fonctionne en trois étapes : l'extraction de caractéristiques des données d'entrée, l'agrégation de ces caractéristiques, puis la résolution du problème (dans ce cas précis, la détection des objets)²³. Le modèle est disponible en accès libre, et constitue ainsi une base solide pour les projets souhaitant utiliser un modèle de détection d'objets : YOLOv5 peut, en effet, être entraîné sur un jeu de données choisi, pour en affiner la détection et l'appliquer à des images plus spécifiques, telles que des pages d'ouvrages.

Les projets EIDA et VHS entraînent, pour la détection d'illustrations dans les numérisations d'ouvrages, des modèles ayant pour base YOLOv5. Sans entraînement spécifique, en s'appuyant exclusivement sur le pré-entraînement fait avant la mise en ligne du modèle, les performances sont peu satisfaisantes : les jeux de données utilisés pour ce pré-entraînement, qu'il s'agisse d'ImageNet ou de MS COCO, sont en effet des jeux de données d'images réelles, faits pour l'apprentissage de la classification d'objets du quotidien, et ne sont donc initialement pas adaptés à la segmentation de pages de manuscrits ou à la détection d'illustrations. Il faut ainsi compter sur les propriétés de portabilité de l'apprentissage, qui assurent que ce pré-entraînement sur des images réelles permet d'accélérer le processus de développement du modèle, puisqu'elles permettent de s'intéresser immédiatement à l'entraînement à partir de données spécifiques, et réduisent le volume nécessaire.

17. Ultralytics, *Ultralytics YOLOv8 Docs*, URL : <https://docs.ultralytics.com/> (visité le 15/08/2023).

18. J. Büttner, J. Martinetz, H. El-Hajj, *et al.*, « CorDeep and the Sacrobosco Dataset... ».

19. COCO - Common Objects in Context, URL : <https://cocodataset.org/#home> (visité le 15/08/2023).

20. PyTorch, URL : <https://pytorch.org/> (visité le 18/08/2023).

21. Darknet : Open Source Neural Networks in C, URL : <https://pjreddie.com/darknet/> (visité le 18/08/2023).

22. Aditya Sharma, *Training the YOLOv5 Object Detector on a Custom Dataset*, juin 2022, URL : <https://pyimagesearch.com/2022/06/20/training-the-yolov5-object-detector-on-a-custom-dataset/> (visité le 18/08/2023).

23. J. Büttner, J. Martinetz, H. El-Hajj, *et al.*, « CorDeep and the Sacrobosco Dataset... ».

docExtractor : un modèle pour les sources historiques

docExtractor est un modèle *off-the-shelf* de détection d'objet, dédié spécifiquement aux documents historiques²⁴. Développé par Tom Monnier, ce modèle a pour vocation d'être un outil efficace et prêt à l'usage pour le traitement de documents numérisés, capable de détecter le texte et segmenter les lignes, et d'extraire les illustrations en détectant leurs contours précis²⁵. docExtractor est entraîné à partir d'images produites par un générateur de documents historiques synthétiques (fig. 4.1), qui promet de bonnes performances même sans *fine-tuning* : l'utilisation de ce générateur automatique, appelé SynDoc²⁶, vise à répondre aux problématiques de création de jeux de données d'entraînement à partir de documents historiques, limités à la fois par le nombre de documents disponibles et les moyens humains que l'annotation d'images nécessite. SynDoc permet ainsi, dans la création d'un modèle, de combler l'absence d'un jeu de données de grande échelle d'images de documents historiques annotées²⁷.

Contrairement à YOLOv5, docExtractor est donc développé spécifiquement pour le traitement des sources historiques, en prenant en compte les spécificités de ces documents iconographiques : le modèle est ainsi adapté au traitement d'images de pages contenant du texte et des illustrations, et prévoit également des outils de traitement du texte, tels que la détection des lignes en prévision²⁸. docExtractor semble ainsi être le modèle à favoriser dans le cadre des projets mentionnés dans ce mémoire, cependant, les premières évaluations des modèles²⁹ sur les sources d'EIDA et de VHS semblent témoigner de performances équivalentes.

En effet, les détections de diagrammes lancées sur des sources du corpus EIDA en prévision de leur annotation par les chercheurs pour la constitution du jeu d'entraînement ont témoigné de performances prometteuses aussi bien pour docExtractor que pour YOLOv5 (fig. 4.2), avec des diagrammes non-détectés dans les deux cas : les performances

24. T. Monnier et M. Aubry, « docExtractor : An off-the-Shelf Historical Document Element Extraction », dans *2020 17th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 2020, p. 91-96, DOI : 10.1109/ICFHR2020.2020.00027, arXiv : 2012.08191 [cs].

25. Contrairement à YOLOv5 qui produit des annotations rectangulaires et ne gère pas les polygones.

26. Les images générées par SynDoc sont composées aléatoirement à partir d'une sélection d'images de fonds (pages et contextes), auxquelles sont ajoutées une mise en page selon laquelle est disposé un contenu image et texte, puis du bruit. Tous ces éléments sont tirés d'un jeu d'images pré-établi (constitué de 177 images de pages, 15 contextes, plus de 8000 œuvres d'art provenant de WikiArt, des lettrines générées à partir d'une lettre aléatoire avec 91 fonts possibles, et des dessins, schémas et textes tirés d'articles aléatoires sur Wikipedia, avec plus de 400 fonts) et composent des pages aléatoires mêlant images, texte et bruit, avec un nombre vaste de possibilités qui égale les plus grands jeu de données d'entraînement. Ces pages ne nécessitent pas d'annotations manuelles, puisque chaque élément de contenu est pré-annoté.

27. J. Büttner, J. Martinetz, H. El-Hajj, *et al.*, « CorDeep and the Sacrobosco Dataset... ».

28. La détection des lignes est une étape préliminaire de traitements tels que la reconnaissance optique de caractères (OCR) ou la reconnaissance de l'écriture manuscrite (HTR).

29. Les deux modèles ont fait l'objet d'un entraînement préliminaire sur des sources historiques qui ne sont cependant pas celles des deux projets, pour un premier affinement de leurs performances.

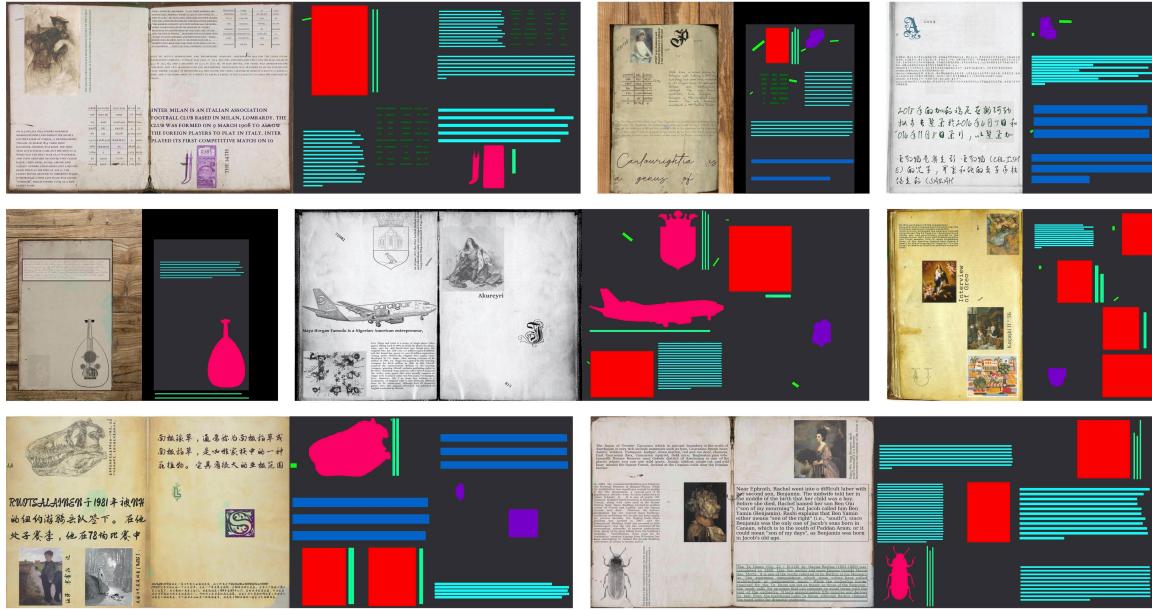


FIGURE 4.1 – Exemples tirés d'un jeu de données généré avec SynDoc

de chacun des deux modèles varient en fonction des pages, et il sera donc nécessaire de les départager en évaluant plus précisément leurs performances en amont de l'entraînement, puis après un premier entraînement des deux modèles sur les sources du projet³⁰. En tant que modèles *off-the-shelf* en libre accès, YOLOv5 et docExtractor ont pour vocation d'être efficaces et performants même sans entraînement sur des données spécifiques, et permettent ainsi de mener de manière satisfaisante des tâches de détection et d'extraction sur des données personnelles. Cependant, pour obtenir de meilleures performances, il est préférable d'entraîner ces modèles *off-the-shelf* sur des images tirées du corpus du projet, pour l'adapter et le spécialiser dans la résolution de ces problèmes particuliers.

4.2.2 Entrainement et *fine-tuning* d'un modèle de détection

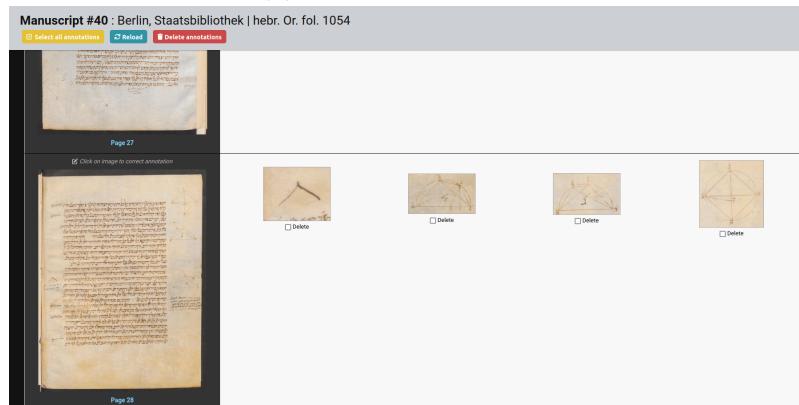
Démarche et volume des données

L'entraînement d'un modèle sur un jeu de données spécifiques permet de préciser les tâches effectuées et d'obtenir un modèle adapté aux besoins spécifiques de chaque projet. L'utilisation d'un modèle pré-entraîné permet, comme mentionné précédemment, de pallier aux limites de jeux de données avec trop peu d'exemples, en exploitant les représentations apprises précédemment et en les spécialisant sur les données spécifiques du projet qui l'utilise.

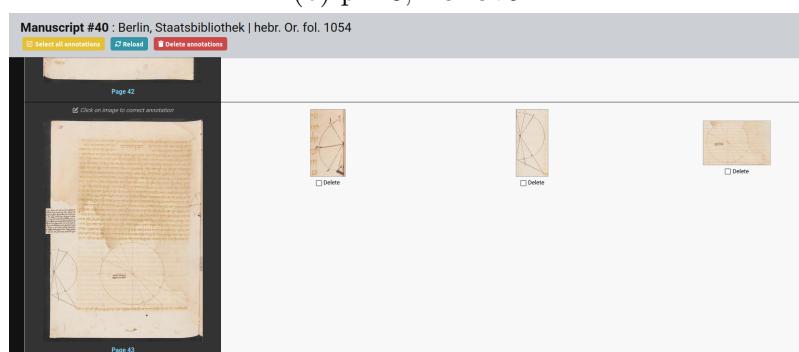
30. Sur le projet EIDA, les modèles ont été fournis par les chercheurs du laboratoire IMAGINE et l'ingénieur du projet VHS, n'ont donc pas encore été évalués sur les sources astronomiques du projet. La vérité de terrain étant encore en cours de production à l'été 2023, l'évaluation sera faite à partir d'un fragment de ce jeu de données lorsqu'il sera disponible.



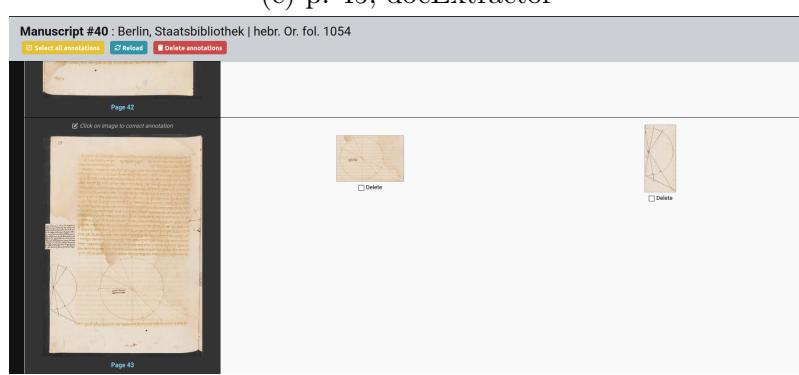
(a) p. 28, docExtractor



(b) p. 28, YOLOv5



(c) p. 45, docExtractor



(d) p. 45, docExtractor

FIGURE 4.2 – Diagrammes détectés par les modèles YOLOv5 et docExtractor dans deux pages du manuscrit hebr. Or. fol. 1054 de la Staatsbibliotek (Berlin).

Les modèles de détection *off-the-shelf* comme YOLOv5 prévoient souvent un *workflow* pour l'entraînement du modèle³¹ par des projets qui souhaiteraient le spécialiser. Ainsi, dans le cas de YOLOv5, l'entraînement étant prévu par les développements mis à disposition par Ultralytics, il est nécessaire pour les ingénieurs de prévoir des jeux de données d'entraînement adaptés au format requis par le modèle³² : l'annotation par les chercheurs doit donc être effectuée en prenant en compte ces restrictions, et les outils développés par les ingénieurs pour l'annotation des images doivent avoir pour sortie des fichiers aux formats appropriés. Un document de spécification sur les formats d'image et d'annotation a été réalisé dans le cadre du projet EIDA (Annexe A), pour garder traces des besoins techniques de l'entraînement, étape suivante du projet.

L'entraînement d'un modèle YOLOv5 se fait à partir d'un *dataset* d'images et d'étiquettes (*labels*) : il s'organise ainsi en un dossier de fichiers image (au format .png ou .jpg), et un dossier de fichiers texte (au format .txt) contenant une liste d'objets. Chaque fichier texte correspond à un ou plusieurs fichiers image – plusieurs dans le cas de numérisation d'ouvrages, notamment, qui se composent d'un fichier d'annotations par ouvrage numérisé et non par image – et listent les objets détectés dans l'image, à raison d'une ligne par objet. L'objet est caractérisé par ses dimensions (selon un rectangle qui l'encadre) et ses coordonnées sur l'image. Par exemple, le fragment de fichier d'annotation suivant correspond aux quatre objets détectés sur la page 6 du manuscrit 5 (Alm. 1) de la Gurukul Kangri Haridwar Collection (fig. 4.3) :

```

1   6 ms143_0006.jpg
2   243 1852 464 549
3   168 7 763 618
4   440 951 1174 1130
5   919 47 851 826

```

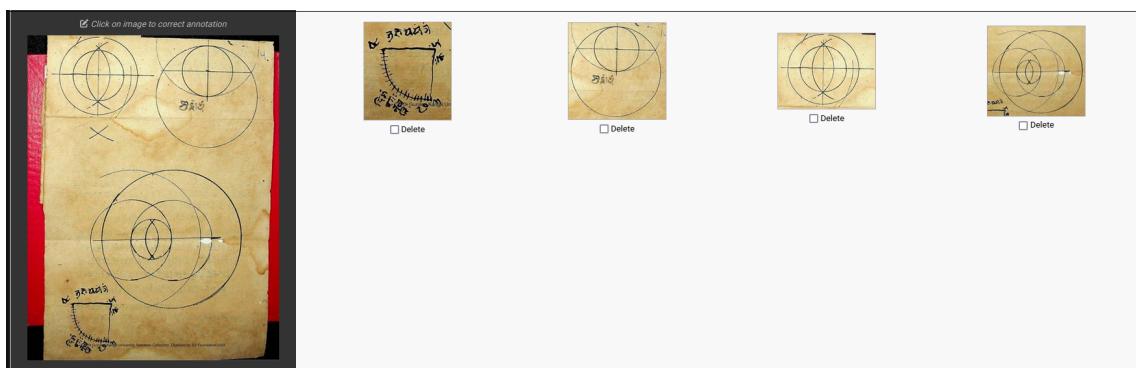


FIGURE 4.3 – Objets détectés sur la page 6 du manuscrit 5 (Alm. 1) de la Gurukul Kangri Haridwar Collection

31. A. Sharma, *Training the YOLOv5 Object Detector on a Custom Dataset...*

32. Les projets VHS et EIDA emploient tous deux YOLOv5 comme modèle, ou docExtractor intégré au *workflow* développé par Ultralytics pour YOLOv5 : la structure des jeux de données d'entraînement est donc la même pour les deux modèles.

À partir de cet ensemble d'images et d'annotations, l'entraînement peut être effectué : YOLOv5 prévoit un script pour l'entraînement d'un modèle, dont seuls les paramètres doivent être modifiés pour l'adapter aux besoins du projet. Ainsi, les modèles de détection *off-the-shelf* tels que YOLO ou docExtractor prévoient la possibilité d'un entraînement, simplifié par la mise à disposition d'outils permettant d'effectuer cette étape en écrivant peu de code. Cette volonté de rendre accessible des outils adaptables, construits comme un socle solide pour des objectifs variés, permet l'intégration de la vision artificielle à des projets divers, sans demander de ces derniers de créer ou d'entraîner de zéro des réseaux de neurones ou des modèles de détections. La détection étant, en effet, une tâche de base de la vision par ordinateur, il est peu pertinent d'allouer des ressources pour reproduire des techniques déjà appliquées par de nombreux projets, et les outils *off-the-shelf* offrent donc la liberté d'envisager la vision artificielle comme élément d'une chaîne de traitement des sources en allégeant les besoins et ressources – notamment humains et temporels – que demandent le développement et l'application de ces techniques.

Sur-ajustement et diversité des exemples

L'utilisation d'un modèle pré-entraîné permet d'obtenir un modèle de détection fonctionnel malgré une faible quantité de données disponible, puisque celui-ci a la capacité de généraliser les caractéristiques apprises à partir des données de pré-entraînement et de les appliquer aux données spécifiques du projet. Un trop faible volume de données présente en effet le risque de produire un modèle peu performant à cause du sur-ajustement (ou *overfitting*) : on parle de sur-ajustement lorsqu'un modèle est bien ajusté aux données d'entraînement, mais incapable de généraliser face à de nouvelles données³³.

Il est crucial de prévoir, lors de la constitution du jeu de données d'entraînement, une portion dédiée à la validation : on considère généralement que 20% du jeu de données total est un volume suffisant pour cette étape. Les données de validation sont des données – dans le cas de la détection, des images et annotations produites par des humains – qui n'ont jamais été vues par le modèle et n'ont donc pas été utilisées pour l'entraînement. Comme les données d'entraînement, ces données de validations doivent être représentatives et diversifiées, puisqu'elles permettent de vérifier les performances du modèles, et d'identifier un sur-ajustement si le modèle ne donne pas, après entraînement, des résultats satisfaisants sur cet ensemble. Les données de validation permettent de mettre en avant les pertes à mesure que le modèle s'ajuste aux données d'entraînement³⁴.

Pour prévenir le sur-ajustement d'un modèle³⁵, il est nécessaire de prévoir un jeu

33. F. Chollet, *L'apprentissage profond avec Python...*

34. Bert Carremans, *Handling Overfitting in Deep Learning Models*, janv. 2019, URL : <https://towardsdatascience.com/handling-overfitting-in-deep-learning-models-c760ee047c6e> (visité le 22/08/2023).

35. Il existe diverses méthodes pour corriger le sur-ajustement d'un modèle de détection après entraînement, cependant, ce mémoire traitant de l'intégration de la vision artificielle à une chaîne de traitement

de données d’entraînement au volume suffisant – supérieur à plusieurs milliers d’exemples dans le cas d’images³⁶ – et à la diversité représentative des données que le modèle pourra rencontrer. Le modèle de détection entraîné présente une capacité de généralisation qui rend son application pertinente pour le traitement des sources : l’entraînement d’un modèle *off-the-shelf* permet ainsi de partir d’un modèle relativement performant dans des contextes variés et d’obtenir un modèle répondant aux besoins du projet, en trouvant l’équilibre entre manque d’ajustement et sur-ajustement.

Les modèles de vision *off-the-shelf* ouvrent à des projets divers la possibilité d’intégrer la vision artificielle à leurs méthodologies, en réduisant le coût humain et temporel du développement d’un modèle de *deep learning* par la mise à disposition en accès libre d’outils déjà performants, qu’il est possible d’entraîner pour les ajuster à des données spécifiques. Pour la détection d’objets, des jeux de données en accès libre tels qu’ImageNet permettent un pré-entraînement de ces modèles *off-the-shelf*, qui apprennent alors des caractéristiques larges qui peuvent être précisées par un entraînement sur des sources plus spécifiques, en nécessitant un volume de données moins important qu’un modèle créé de zéro : ces modèles de détection *off-the-shelf* présentent ainsi une solution aux limites que peuvent présenter les sources historiques en termes de volume des données disponibles, et permettent également aux projets de se construire sur des bases solides, sans allouer de ressources à la création d’outils déjà existants, déjà performants, pour des tâches telles que la détection d’objet qui font partie des tâches canoniques de la vision par ordinateur.

des sources par le prisme du lien avec les équipes de recherche en histoire, nous n’aborderons pas ces questions qui concernent plus spécifiquement les aspects techniques de la création d’un modèle de détection. *Ibid.*

36. Le projet EIDA compte plusieurs dizaines de milliers de pages dans son jeu de données d’entraînement.

Chapitre 5

Construire une plateforme pour la détection : outils, interfaces et modèles de données

L'intégration du *deep learning* aux pratiques des chercheurs en sciences historiques passe par le développement d'une plateforme qui leur permet d'exploiter simplement ces outils pour traiter leurs sources. Cette plateforme à interface graphique doit être adaptée aux sources traitées, et pensée pour intégrer toutes les fonctionnalités souhaitées. En s'appuyant sur les développements réalisés par les projets EIDA et VHS, cette partie revient sur la construction d'une application et d'une API dédiées à la détection de diagrammes et images scientifiques dans les numérisations d'ouvrages manuscrits ou imprimés, avec une interface pour la correction de la détection : nous évoquons ainsi les méthodes de développement, les réflexions liées aux données, les besoins matériels liés à l'utilisation d'algorithmes de vision, et le développement d'une API pour lancer l'inférence sur un GPU (*Graphics Processing Unit*).

5.1 Penser une application pour la détection d'objets

5.1.1 Une interface pour le traitement des sources

Construire une application réutilisable

La création d'un modèle performant est une base essentielle à tout projet intégrant l'intelligence artificielle à sa chaîne de traitement des sources : il s'agit du socle sur lequel repose l'efficacité des outils développés pour les chercheurs. Cependant, d'un point de vue pratique, les historiens n'interagissent pas directement avec ce modèle, il est donc nécessaire de développer une application qui sert de pont entre les utilisateurs et l'algorithme de détection, et qui permet ainsi le dépôt et le traitement des sources par l'intermédiaire

d'une interface graphique, accessible et utilisable par les chercheurs¹. Cette plateforme doit ainsi être développée en prenant en compte la diversité des sources étudiées par ses utilisateurs.

La détection d'objets dans les images représente une première étape de nombreuses tâches de vision artificielle, et présente ainsi un intérêt pour de nombreux projets aux objectifs divers. Le développement d'une application pour la détection d'objets dans des images est donc d'intérêt pour plusieurs projets en cours, tels que VHS et EIDA : pour éviter à de multiples projets de développer des outils aux fonctionnalités similaires, ces deux projets ont fait le choix de construire une application réutilisable, dont le code sera proposé en accès libre sur GitHub². Le développement des applications EIDA et VHS est collaboratif, porté par les ingénieurs des deux projets qui travaillent en parallèle pour la mise en place d'une plateforme adaptée aux besoins de chacun des projets : l'enjeu est donc de créer une application suffisamment spécifique pour répondre aux besoins des deux projets, tout en étant suffisamment généraliste pour être réemployée à l'avenir par d'autres projets.

Il n'a, à ce jour, pas été développé d'interface pour les applications EIDA et VHS. Ces dernières sont accessibles pour les chercheurs qui demandent aux équipes d'ingénierie la création d'un identifiant donnant accès à l'interface administrateur des applications : celles-ci reposent sur l'interface par défaut proposée par Django (fig. 5.1). Un développement spécifique des interfaces destinées aux utilisateurs³ est prévu, afin de rendre ces applications plus ergonomiques. L'objectif actuel étant de mettre à disposition des chercheurs partenaires des applications pour le dépôt et le traitement de leurs sources, celles-ci ne sont pas accessibles au grand public, et ont pour objectif de permettre la constitution de jeux de données d'entraînement par le biais d'une interface graphique.

Développement en collaboration, programmation modulaire et remploi du code

Dans ce contexte, les ingénieurs des projets EIDA et VHS travaillent sur différentes branches d'un même dépôt GitHub, qui dispose alors d'une branche pour la mise en production de l'application EIDA et d'une autre pour la mise en production de l'application VHS. Ainsi, il est possible pour un projet d'exploiter les développements faits dans le cadre

1. Les applications EIDA et VHS sont développées en prenant en considération leur utilisation future par un public plus large que les chercheurs affiliés à ces projets. Elles restent cependant axées, dans leur développement, à un public provenant du monde de la recherche, nous utilisons donc le terme « chercheur » pour décrire ses utilisateurs. D'autres projets, tels que le projet Filigranes pour tous, développent leurs outils autour des utilisateurs grand public ; nous les prenons en compte dans ce mémoire, en ayant conscience de la différence de réflexion que cela apporte sur la manière de construire une plateforme.

2. Le développement des applications EIDA et VHS étant toujours en cours, le code n'est à cette date pas disponible en accès libre.

3. Le développement web frontal, ou *front-end*, désigne les productions d'une application avec lesquelles l'utilisateur peut interagir directement.

The screenshot shows the 'Administration de EIDA' interface. On the left, there's a sidebar with navigation links: 'Accueil / Vhsapp - Manuscripts', 'Écrivez ici pour filtrer...', 'AUTHENTIFICATION ET AUTORISATION', 'Groupes + Ajouter', 'Utilisateurs + Ajouter', 'VHSAPP', 'Imprimés + Ajouter', and 'Manuscrits + Ajouter'. The main area is titled 'Sélectionnez l'objet Manuscript à changer' and contains a search bar ('Rechercher') and a table with the following data:

Action	ID	SHELFMARK	CONSERVATION PLACE	AUTEURS ET/OU ÉDITEURS	IIF-MANIFEST	ANNOTATION
<input type="checkbox"/>	1	O II 10	El Escorial, Biblioteca del Monasterio de El Escorial	None		<button>EDIT ANNOTATIONS</button>
<input type="checkbox"/>	2	Chinois 4957	Paris, BnF	None		<button>EDIT ANNOTATIONS</button>
<input type="checkbox"/>	3	Lat. 7214	Paris, BnF	None		<button>EDIT ANNOTATIONS</button>
<input type="checkbox"/>	4	Lat. 16211	Paris, BnF	None		<button>EDIT ANNOTATIONS</button>
<input type="checkbox"/>	5	F II 7	Basel, Universitätsbibliothek	None		<button>EDIT ANNOTATIONS</button>
<input type="checkbox"/>	6	gt. 2925	Paris, Bibliothèque nationale de France	None		<button>EDIT ANNOTATIONS</button>
<input type="checkbox"/>	7					<button>EDIT ANNOTATIONS</button>
<input type="checkbox"/>	8					<button>EDIT ANNOTATIONS</button>

At the top right, there are buttons for 'AJOUTER MANUSCRIT' and 'EXPORTER IMAGES'. At the bottom right, there's a link 'BIENVENUE, ADMIN. VOIR LE SITE / MODIFIER LE MOT DE PASSE / DÉCONNEXION'.

FIGURE 5.1 – Interface administrateur de l'application EIDA

du second, ou de choisir de ne pas les employer s'ils ne semblent pas pertinents pour les besoins spécifiques du projet concerné. Ce développement collaboratif par le biais d'un dépôt GitHub assure une liaison constante entre les équipes des deux projets, tout en offrant la liberté de ne pas mettre en production tous les développements faits dans le cadre du projet jumeau.

La programmation modulaire est une solution favorisée pour la création d'une application réutilisable, puisqu'elle permet le développement de modules indépendants qui répondent à des besoins spécifiques, et qui peuvent être réemployés par d'autres projets sans être dépendants du reste de l'application. Ainsi, les différents traitements appliqués aux numérisations déposées par les utilisateurs font appel à des outils divers, indépendants, dédiés chacun à une tâche spécifique du *workflow* : cette disposition permet l'amélioration et la modification de chacun des modules sans impacter la structure globale de l'application, et permet également la récupération d'éléments spécifiques par des programmes futurs. Ainsi, le cœur de l'application EIDA/VHS permet le dépôt, le stockage et l'affichage des numérisations d'ouvrage, ainsi que la correction des annotations. La détection d'objet est gérée par une API, dont le développement est détaillé dans la partie suivante.

5.1.2 Décrire les sources : images et objets

Modèle de données pour les ouvrages historiques

L'application VHS est développée avec le *framework* Django et adossée sur une base de données gérée avec PostgreSQL. Le modèle de données doit permettre une description des sources historiques des projets, tout en étant adapté aux besoins spécifiques liés aux outils de détection utilisés. Comme l'application, le modèle de données construit pour le projet a pour vocation d'être suffisamment spécifique pour répondre aux besoins de description des sources historiques de VHS et EIDA, tout en étant suffisamment généraliste pour être réemployé à l'avenir par des projets différents qui souhaiteraient appliquer des

algorithmes de détection d'objet à des numérisations d'ouvrages.

Le modèle de données initialement construit (fig. B.1) pour l'application VHS prévoit l'existence de sources manuscrites et imprimées, qui correspond en effet aux supports représentés dans les corpus d'EIDA et de VHS, mais présente rapidement des limites, notamment liées à la description de séries d'ouvrages contenant une même œuvre, ou à la description d'un ouvrage contenant plusieurs œuvres. De plus, la pertinence de la distinction absolue des manuscrits et imprimés dans la description des objets est remise en question, menant à la refonte de ce modèle de données pour une version basée sur le témoin comme unité centrale (Annexe B).

Ce modèle, en cours d'implémentation dans l'application EIDA, présente l'avantage d'une plus grande flexibilité, pertinente aussi bien pour les projets qui le construisent que pour des projets futurs qui souhaiteraient utiliser le code en accès libre de l'application développée. En l'absence d'objets spécifiques *Manuscript* et *Printed*, le modèle de données permet la description d'ouvrages aux configurations diverses, sans nécessiter dans l'architecture de l'application des chaînes de traitement indépendantes pour les manuscrits et les imprimés. Ce modèle est ainsi appuyé sur une entité générique représentant l'objet physique, le *Witness*, tandis que le précédent modèle distinguait de manière plus nette les entités *Volume*, *Manuscript* et *Printed*, résultant en un modèle de données moins centralisé autour de l'objet physique.

L'entité *Witness* désignant un objet physique permet l'introduction d'une granularité plus fine à travers les entités *Content* et *Work*, qui permettent la description de manuscrits composites contenant plusieurs œuvres : ce niveau de description est pensé pour faciliter, après l'implémentation d'algorithmes de détection de similarité, la comparaison des illustrations ou diagrammes dans différentes versions d'une même œuvre.

L'entité numérisation (*Digitization*) est centrale à ce modèle de données, en tant qu'élément de base d'une chaîne de traitement pour la détection d'objet : dans le modèle de données de la plateforme VHS, la numérisation est rattachée au *Volume* et non à l'entité *Printed*, créant une distinction entre le traitement des sources manuscrites et imprimées. Le modèle de données modifié attache la *Digitization* au *Witness* pour dépasser cette différence de traitement, en considérant la numérisation d'un objet physique unique qu'est le témoin. Cette variation de l'entité *Digitization* offre également la possibilité cruciale d'attacher plusieurs numérisations à un même *Witness*, en liant l'identifiant du témoin à la numérisation et non la numérisation au manuscrit ou au volume, comme fait précédemment. La possibilité de lier plusieurs numérisations à un témoin permet, du point de vue de la détection d'objets, la comparaison des résultats de détection sur diverses versions numérisées, et de conserver les meilleurs résultats.

Ainsi, le modèle de données est construit en prenant en compte, d'une part, les besoins des historiens pour la description des sources historiques, en mettant l'accent sur la résolution des différences matérielles entre manuscrit et imprimé pour obtenir un modèle

global, et en projetant déjà, d'autre part, les traitements appliqués sur les numérisations des ouvrages avec des algorithmes de vision artificielle, de la détection d'objet à la détection de similarité. Le modèle résultant de ces réflexions, menées en ateliers réunissant les équipes d'histoire des sciences et les équipes de vision par ordinateur, correspond ainsi aux besoins parfois divergents des différents acteurs des projets.

Images et objets détectés : une nouvelle source ?

Il est une entité que le modèle de données des projets VHS et EIDA ne prend à ce jour pas en compte : les objets détectés. La chaîne de traitement automatique n'étant pas déployée, les objets détectés ne comptent à ce jour pas parmi les éléments décrits ; il reste cependant nécessaire de considérer leur intégration au modèle de données dans le cadre d'une réflexion sur les étapes futures du développement, et sur les traitements qu'il est souhaitable d'appliquer aux sources historiques. L'algorithme de détection d'objets utilisés sur les numérisations de témoins génère en sortie une série d'images correspondant aux objets – diagrammes ou illustrations scientifiques – détectés dans les images des pages des ouvrages. Ces nouveaux objets image, stockés dans l'application, constituent une nouvelle entité non-exploitée à ce jour, mais qui présentera, dans le cadre des futures applications d'algorithmes de vectorisation ou de détection de similarité, un intérêt particulier. Il est donc nécessaire d'envisager une extension du modèle de données pour l'intégration de ces objets, qui doivent être liés au témoin auquel ils appartiennent, en vue, par exemple, du développement de fonctionnalités d'édition ou pour permettre la comparaison des illustrations dans différentes versions d'une œuvre.

Ces objets, exclus à ce jour du modèle de données décrit, représentent en effet l'élément d'intérêt des études qui pourront être faites des sources traitées. Les fonctionnalités envisagées, qu'il s'agisse de vectorisation ou de détection de similarité dans les illustrations, ne s'appuient pas sur les numérisations complètes des témoins, mais exclusivement sur ces images extraites. Il est ainsi nécessaire que ces objets soient intégrés au modèle de données en tant qu'entité : leur exploitation requiert l'existence lien avec le témoin – ainsi qu'avec la numérisation – duquel elles sont extraites, pour permettre leur identification. La détection constitue la première étape d'une chaîne de traitement plus longue⁴, et le modèle de données actuellement instauré peut être étendu dans le cadre de développements futurs ayant pour vocation la mise en place de ces fonctionnalités.

4. Les applications développées par EIDA et VHS étant à ce jour focalisées sur la création de données d'entraînement – basées sur le modèle des jeux de données YOLOv5 qui nécessite des images et les annotations au format texte – et la détection d'objets, ces sorties ne sont pas actuellement exploitées. La chaîne de traitement faisant appel à IIIF pour l'affichage des images et de leurs annotations, aucun traitement n'est appliqué aux fichiers image des objets détectés, et le fichier texte suffit à l'indexation des annotations pour leur affichage dans visualiseur IIIF. Nous décrivons l'architecture et la chaîne de traitement des sources dans les sections suivantes.

5.2 Annoter sur un GPU : extractorAPI

extractorAPI⁵ est une API développée en tant que module de l’application EIDA : elle a pour vocation de séparer l’algorithme de détection d’objets du reste de l’application. extractorAPI est conçue autour d’une unique fonctionnalité : exécuter un algorithme de détection d’objet sur des images reçue de l’application EIDA, et retourner les résultats de cette détection.

5.2.1 Architecture du système

CPU ou GPU ?

L’apprentissage profond et la vision artificielle représentent le plus souvent des tâches qui requièrent une puissance de calcul importante. L’utilisation d’un GPU plutôt que d’un CPU (*Central Processing Unit*) est donc de bonne pratique, si ce n’est une nécessité, pour assurer la complétion des tâches en un temps correct, particulièrement lorsqu’il s’agit de tâches qui doivent retourner rapidement un résultat à un utilisateur. Un GPU est un processeur conçu pour avoir de meilleures performances dans le traitement d’images, et dans l’exécution parallèle de tâches⁶, tandis qu’un CPU est un processeur conçu pour effectuer des calculs plus basiques, pour répondre aux besoins quotidiens. Le nombre important de coeurs que contient un GPU permet une gestion simultanée de nombreuses tâches, et permet donc une plus grande efficacité sur des traitements basés sur le parallélisme, comme le sont les réseaux de neurones, éléments centraux du *deep learning*.

D’un point de vue applicatif, l’utilisation d’un GPU pour les tâches de vision artificielle permet d’assurer la fluidité du fonctionnement de l’application qui intègre ces algorithmes, et de retourner, comme mentionné précédemment, un résultat rapide à l’utilisateur qui souhaiterait soumettre ses sources à un traitement via l’application. Outre le *machine learning*, un GPU a les capacités de traiter les images de manière plus performante. Les traitements par des algorithmes de vision artificielle nécessitent en effet un enregistrement des images dans l’application pour que s’effectuent les tâches, ainsi, d’un point de vue matériel, il est essentiel de prévoir, d’une part, la mémoire nécessaire au stockage – même s’il est temporaire – des images, ainsi que la puissance de calcul pour leur enregistrement puis leur traitement dans un délai suffisamment court pour retourner une réponse à l’utilisateur en un temps satisfaisant.

5. Jade Norindr, *Element Extraction on a GPU*, juin 2023, URL : <https://github.com/jnorindr/extractorAPI> (visité le 26/08/2023).

6. German Sharabok, *Why Deep Learning Uses GPUs?*, juill. 2020, URL : <https://towardsdatascience.com/why-deep-learning-uses-gpus-c61b399e93a0> (visité le 26/08/2023).

De l'utilisateur au modèle de détection

Les applications VHS et EIDA pour la détection d'objets dans les numérisations reposent sur une même architecture globale, construite pour être adaptable par des projets qui souhaiteraient traiter des sources historiques par la vision artificielle : dans une optique de programmation modulaire, et pour permettre l'utilisation d'un GPU pour le traitement des images, la solution proposée par ces projets est l'exclusion du modèle de détection de la structure de l'application en elle-même, et le développement d'une API sur un GPU pour cette tâche spécifique.

Ainsi, l'application est développée, avec ses dépendances, sur un serveur spécifique, qui accueille donc le code de l'application et la base de donnée PostgreSQL à laquelle elle est adossée, et stocke les images versées par les utilisateurs qui servent à la génération de manifestes IIIF propres à EIDA. Par le biais de l'interface web, l'utilisateur envoie des requêtes à cette application, et l'application retourne des réponses à l'utilisateur. Le modèle de détection existe sur un serveur indépendant du serveur de l'application, sur un GPU. Pour connecter le modèle à l'application, une API est développée : celle-ci reçoit de la part de l'application EIDA des requêtes contenant les informations soumises par l'utilisateur, et retourne à l'application une réponse, qui est ensuite retournée à l'utilisateur par son biais. Ainsi, le lien de l'utilisateur au modèle de détection se fait par le biais de l'application, elle-même liée au modèle par une interface qu'est l'API (fig. 5.2) : l'API n'a pas d'interface destinée à être vue par l'utilisateur⁷.

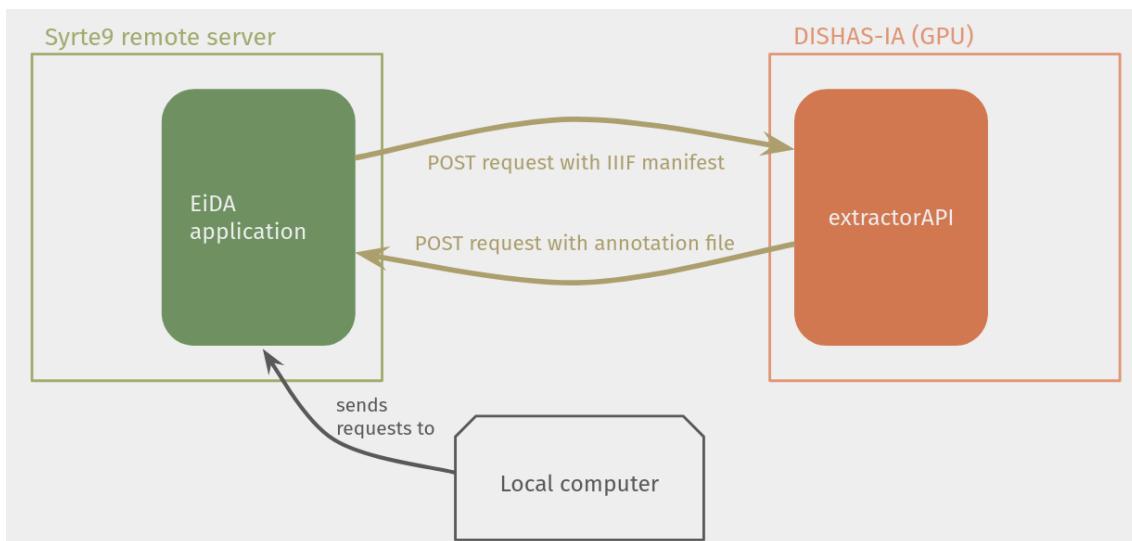


FIGURE 5.2 – Schéma de l'architecture de l'application EIDA et de l'API extractorAPI

7. Contrairement à l'application, composée d'un *back end* et d'un *front end*, l'API n'a qu'un *back end*, c'est-à-dire le code inaccessible à l'utilisateur permettant son fonctionnement.

5.2.2 Cahier des charges et analyse des besoins

Programmation modulaire et *open source*

Avant le développement de l’API, il est nécessaire d’établir les besoins auxquels cette dernière doit répondre par la rédaction d’un cahier des charges (Annexe C), qui détaille les attentes techniques liées à l’outil créé. La création d’outils *open source* réutilisables par d’autres projets étant un élément central du projet EIDA, il est attendu de l’API qu’elle réponde aux besoins spécifiques du projet tout en présentant la possibilité d’être réemployées dans d’autres contextes. Le lancement d’algorithmes de détection d’objets sur un GPU est une tâche suffisamment généraliste pour être appliquée par d’autres projets de recherche utilisant la vision par ordinateur pour le traitement des sources historiques, il est donc désirable de créer une API réemployable pour éviter la multiplication d’outils similaires développés par des projets aux objectifs proches.

Dans cette optique de libre accès, le code de l’API est mis à disposition sur GitHub⁸, avec la possibilité de le réemployer, en l’adaptant à l’architecture spécifique d’un projet et en modifiant simplement le modèle utilisé pour la détection. La création d’une API indépendante de l’application EIDA renforce la possibilité de remployer des outils développés, puisqu’en tant qu’entité séparée de l’application du projet, l’API effectue des tâches peu spécifiques, et peut ainsi être intégrée à une application différente sans grandes modifications du code.

API REST

La communication en SSH (*Secure Shell*) entre le GPU et l’application EIDA n’étant pas satisfaisante du point de vue de la sécurité, le développement d’une API permet de gérer ces échanges par une voie sécurisée. Il est défini dans le cahier des charges que l’API communique avec l’application par le biais de requêtes HTTP qu’elle reçoit et envoie : en tant qu’interface pour la communication de l’application et du modèle de détection, l’API reçoit des requêtes HTTP et retourne une ressource spécifique dépendant de l’URL (*Uniform Resource Locator*) à laquelle la requête est envoyée.

Par définition, l’API développée est donc une API REST (*Representational state transfer*). REST définit un ensemble de contraintes architecturales pour le développement de services web, qui permettent la communication et l’échange de données sur Internet. Les API REST sont basées sur des ressources localisées par des URL⁹ : lorsqu’un client demande une ressource, il envoie une requête HTTP¹⁰ à un *endpoint* de l’API, qui retourne en ré-

8. J. Norindr, *Element Extraction on a GPU...*

9. Une API REST, c’est quoi ?, URL : <https://www.redhat.com/fr/topics/api/what-is-a-rest-api> (visité le 27/08/2023).

10. Les actions à effectuer sur la ressource sont définies par la requête HTTP : dans ce mémoire, nous mentionnons spécifiquement les requêtes GET, qui permettent au client de récupérer des données, et les requêtes POST, qui permettent d’envoyer des données avec la requête.

ponse à cette requête une ressource dans un format standard¹¹. Dans une API REST, les communications entre client et serveur sont *stateless*, signifiant que les informations du client ne sont pas stockées entre les requêtes, et que les requêtes sont traitées indépendamment¹². Les API REST reposent également sur une interface uniforme pour les interactions avec le client, et un système de couches qui assurent la flexibilité de l'API et la sécurité des serveurs.

Ainsi, une API REST permet la création de services web interopérables : elle facilite la communication entre différents systèmes informatiques en utilisant des requêtes HTTP pour accéder et manipuler des ressources. Ce mode d'échange des données étant standardisé, il est envisageable pour des projets futurs d'intégrer l'API à leurs développements, à la seule condition que soit développée une application envoyant des requêtes en accord avec le protocole HTTP, et apte à recevoir des requêtes de la part de l'API. L'architecture REST étant pensée pour l'uniformisation et l'intéropérabilité, un grand nombre d'applications sont construites autour de ces principes, et de nombreux *frameworks* les prennent en compte dans les outils qu'ils proposent.

Sécurité et robustesse

Le cahier des charges rédigé en amont du développement de l'API définit les contraintes du point de vue de la sécurité, des fonctionnalités souhaitées et des échanges de données. extractorAPI répond ainsi aux besoins spécifiques du projet et de l'application EIDA, tout en faisant le choix d'outils libres, standardisés, qui lui assurent d'être réutilisable dans des contextes et architectures variés.

L'API étant déployée sur un GPU mis à disposition de l'Observatoire de Paris. Celle-ci se doit d'être sécurisée, pour interdire les requêtes provenant de sources non-identifiées, et éviter ainsi un risque de surcharge de l'API par des requêtes externes. Une authentification avec un token basée sur les principes REST¹³ a été envisagée, ainsi qu'une restriction des hôtes autorisés à envoyer des requêtes aux *endpoints* de l'API.

La question de la robustesse est également soulevée parmi les besoins fondamentaux en amont du développement : l'API est une interface pour des algorithmes appliquant des traitements à des fichiers volumineux – des images – en grand nombre, et doit donc avoir la robustesse nécessaire pour accomplir ces tâches. En effet, l'API reçoit des numérisations d'ouvrages sous la forme de manifestes IIIF, qui représentent ainsi plusieurs centaines d'images à enregistrer avant d'en extraire les objets. Les tâches de lecture du manifeste, d'extraction et d'enregistrement des images sont particulièrement chronophages, tout comme la détection d'objet, il est donc nécessaire de prévoir un gestionnaire de

11. Le format JSON, par exemple, est un format lisible et interprétable par les machines comme par les humains : il est le plus communément utilisé. *Ibid.*

12. *Ibid.*

13. Miguel Grinberg, *RESTful Authentication with Flask*, URL : <http://blog.miguelgrinberg.com/post/restful-authentication-with-flask> (visité le 27/08/2023).

tâches, qui crée une file d'attente des requêtes reçues, afin que ces requêtes successives se lancent dans leur ordre de réception par l'API, et que le traitement d'une nouvelle source ne se lance qu'après complétion du traitement de la source précédente. Le gestionnaire des tâches permet ainsi d'éviter la surcharge de l'API en cas de requêtes simultanées, et assure donc son bon fonctionnement, un temps de traitement correct et une gestion cohérente des requêtes.

Le mode de communication, la sécurité et la robustesse sont trois points fondamentaux du développement applicatif, et représentent donc les attentes principales détaillées dans le cahier des charges pour le développement de l'API. L'objectif d'extractorAPI est, en effet, de proposer une structure flexible et légère pour l'inférence d'un modèle de vision artificielle sur un GPU, afin d'extraire la détection d'objet de l'application du projet, pour en faire un module externe aisément modifiable, réutilisable, et adaptable à des modèles de détection, architectures et besoins variés.

5.2.3 Développement de l'API

Choix du *framework* : Flask, Django ou FastAPI ?

Il existe divers *frameworks*¹⁴ Python permettant le développement d'une API. Ces *frameworks* présentent tous fonctionnalités spécifiques, la première étape du travail de développement d'extractorAPI est donc la sélection du *framework* qui servira de squelette à l'application. Trois options ont été envisagées : Django¹⁵, Flask¹⁶ et FastAPI¹⁷.

Django est un *framework* développé en 2003 dédié au développement applicatif en Python, gratuit et en libre accès. L'application EIDA, avec laquelle communique extractorAPI, est basée sur ce *framework*, qui constitue ainsi une option logique dans la continuité du développement de l'application du projet. Les ressources en ligne pour l'intégration de PyTorch, module de *machine learning*, à une application Django sont nombreuses¹⁸, et il existe un *framework* REST¹⁹ qui simplifie le développement d'une API à partir de Django. Apprécié pour sa robustesse, Django est cependant un *framework* assez lourd, constitué de nombreux modules intégrés qui tendent le ralentir²⁰. Django permet la création d'API

14. Un *framework*, littéralement « cadre de travail » est une bibliothèque de composants logiciels ayant pour vocation à servir de base, ou de squelette, pour le développement d'une application dans un langage donné. Les *frameworks* ont pour vocation de simplifier le travail des développeurs, tout en uniformisant les applications produites.

15. *Django*, URL : <https://www.djangoproject.com/> (visité le 27/08/2023).

16. *Flask Documentation (2.3.x)*, URL : <https://flask.palletsprojects.com/en/2.3.x/> (visité le 27/08/2023).

17. *FastAPI*, URL : <https://fastapi.tiangolo.com/> (visité le 27/08/2023).

18. Görkem Arslan, *Create a Machine Learning API With Django Rest Framework / Better Programming*, juill. 2021, URL : <https://betterprogramming.pub/create-a-machine-learning-api-with-django-rest-framework-967571640c46> (visité le 27/08/2023).

19. *Django REST Framework*, URL : <https://www.django-rest-framework.org/> (visité le 27/08/2023).

20. James Sandy, *Choosing between Django, Flask, and FastAPI*, URL : <https://www.section.io/>

puissantes, avec des modules nombreux ; cependant, pour le développement d'une API dont la priorité est la rapidité et la flexibilité, il existe des *frameworks* plus légers, tels que Flask.

Flask est un *framework* pour le développement web en Python, basé sur un noyau simple dont les fonctionnalités peuvent être étendues. Minimaliste dans ses modules de base, les principes fondateurs de Flask sont la légèreté et la flexibilité, qui en font un *framework* aisément adaptable à de nombreux projets, simple à prendre en main, et aux possibilités variées. extractorAPI ayant pour vocation d'être un module léger complémentaire à l'application EIDA, il a été choisi d'utiliser Flask pour son développement, afin de produire une API simple et flexible, qu'il serait aisément d'adapter à d'autres projets, ou aux étapes futures d'EIDA, qui impliquent des modèles de vision artificielle aux objectifs différents des modèles de détection utilisés à ce stade.

Le *framework* FastAPI a également été envisagé pour le développement d'extractorAPI. Ce dernier ayant été publié en 2018, il intègre des fonctionnalités facilitant la rédaction de documentation²¹ pour l'API développée, et est réputé pour sa rapidité²². Sa flexibilité rendait ce *framework* adapté au projet EIDA et aux attentes entourant extractorAPI, cependant, s'agissant d'un outil récent, la documentation comme la communauté ne sont pas aussi riches que pour les *frameworks* précédemment mentionnés, il a donc été décidé de favoriser Flask pour ce projet, pour permettre un développement fluide et une mise à disposition rapide de l'API.

Intégration du modèle de détection

L'application EIDA et extractorAPI ayant, en premier lieu, pour vocation d'être des outils à la constitution de jeux de données d'entraînement et de vérité de terrain, il n'existe pas à ce jour de modèle de détection d'objet définitif à intégrer à l'API. Dans son développement, il est donc nécessaire de prévoir la possibilité de choisir le modèle de détection utilisé, afin de substituer les modèles temporaires utilisés actuellement²³ par un modèle entraîné lorsque l'étape d'entraînement sera complétée.

Le script d'inférence utilisé est celui de YOLOv5²⁴, reproduit et importé comme un module dans le code de l'API, et appelé par une fonction lorsque le client envoie une requête à l'*endpoint* servant à la détection d'objet. Le code de YOLOv5 étant en accès

[engineering-education/choosing-between-django-flask-and-fastapi/](https://engineering-education.org/choosing-between-django-flask-and-fastapi/) (visité le 27/08/2023).

21. *FastAPI, le framework python alternatif à Django et Flask ?*, juin 2023, URL : <https://www.agencedebord.com/blog/fastapi-le-framework-python-alternatif-a-django-et-flask> (visité le 27/08/2023).

22. Id., *Choosing between Django, Flask, and FastAPI...*

23. Les modèles utilisés actuellement ne sont pas entraînés sur les données du projet, mais permettent de fournir aux chercheurs qui créent les données d'entraînement une première annotation imparfaite, qu'ils corrigent depuis l'interface d'annotation de l'application, accélérant le travail de création de cette vérité de terrain.

24. Il s'agit du script `detect.py` disponible en accès libre dans le dépôt GitHub du projet. <https://github.com/ultralytics/yolov5/blob/master/detect.py>

libre, le script pour le lancement de la détection est réutilisé sans modification (Annexe D), et comporte donc les options liées aux fichiers vidéo que YOLO propose initialement : la restriction des formats envoyés est gérée par l'application, qui ne supporte que les manifestes IIIF, les fichiers PDF et les fichiers image, qui sont transmis à l'API sous forme de manifestes IIIF au format JSON. Par défaut, le script fait appel à un modèle de détection YOLOv5 générique, défini par le paramètre `weights` de la fonction `run_vhs` : l'appel à un modèle personnalisé est défini dans les routes de API²⁵.

La route correspondant à l'*endpoint* `/run_detect` est utilisée pour soumettre une numérisation à traiter par le modèle : recevant de la part de l'application des requêtes HTTP POST contenant des données, celle-ci prévoit la réception de l'un d'un manifeste IIIF contenant les images à annoter, ainsi que le nom du fichier contenant le modèle à utiliser pour la détection. Elle permet ainsi, lors du paramétrage des échanges entre l'application et l'API, de déterminer le modèle utilisé pour la détection d'objets.

```

1 @app.route("/run_detect", methods=['POST'])
2 def run_detect():
3     # Get manifest URL from the request form
4     manifest_url = request.form['manifest_url']
5     model = request.form.get('model')
6
7     # function.delay() is used to trigger function as celery task
8     detect.delay(manifest_url, model)
9     return f"Run detect task triggered with Celery! Check terminal to see
the logs..."
```

La fonction `run_detect()` fait appel à une fonction `detect()` lancée en tâche de fond, dans laquelle la ligne de code suivante permet d'établir l'utilisation d'un modèle par défaut si aucun modèle spécifique n'est défini dans le formulaire envoyé à l'*endpoint* par l'application :

```
1     model = DEFAULT_MODEL if model is None else model
```

Le modèle par défaut est défini dans le module d'utilitaires de l'API²⁶, et permet d'éviter une erreur si le modèle de détection n'est pas défini dans la requête envoyée par le client. La fonction `detect` fait ensuite appel à la fonction `run_vhs`²⁷, avec pour paramètres les informations envoyées dans le formulaire de la requête POST à l'*endpoint* de détection, ainsi que les chemins des images sur enregistrées sur lesquelles performer la détection d'objets, et le chemin du fichier dans lequel doivent être écrites les annotations.

25. Les routes sont définies dans le fichier `app.py` : l'API ne contenant que deux routes, une architecture simple a été choisie, et les routes ne sont donc pas dissociées du corps du code. <https://github.com/jnorindr/extractorAPI/blob/main/app.py>

26. Parmi les utilitaires, le fichier `path.py` permet de définir les chemins dans l'application, et de les appeler en minimisant le risque d'erreurs. <https://github.com/jnorindr/extractorAPI/blob/main/utils/path.py>

27. Cette fonction correspond au script d'inférence de YOLOv5.

```

1 run_vhs(
2     weights=f"{MODEL_PATH}/{model}",
3     source=wit_path / img,
4     anno_file=anno_file,
5     img_nb=i
6 )

```

Ainsi, plusieurs modèles de vision artificielle peuvent être importés sur le GPU, dans le dossier de l'API, au format .pt²⁸. Ces modèles peuvent ensuite être appelés par le script dédié au lancement de l'inférence, avec des paramètres établis dans la requête envoyée par l'application EIDA à l'*endpoint* d'extractorAPI qui reçoit des numérisations pour leur traitement par l'algorithme de détection.

Formats des fichiers et échanges de données avec l'application

extractorAPI échange des données avec EIDA dans des formats spécifiques, définis dès le cahier des charges, pour correspondre notamment aux exigences de formats pour l'entraînement d'un modèle.

L'API reçoit de la part de l'application une requête POST contenant un manifeste IIIF, au format JSON, qui permet d'échanger rapidement et de manière efficace des numérisations comportant un grand nombre d'images. Bien que les images de ces numérisations soient enregistrées dans l'application EIDA après leur soumission par l'utilisateur, il n'est en effet pas envisageable d'envoyer par requête HTTP à l'API un tel volume de fichiers image qui, en plus de représenter plus d'une centaine de fichiers par numérisation, sont des fichiers lourds. L'envoi de manifestes IIIF implique, après réception de la requête par l'API, l'enregistrement des images sur le GPU, qui constitue un traitement chronophage mais nécessaire pour la détection d'objets qui ne peut s'effectuer que sur des fichiers image enregistrés²⁹. Pour plus d'efficacité, les images ne sont pas supprimées après l'étape de détection d'objet : celles-ci sont conservées pour permettre le lancement rapide de la détection en cas de nouvelle requête envoyée par le client pour une même numérisation. Cependant, le corpus traité par un projet tel qu'EIDA représentant des plusieurs centaines – ou plusieurs milliers – de numérisations, il est nécessaire de prendre en compte la question de l'espace de stockage sur le GPU utilisé pour la détection ; ainsi, une fonction avec

28. Le format .pt est un format standard pour les modèles de *machine learning* créés avec PyTorch. Jim Dowling, *Guide to File Formats for Machine Learning : Columnar, Training, Inferencing, and the Feature Store*, oct. 2019, URL : <https://towardsdatascience.com/guide-to-file-formats-for-machine-learning-columnar-training-inferencing-and-the-feature-store-2e0c3d18d4f9> (visité le 29/08/2023)

29. Il est peu satisfaisant, en termes de performances et de gestion de la mémoire, d'enregistrer les images une première fois sur le serveur de l'application EIDA, puis de les enregistrer une seconde fois sur le GPU. Cependant, à ce jour, aucune chaîne de traitement ne nécessitant qu'un seul enregistrement des images ne s'est avérée satisfaisante – des réflexions sont actuellement en cours pour modifier cette manière de procéder pour la rendre plus efficace.

une tâche périodique, planifiée avec `crontab`³⁰, supprime les images enregistrées sur le GPU après une semaine.

Après finalisation de la détection d'objets dans les images du document numérisé, extractorAPI retourne à l'application un fichier texte `.txt` contenant toutes les annotations, c'est-à-dire une liste des noms des fichiers image suivis des coordonnées des objets détectés. Un unique fichier d'annotation est généré pour chaque numérisation, et celui-ci contient les données pour chaque image de la numérisation. Le fichier d'annotation est envoyé à l'application EIDA par une requête POST provenant de l'API.

Gestion des tâches et files d'attente

extractorAPI a été conçue pour lancer sur le GPU l'inférence d'un modèle de détection d'objets dans les images : l'API est donc construite pour effectuer des tâches lourdes, requérant une grande puissance de calcul, qui ne peuvent être menées simultanément. extractorAPI a pour tâches le téléchargement de plusieurs centaines d'images extraites de manifestes IIIF, puis la détection d'objets dans ces images enregistrées : pour éviter sa surcharge, un gestionnaire de tâches robuste, permettant de paramétriser le nombre de requêtes traitées en parallèle, est nécessaire.

Pour gérer la réception et le traitement des requêtes, qui peuvent arriver simultanément, alors qu'une tâche est déjà en cours, Celery³¹ a été implémenté dans l'API, en tant que gestionnaire de file d'attente. Après création d'une instance Celery³², les fonctions appelées en tâche de fond sont définies à l'aide d'un décorateur, puis sont appelées dans les fonctions des routes à l'aide de la méthode `.delay()`.

```
1 @celery.task
2 def detect(manifest_url, model):
3
4     detect.delay(manifest_url, model)
```

Ainsi, la lecture du manifeste IIIF, l'extraction des images et la détection d'objets sont définis en tant que tâches de fond gérées par Celery : il est ensuite possible, au lancement de l'application et de Celery, de définir un nombre de *workers*, c'est-à-dire le nombre de tâches pouvant être menée simultanément. Si le nombre de requêtes reçues à un moment donné dépasse le nombre de *workers* disponibles, une file d'attente se crée, et les requêtes en attente sont traitées après complétion des requêtes en cours. Celery permet

30. `cron` est un programme permettant d'exécuter des tâches à une heure spécifique, selon un cycle spécifié à l'avance. « cron », Wikipédia (, juill. 2023), URL : <https://fr.wikipedia.org/w/index.php?title=Cron&oldid=206272232> (visité le 29/08/2023)

31. *Celery - Distributed Task Queue — Celery 5.3.1 Documentation*, URL : <https://docs.celeryq.dev/en/stable/> (visité le 29/08/2023).

32. La création de l'instance Celery fait appel à une classe `Celery` dont on définit en argument les modules ainsi que le *back end* et le *broker*, qui gère les messages. Cette instanciation suffit à l'implémentation de Celery dans une application : Celery est conçu pour permettre une prise en main facile, offrant une solution simple à des problèmes de gestion des files d'attente qui peuvent s'avérer complexes. https://github.com/jnorindr/extractorAPI/blob/main/utils/celery_utils.py

ainsi d'éviter la surcharge de l'API en cas de réception d'un grand nombre de requêtes, et d'assurer la gestion fluide des demandes de détection envoyées par les utilisateurs. Le développement d'une API pour le lancement de l'inférence d'un modèle de *deep learning* sur un GPU doit ainsi se faire en prenant en considération les besoins techniques de la vision artificielle en termes d'espace de stockage, de robustesse et de puissance de calcul. Il s'agit en effet de traitements réalisés sur des images qu'il est nécessaire d'enregistrer, de stocker, de traiter, et de tâches de détection chronophages, qui répondent à des requêtes envoyées par le client – l'application – et doivent retourner une réponse en un temps satisfaisant. Ainsi, le développement d'extractorAPI se focalise en premier lieu sur la gestion des tâches et l'établissement d'une file d'attente pour éviter sa surcharge en cas de réception de nombreuses requêtes. Si une API ne présente pas d'interface accessible aux utilisateurs, sa construction doit malgré tout se faire en prenant en compte leur expérience, ainsi, il est souhaitable de traiter les requêtes dans leur ordre d'arrivée, de manière cohérente, pour assurer une réponse sans temps d'attente trop long. L'anticipation de la réception de nombreuses requêtes, des limites des capacités de stockage du GPU ou des changements de modèle de détection permet d'assurer le développement d'un outil solide et flexible, réutilisable dans des contextes variés et adaptable à de nombreux projets aux ambitions diverses.

L'*open source* est une préoccupation centrale lors du développement d'outil pour les projets de recherche : il est en effet souhaitable de produire, dans la mesure du possible, des applications réutilisables par des projets futurs, pour assurer une continuité dans les travaux produits sans que se démultiplient les développements d'outils similaires. Ainsi, le remploi du code est au cœur des ambitions des équipes d'ingénierie : l'intégration de l'apprentissage profond à des projets d'histoire nécessite des outils spécifiques, construits autour de ces traitements des sources par l'intelligence artificielle, qui restent néanmoins similaires – ou comparables dans leurs besoins techniques – d'un projet à l'autre. L'utilisation d'un GPU pour la vision artificielle compte parmi les besoins matériels qui se traduisent entre les projets, ainsi, dans une optique de programmation modulaire, une API externe à l'application VHS/EIDA a été développée pour répondre à ce besoin, sans être hautement spécifique aux besoins de ces projets comme pourrait l'être un module intégré à l'application. extractorAPI, l'API développée dans ce cadre, est ainsi conçue, dès son cahier des charges, pour répondre aux besoins du projet EIDA tout en restant réemployable dans d'autres contextes – sont ainsi utilisés, pour sa construction, des outils libres qui en standardisent le développement, avec la vocation de produire une API aussi flexible que robuste.

Chapitre 6

Intégrer la vision artificielle aux pratiques des chercheurs

Le chapitre précédent établit, du point de vue du *back end*, les besoins techniques pour l'intégration de la vision artificielle à une chaîne de traitement des sources historiques : ces préoccupations en termes d'architecture et de puissance de calcul sont celles des équipes d'ingénierie, et existent en parallèle des préoccupations qui concernent plus directement les utilisateurs, et impactent réellement les pratiques des chercheurs. Ce chapitre revient ainsi sur la chaîne de traitement des sources historiques dans sa totalité, depuis la numérisation fournie par l'utilisateur jusqu'aux résultats retournés pour le traitement par les chercheurs, et sur les échanges avec les utilisateurs, en termes d'interface mais également en termes de médiation.

6.1 De la numérisation à l'annotation : l'automatisation du traitement des sources

6.1.1 Chaîne de traitement des sources historiques

Du manuscrit aux diagrammes : chaîne de traitement des sources astronomiques

La chaîne de traitement proposée par EIDA (fig. 6.1) propose ainsi une alternance d'étapes automatisées, en bleu sur le schéma, et d'étapes d'analyse par les chercheurs du projet.

Traitements automatiques, traitements manuels

qu'est-ce qui peut être automatisé, qu'est-ce qui ne l'est pas, intervention des chercheurs, etc etc

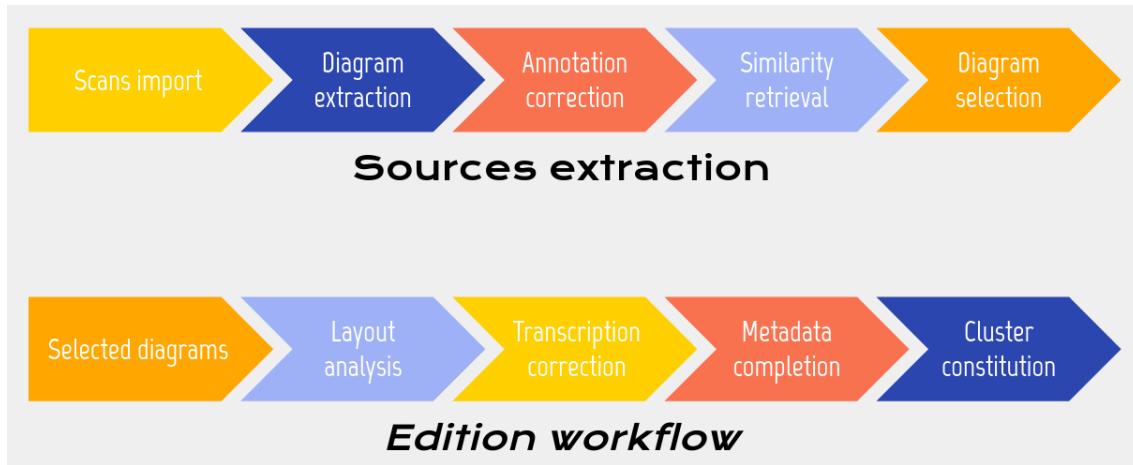


FIGURE 6.1 – *Workflow* de traitement des sources du projet EIDA

6.1.2 Échanges et transformation des données

Sources d'entrée, sources de sortie

de l'image au manifeste à l'image, au txt puis sas

Traitement des résultats de la détection

après avoir obtenu les diagrammes : deux possibilités, plus de traitements automatiques, cf partie 3, ou résultat en lui même

par exemple détection d'objets permet de constituer un corpus qui sera traité manuellement ?

6.1.3 Créer des interfaces pour les étapes manuelles

Correction par les chercheurs : au-delà de l'entraînement

expliquer que la correction se fait pas juste pour l'entraînement mais que l'interface permettra plus tard aux chercheurs de sélectionner les diagrammes qui les intéressent

Interface d'annotation

interface pour les chercheurs, pour la correction : captures d'écran réelles réflexions techniques pour la praticité

6.2 Médiation et documentation

6.2.1 Échanges avec les chercheurs

ANNEXE PPT SÉMINAIRE DH

Séminaires DH

Ateliers d'annotation

6.2.2 Documentation et Wiki : penser le réemploi du code

ANNEXE PAGE WIKI

Commentaires dans le code

Rédaction du Wiki

Troisième partie

Perspectives pour le traitement des sources : vers un outil pour l'édition et la recherche

Chapitre 7

Éditer des diagrammes : vectorisation et édition critique

7.1 Édition critique des diagrammes astronomiques

7.1.1 Historique

Sous-sous-section 1

Sous-sous-section 2

7.1.2 Éditer les diagrammes en format numérique

Sous-sous-section 1

Sous-sous-section 2

7.2 De l'image aux vecteurs : la vision artificielle pour l'édition numérique

7.2.1 Automatiser la transcription

Sous-sous-section 1

Diagrammes en SVG

7.2.2 Données d'entraînement et annotations pour la vectorisa- tion

Sous-sous-section 1

Sous-sous-section 2

Chapitre 8

Regroupement par similarité et *clustering*

8.1 *Similarity retrieval* et navigation des corpus

8.1.1 Principe et méthodes

Sous-sous-section 1

Sous-sous-section 2

8.1.2 Fouiller des corpus par l'image

Sous-sous-section 1

Sous-sous-section 2

8.2 Le *clustering* comme outil pour les chercheurs

8.2.1 Sous-section 1

Sous-sous-section 1

Sous-sous-section 2

8.2.2 Sous-section 2

Sous-sous-section 1

Sous-sous-section 2

Conclusion

Étudier et exploiter les résultats automatiques : limites et perspectives pour les sciences historiques

Annexes

Annexe A

Prepare custom data for training using the YOLOv5 workflow

A.1 What is yolov5 ?

- Built to detect objects in images
- Trained with a dataset of real images (ImageNet)
- Default model based on 80 object classes

The goal of the training is to **adapt the model to our images and our classes**, for it to detect the objects we want it to.

A.2 Training dataset

- **Annotate the objects** of interest on the images to **create labels** : the objects are defined by the **position and dimensions of the rectangular box** surrounding them ;
- The label file is a **.txt** file ;
- In the label file, each object corresponds to **one row : class x_center y_center width height**
- Organize labels and images in directories : **one directory for images and one for labels**, each containing a **training, validation and testing dataset** ;
- **Training is 80% of the dataset**, validation is either 20% or 10% if a testing dataset (10%) is needed ;
- The training and validation datasets must be **different data** ;
- The image file and labels file must **have the same name**.

A.3 Config file

The `dataset.yaml` is the **configuration file for the training**. It contains :

- The **dataset directory path** ;
- The relative paths to the training, validation and test directories ;
- A **class names dictionary** : in our case, the only class is 0: **illustration**.

```
path: ../dataset # dataset root dir
train: images/train # training images (relative to 'path')
val: images/val # validation images (relative to 'path')
test: # test images (optional)

# Classes names:
0: illustration
```

A.4 Notes

- The training dataset is made of the **images the researchers annotate** ;
- Our model needs to be able to detect **each illustration individually** rather than extracting illustrated pages as a unique illustration ;
- Our model needs to be able to differentiate tables from diagrams ;
- The model is trained on both images of **manuscripts and printed books** : does it impact the quality of the extraction ?

Annexe B

Modèles de données des applications VHS et EIDA

B.1 Modèle de données initial de l'application VHS

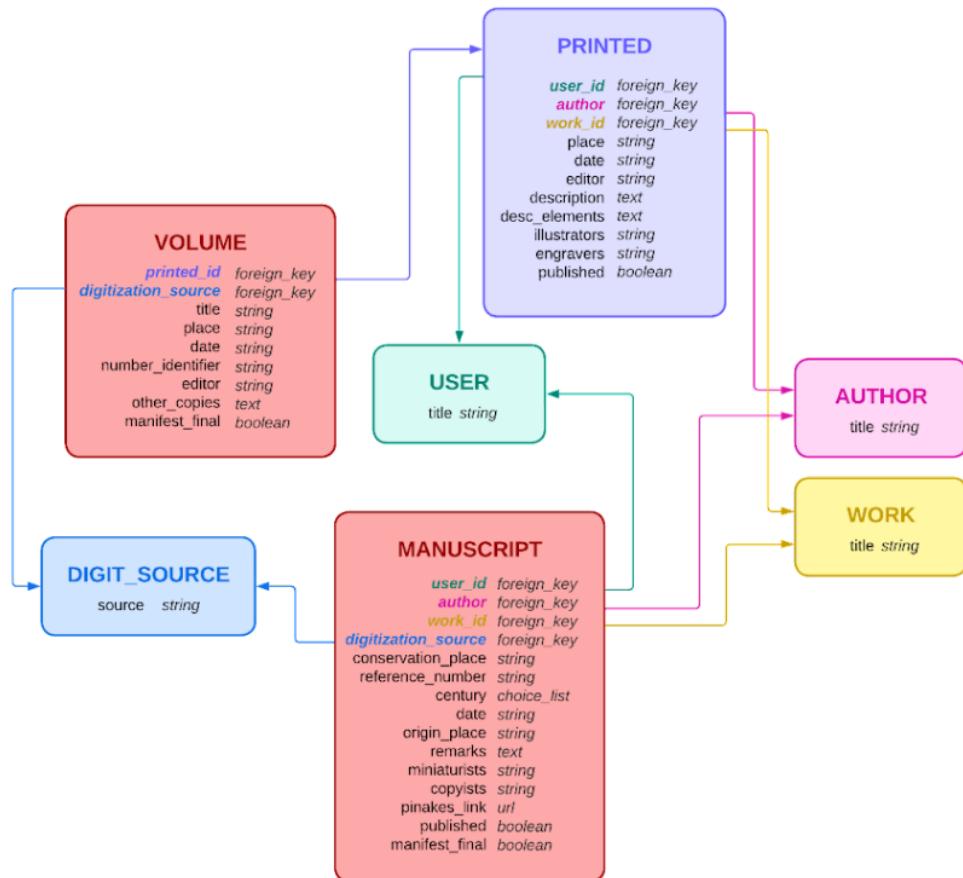
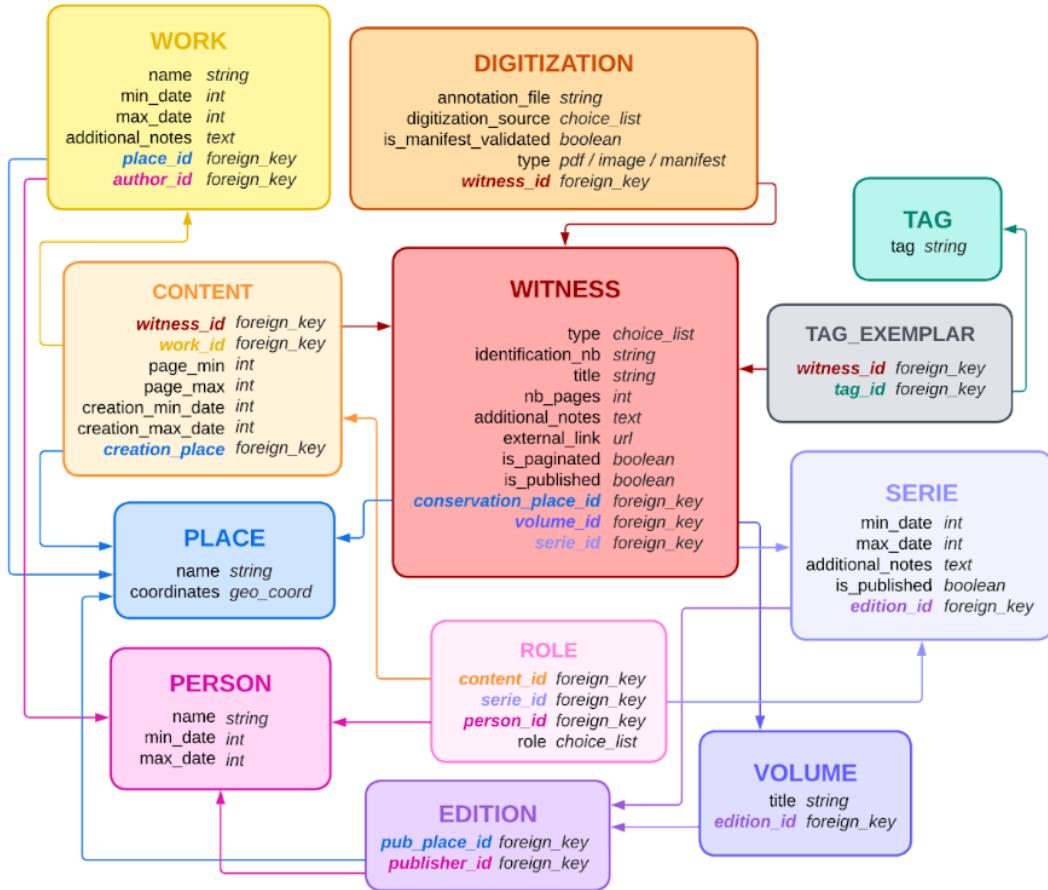


FIGURE B.1 – Modèle de données de l'application VHS avant refonte

B.2 Modèle de données de l'application VHS/EIDA



(a) Nouveau modèle de données de l'application EIDA appliqué à VHS

Annexe C

extractorAPI : cahier des charges

C.1 Fonctionnalités

L’API permet de déplacer l’annotation des images envoyées par les utilisateur-rice-s de EIDA sur le GPU : dans une logique de développement modulaire, le modèle entraîné n’est pas intégré directement à l’application, mais dans une API indépendante. Cette séparation du corps de l’application et du module générant les annotations permet d’éviter le ralentissement général de l’exécution de l’application sur le serveur Web : l’utilisation d’une API permet de profiter de la capacité de calcul du GPU lors de l’inférence, alors que l’application est développée sur un serveur différent. La communication en SSH entre le GPU et l’application EIDA n’étant pas satisfaisante du point de vue de la sécurité, le développement d’une API permet de gérer ces échanges par une voie sécurisée.

Du point de vue de l’open source, cette forme de développement permet la création d’un outil plus flexible, qui pourra être réutilisé par d’autres projets en fonction de leurs besoins et de leur propre architecture.

L’API reçoit une requête HTTP de la part de l’application EIDA lorsque l’utilisateur-rice envoie une nouvelle numérisation de son témoin dans la base de données via un formulaire. Le manifeste généré par l’application est envoyé à un API endpoint qui l’enregistre et en extrait les images, qui sont ensuite traitées par le modèle. Les annotations générées sont retournées à l’application en réponse à la requête, et peuvent ainsi être visualisées par l’utilisateur-rice.

L’objectif d’extractorAPI – par rapport à notre *workflow* actuel – est d’automatiser la détection des diagrammes par le modèle, pour que celle-ci puisse être faite à la demande de l’utilisateur-rice, sans intervention humaine, et que les annotations lui soient retournées automatiquement après envoi des numérisations.

C.1.1 *Workflow*

1. Envoi d’une numérisation par l’utilisateur-rice sur l’application EIDA ;

2. Création automatique d'un manifeste IIIF à partir des images ou du manifeste ;
3. Envoi du manifeste IIIF via requête HTTP POST à l'API endpoint ;
4. Appel de la fonction Python :
5. Vérification et enregistrement du manifeste ;
6. Enregistrement des images issues du manifeste pour leur traitement (IIIF downloader) ;
7. Traitement des images par le modèle : création des annotations (fichiers .txt) ;
8. *Return* : envoi des annotations à l'application ;
9. Affichage des images annotées à l'utilisateur-rice.

L'architecture actuelle de l'application implique d'enregistrer deux fois les images : une première fois dans l'application EIDA pour générer le manifeste IIIF, puis une seconde fois sur le GPU pour lancer l'inférence. On peut envisager le déplacement du serveur Cantaloupe sur le GPU pour éviter la duplication de cette tâche.

C.1.2 Outils

L'API est développée avec Flask, plus simple et flexible que Django pour la construction d'une API légère, et le modèle est déployé à l'aide du module PyTorch.

Modèle

Le modèle utilisé a pour base docExtractor – appuyé sur le réseau de neurones U-Net –, un modèle entraîné sur des images synthétiques de manuscrits générées par l'algorithme SynDoc et sur ImageNet. Le script d'inférence utilisé est celui de YOLOv5. Il est nécessaire de prendre en compte, dans le développement de l'API, que d'autres modèles pour la détection d'objets seront testés, et que sera conservé le modèle produisant les meilleurs résultats. Le modèle final sera un modèle entraîné, idéalement commun avec vhs. Il est cependant envisageable d'avoir un modèle par type de source (manuscrit ou imprimé), ce qui nécessiterait d'adapter la structure de l'API pour pouvoir choisir le modèle utilisé en fonction de la requête.

Task queues

Pour éviter la surcharge de l'API et gérer la réception de multiples requêtes, il est nécessaire de mettre en place un task manager comme Celery, qui traite en tâche de fond les requêtes reçues, et permet d'imposer qu'elles soient traitées successivement.

C.1.3 Échange de données

Données envoyées

Le manifeste IIIF généré par EIDA. S'il est décidé après le réentraînement de conserver un modèle pour chaque type de témoin (manuscrit ou imprimé), il est nécessaire d'envoyer une donnée sur le type de support pour choisir le modèle correspondant.

Données reçues

Un fichier d'annotations au format `.txt` par manifeste. Celui-ci contient, pour chaque image :

- le numéro de l'image généré par une fonction `enumerate()`,
- le nom du fichier image annoté,
- les coordonnées de la détection au format `x y width height`.

Par exemple, pour deux objets détectés sur la 21e image du manuscrit 22 :

```

1  21_ms22_0021.jpg
2  514 907 1700 1685
3  2424 831 1441 1423

```

C.1.4 Sécurité

Pour assurer la sécurité de l'API, une authentification avec un token a été envisagée. Pour plus de simplicité, l'API utilise un décorateur pour **restreindre les hôtes autorisés** à envoyer des requêtes aux *endpoints* : seuls l'hôte `obspm.fr` peut envoyer des requêtes pour l'inférence, évitant ainsi un risque de surcharge par des requêtes envoyées depuis un autre hôte.

C.1.5 Interactions

L'annotation des images est automatique après envoi de la numérisation ; il est cependant nécessaire de prendre en compte le temps de réponse de la requête du point de vue de l'utilisateur-rice.

L'application EIDA répond actuellement à cette problématique en affichant le message flash suivant : “Le processus de conversion de.s fichier.s PDF en images et/ou d'extraction des images à partir de manifeste.s externe.s est en cours. Veuillez patienter quelques instants pour corriger les annotations automatiques.”

On peut envisager de désactiver les boutons de visualisation et d'édition du manifeste jusqu'à réception de la réponse à la requête.

Annexe D

Script YOLOv5 pour lancer la détection d'objet

Le script suivant est une reproduction du script `detect_vhs.py` utilisé dans extractorAPI pour lancer la détection d'objet dans les images envoyées par l'application :
https://github.com/jnorindr/extractorAPI/blob/main/yolov5/detect_vhs.py

```
1 # -*- coding: utf-8 -*-
2 # YOLOv5 by Ultralytics, GPL-3.0 license
3 import argparse
4 import os
5 import platform
6 import sys
7 from pathlib import Path
8 import torch
9
10 FILE = Path(__file__).resolve()
11 ROOT = FILE.parents[0] # YOLOv5 root directory
12 if str(ROOT) not in sys.path:
13     sys.path.append(str(ROOT)) # add ROOT to PATH
14 ROOT = Path(os.path.relpath(ROOT, Path.cwd())) # relative
15
16 from models.common import DetectMultiBackend
17 from yolov5.utils.dataloaders import IMG_FORMATS, VID_FORMATS, LoadImages
18     , LoadScreenshots, LoadStreams
19 from yolov5.utils.general import (Profile, check_file, check_img_size,
20     check_imshow, cv2,
21     increment_path, non_max_suppression, scale_boxes, strip_optimizer,
22     xyxy2xywh)
23 from yolov5.utils.plots import Annotator, colors, save_one_box
24 from yolov5.utils.torch_utils import select_device, smart_inference_mode
25
26 @smart_inference_mode()
```

```

25 def run_vhs(
26     weights=ROOT / 'yolov5s.pt',    # model path or triton URL
27     source=ROOT / 'data/images',   # file/dir/URL/glob/screen/0(webcam)
28     data=ROOT / 'data/coco128.yaml', # dataset.yaml path
29     imgsz=(640, 640),   # inference size (height, width)
30     conf_thres=0.25,   # confidence threshold
31     iou_thres=0.45,   # NMS IOU threshold
32     max_det=1000,    # maximum detections per image
33     device='',       # cuda device, i.e. 0 or 0,1,2,3 or cpu
34     view_img=False,  # show results
35     save_txt=False,  # save results to *.txt
36     save_conf=False, # save confidences in --save-txt labels
37     save_crop=False, # save cropped prediction boxes
38     nosave=False,   # do not save images/videos
39     classes=None,   # filter by class: --class 0, or --class 0 2 3
40     agnostic_nms=False, # class-agnostic NMS
41     augment=False,  # augmented inference
42     visualize=False, # visualize features
43     update=False,   # update all models
44     project=ROOT / 'runs/detect', # save results to project/name
45     name='exp',      # save results to project/name
46     exist_ok=False,  # existing project/name ok, do not increment
47     line_thickness=3, # bounding box thickness (pixels)
48     hide_labels=False, # hide labels
49     hide_conf=False, # hide confidences
50     half=False,     # use FP16 half-precision inference
51     dnn=False,      # use OpenCV DNN for ONNX inference
52     vid_stride=1,   # video frame-rate stride
53     anno_file="annotation.txt", # annotation in which write the output
54     img_nb=1        # increment of the image
55 ):
56     source = str(source)
57     save_img = not nosave and not source.endswith('.txt') # save inference
58     images
59     is_file = Path(source).suffix[1:] in (IMG_FORMATS + VID_FORMATS)
60     is_url = source.lower().startswith(('rtsp://', 'rtmp://', 'http://',
61                                         'https://'))
62     webcam = source.isnumeric() or source.endswith('.txt') or (is_url and
63     not is_file)
64     screenshot = source.lower().startswith('screen')
65     if is_url and is_file:
66         source = check_file(source) # download
67
68     # Directories
69     save_dir = increment_path(Path(project) / name, exist_ok=exist_ok) # increment run

```

```

67     # (save_dir / 'labels' if save_txt else save_dir).mkdir(parents=True,
68     exist_ok=True) # make dir
69
70     # Load model
71     device = select_device(device)
72     model = DetectMultiBackend(weights, device=device, dnn=dnn, data=data,
73     fp16=half)
74     stride, names, pt = model.stride, model.names, model.pt
75     imgsz = check_img_size(imgsz, s=stride) # check image size
76
77     # Dataloader
78     bs = 1 # batch_size
79     if webcam:
80         view_img = check_imshow(warn=True)
81         dataset = LoadStreams(source, img_size=imgsz, stride=stride, auto=pt,
82         vid_stride=vid_stride)
83         bs = len(dataset)
84     elif screenshot:
85         dataset = LoadScreenshots(source, img_size=imgsz, stride=stride, auto=pt)
86     else:
87         dataset = LoadImages(source, img_size=imgsz, stride=stride, auto=pt,
88         vid_stride=vid_stride)
89         vid_path, vid_writer = [None] * bs, [None] * bs
90
91     # Run inference
92     model.warmup(imgsz=(1 if pt or model.triton else bs, 3, *imgsz)) # warmup
93     seen, windows, dt = 0, [], (Profile(), Profile(), Profile())
94     for i, (path, im, im0s, vid_cap, s) in enumerate(dataset, start=1):
95         with open(anno_file, 'a') as f:
96             f.write(f"{img_nb} {path.split('/')[-1]}\n") # write canvas and
97             image name
98         with dt[0]:
99             im = torch.from_numpy(im).to(model.device)
100            im = im.half() if model.fp16 else im.float() # uint8 to fp16/32
101            im /= 255 # 0 - 255 to 0.0 - 1.0
102            if len(im.shape) == 3:
103                im = im[None] # expand for batch dim
104
105    # Inference
106    with dt[1]:
107        visualize = increment_path(save_dir / Path(path).stem, mkdir=True) if
108        visualize else False
109        pred = model(im, augment=augment, visualize=visualize)
110
111    # NMS

```



```

144     line = (cls, *xywh, conf) if save_conf else (cls, *xywh) # 
label format
145     with open(f'{txt_path}.txt', 'a') as f:
146         f.write((f'{line} ') * len(line)).rstrip() % line + '\n'
147
148     if save_img or save_crop or view_img: # Add bbox to image
149         c = int(cls) # integer class
150         label = None if hide_labels else (names[c] if hide_conf else f'{names[c]} {conf:.2f}')
151         annotator.box_label(xyxy, label, color=colors(c, True))
152     if save_crop:
153         save_one_box(xyxy, imc, file=save_dir / 'crops' / names[c] / f'{p.stem}.jpg', BGR=True)
154
155     # Stream results
156     im0 = annotator.result()
157     if view_img:
158         if platform.system() == 'Linux' and p not in windows:
159             windows.append(p)
160             cv2.namedWindow(str(p), cv2.WINDOW_NORMAL | cv2.WINDOW_KEEPRATIO)
# allow window resize (Linux)
161             cv2.resizeWindow(str(p), im0.shape[1], im0.shape[0])
162             cv2.imshow(str(p), im0)
163             cv2.waitKey(1) # 1 millisecond
164
165     # Save results (image with detections)
166     if save_img:
167         if dataset.mode == 'image':
168             cv2.imwrite(save_path, im0)
169         else: # 'video' or 'stream'
170             if vid_path[i] != save_path: # new video
171                 vid_path[i] = save_path
172                 if isinstance(vid_writer[i], cv2.VideoWriter):
173                     vid_writer[i].release() # release previous video writer
174                 if vid_cap: # video
175                     fps = vid_cap.get(cv2.CAP_PROP_FPS)
176                     w = int(vid_cap.get(cv2.CAP_PROP_FRAME_WIDTH))
177                     h = int(vid_cap.get(cv2.CAP_PROP_FRAME_HEIGHT))
178                 else: # stream
179                     fps, w, h = 30, im0.shape[1], im0.shape[0]
180                     save_path = str(Path(save_path).with_suffix('.mp4')) # force
*.mp4 suffix on results videos
181                     vid_writer[i] = cv2.VideoWriter(save_path, cv2.
VideoWriter_fourcc(*'mp4v'), fps, (w, h))
182                     vid_writer[i].write(im0)
183
184     if update:

```

```
185     strip_optimizer(weights[0]) # update model (to fix  
SourceChangeWarning)
```

Annexe E

Annotate images from EiDA with extractorAPI

Ce document est extrait du Wiki de l'API, disponible sur GitHub en accompagnement du code. Il a été rédigé pour documenter les interactions entre extractorAPI et l'application EIDA, pour faciliter la réutilisation de l'API par des projets tiers. Le document est disponible à l'adresse suivante : <https://github.com/jnorindr/extractorAPI/wiki/Annotate-images-from-EiDA-with-extractorAPI>

E.1 Workflow

1. User submits a witness
2. Verification : did the user send a digitization ?
3. A flash message informs the user of the delay for receiving the annotations
4. Images are downloaded and IIIF manifest is generated
5. POST request is sent from EIDA to extractorAPI with manifest link and type of witness as parameters
6. API : images are downloaded, detection is launched and annotation file is generated
7. POST request is sent from extractorAPI to EIDA with annotation file
8. Annotations are indexed to be displayed for the user

E.2 Settings

E.2.1 EiDA

In the `.env` file, the `EXAPI` variable must be set with the URL of the API.

```
1 EXAPI=<gpu-api-address>
```

In the settings file `settings.py`, complete the `GPU_PORT` variable so the `GPU_URL` variable can be set.

```
1 GPU_PORT = 5000
2 GPU_URL = f"{ENV('EXAPI')}: {GPU_PORT}"
```

E.2.2 extractorAPI

In the `.env` file, set the `CLIENT_APP_URL` variable with the URL of the app.

```
1 CLIENT_APP_URL="<url>"
```

E.3 Sending for detection

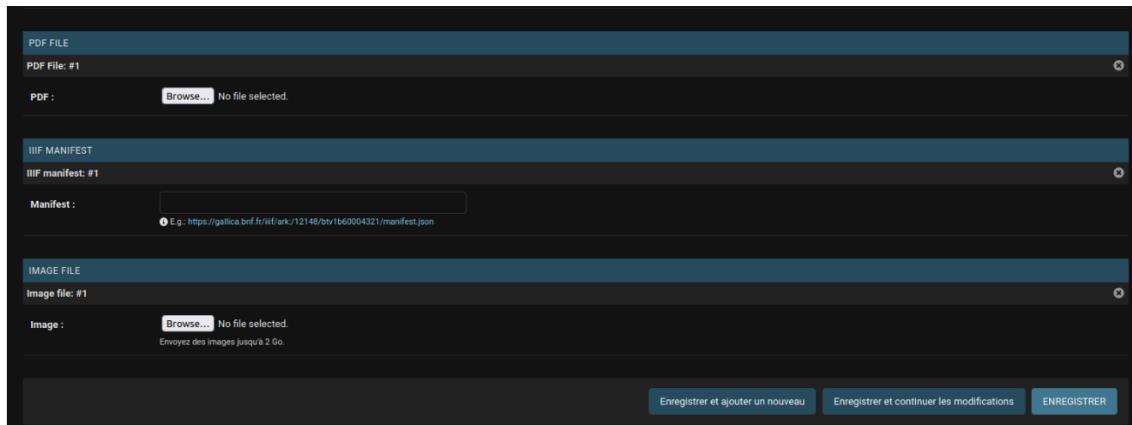


FIGURE E.1 – Capture d’écran du formulaire d’envoi d’une numérisation dans l’application EIDA

E.3.1 IIIF manifest

The user creates a **witness**, fills in the form and submit a digitization (either a PDF file, a IIIF manifest or images).

The images are downloaded and the **eida iiif manifest** is generated : the request to the API is not sent until all of the images are downloaded, to avoid errors and annotation of incomplete manifests.

E.3.2 annotate_wit function

To send requests to extractorAPI from the EIDA application, we created a function in `vhs-platform/vhsapp/utils/iiif/annotation.py`. The function sends a **POST request** to the API endpoint `/run_detect`, which launches detection on the images from the digitization.

The request takes as arguments the **url of the manifest** we want annotated, and the **abbreviation** of the witness type.

The **event** argument is used with the method `.wait()` to ensure the function **waits for an event to be set** before sending the request to the API – this event will be set when the images from the previous step are all downloaded.

```

1 def annotate_wit(event, witness_id, wit_abbr=MS_ABBR, version=
2   MANIFEST_AUTO):
3   wit_type = MS if wit_abbr == MS_ABBR else VOL
4   api_endpoint = f"{GPU_URL}/run_detect"
5
6   manifest_url = (
7     f"{VHS_APP_URL}/{APP_NAME}/iiif/{version}/{wit_type}/{witness_id}/
8     manifest.json"
9   )
10  data = {"manifest_url": manifest_url, "wit_abbr": wit_abbr}
11
12  event.wait()
13  requests.post(url=api_endpoint, data=data)
14
15  return print(f"Witness {witness_id} sent for diagram extraction")

```

E.3.3 Automating the annotation request

To launch annotation automatically after the user submits a digitization, we added a **background task** to the `save` method of `Manifests`, `Pdf` and `Image` objects.

We use **threading** to avoid an interruption of the other tasks of the app while waiting for a response to the request.

User submitted a IIIF manifest

For manifests, we modified the `save` method of the `ManifestManuscript` and the `ManifestVolume` classes in `vhs/vhs-platform/vhsapp/models/digitization.py`. Ideally, we would have a unique `save` method for both types.

An `event` object is created and can be passed through the various **threads** : the first thread `t` calls the function `extract_images_from_iiif_manifest` (in `vhs/vhs-platform/vhsapp/utils/iiif/download.py`) to download images and **set the event** when the function is done running.

```

1 def extract_images_from_iiif_manifest(manifest_url, witness_ref, event):
2   """
3   Extract all images from an IIIF manifest
4   """
5   downloader = IIIFDownloader(manifest_url, witness_ref)
6   downloader.run()

```

```
7     event.set()
```

`event.set()` returns a boolean value : `event` is `True` when the function has **run to its completion**. Because we used an `event.wait()` in the `annotate_wit` function, it cannot run unless this value is `True`.

When the image extraction is done and the event is set, the thread `t2` starts, therefore **sending the newly generated manifest to the api for annotation**.

```
1  class ManifestManuscript(Manifest):
2      manuscript = models.ForeignKey(Manuscript, on_delete=models.CASCADE)
3
4      def save(self, *args, **kwargs):
5          # Call the parent save method to save the model
6          super().save(*args, **kwargs)
7
8          event = threading.Event()
9
10         # Run the async extraction of images from an IIIF manifest in the
11         # background using threading
12         t = threading.Thread(
13             target=extract_images_from_iiif_manifest,
14             args=(
15                 self.manifest,
16                 f'{MS_ABBR}{self.manuscript.id}',
17                 event,
18             ),
19         )
20         t.start()
21
22         t2 = threading.Thread(
23             target=annotate_wit,
24             args=(
25                 event,
26                 f'{self.manuscript.id}',
27                 f'{MS_ABBR}',
28             ),
29         )
30         t2.start()
```

User submitted a PDF file

For PDF files, the process is the same as for manifests. The `event` is set at the end of the `pdf_to_img` function, when all pages are converted to images, in `vhs/vhs-platform/vhsapp/utils/functions.py`.

```
1  def pdf_to_img(event, pdf_name):
2      """
```

```
3 Convert the PDF file to JPEG images
4
5 pdf_path = f"{BASE_DIR}/{MEDIA_PATH}/{pdf_name}"
6
7 # e.g. pdf_name = "volumes/pdf/filename.pdf" => "filename"
8 pdf_name = pdf_path.split("/")[-1].split(".")[0]
9 pdf_info = pdfinfo_from_path(pdf_path, userpw=None, poppler_path=None)
10 page_nb = pdf_info["Pages"]
11 step = 2
12 try:
13     for img_nb in range(1, page_nb + 1, step):
14         batch_pages = convert_from_path(
15             pdf_path,
16             dpi=300,
17             first_page=img_nb,
18             last_page=min(img_nb + step - 1, page_nb),
19         )
20         for page in batch_pages:
21             save_img(page, f"{pdf_name}_{img_nb:04d}.jpg")
22             img_nb += 1
23         event.set()
24 except Exception as e:
25     log(f"[pdf_to_img] Failed to convert {pdf_name}.pdf to images:\n{e}"
```

In `vhs/vhs-platform/vhsapp/models/digitization.py`, the `save` method of the `Pdf` class was modified to start a second thread when the function has run.

```
1 class Pdf(Digitization):
2     class Meta:
3         verbose_name = "PDF File"
4         verbose_name_plural = "PDF Files"
5         abstract = True # TODO: make this class not abstract
6
7     def __str__(self):
8         return self.pdf.name
9
10    def save(self, *args, **kwargs):
11        # Call the parent save method to save the model
12        super().save(*args, **kwargs)
13        # Run the PDF to image async conversion task in the background using
14        # threading
15        # t = threading.Thread(target=self.to_img())
16
17        event = threading.Event()
18
19        t = threading.Thread(target=pdf_to_img, args=(event, f"{self.pdf.name}")
20        ,))
21        t.start()
```

```

20
21     t2 = threading.Thread(
22         target=annotate_wit,
23         args=(
24             event,
25             f"{self.volume.id} if 'vol' in self.pdf.name else self.manuscript.
26             id}",
27             f"{VOL_ABBR} if 'vol' in self.pdf.name else MS_ABBR}",
28         ),
29     )
30     t2.start()

```

User submitted images

```

1  def save(self, *args, **kwargs):
2      if self.image:
3          img = Image.open(self.image)
4          # Check if the image format is not JPEG
5          if img.format != "JPEG":
6              # Convert the image to JPEG format
7              self.image = convert_to_jpeg(self.image)
8          # Call the parent save method to save the model
9          super().save(*args, **kwargs)
10
11     event = threading.Event()
12
13     t = threading.Thread(
14         target=annotate_wit,
15         args=(
16             event,
17             f"{self.volume.id} if 'vol' in self.image.name else self.manuscript.
18             id}",
19             f"{VOL_ABBR} if 'vol' in self.image.name else MS_ABBR}",
20         ),
21     )
22     t.start()

```

E.4 Receiving annotations

For the annotations to be returned from extractorAPI to the eida application after detection, we create an **endpoint** in vhs/vhs-platform/vhs/urls.py.

```

1  path(
2      f"[{APP_NAME}]/{<str:wit_type>}/{<int:wit_id>}/annotate/",
3      receive_anno,

```

```

4     name="receive-annotations",
5     ),

```

The endpoint calls the function `receive_anno` in `vhs/vhs-platform/vhsapp/utils/views.py` which receives a `file` from a **POST request from extractorAPI**. When the file is received, **if it is a text file**, its content is **written in a file in the annotation directory**. If the file already exists, the content is rewritten.

When the annotation file is saved, the annotations are indexed by the function `index_anno`.

```

1  def receive_anno(request, wit_id, wit_type):
2      if request.method == "POST":
3          annotation_file = request.FILES["annotation_file"]
4          file_content = annotation_file.read()
5
6          if is_text_file(file_content):
7              anno_path = f"{BASE_DIR}/{MEDIA_PATH}/{MS_ANNO_PATH} if wit_type =="
8                  'manuscript' else VOL_ANNO_PATH}"
9
10         try:
11             with open(f"{anno_path}/{wit_id}.txt", "w+b") as f:
12                 f.write(file_content)
13
14         except Exception as e:
15             log(f"[receive_anno] Failed to open received annotations for {"
16                 wit_type} #{wit_id}: {e}")
17
18         manifest_url = (f"{VHS_APP_URL}/{APP_NAME}/iiif/v2/{wit_type}/"
19                         "{wit_id}/manifest.json")
20
21         try:
22             index_anno(manifest_url, wit_type, wit_id)
23         except Exception as e:
24             log(f"[receive_anno] Failed to index annotations for {wit_type}"
25                 "#{wit_id}: {e}")
26
27         return JsonResponse({"message": "Annotation received and indexed."})
28
29     else:
30         return JsonResponse({"message": "Invalid request. File is not a"
31                         "text file."}, status=400)
32
33     else:
34         return JsonResponse({"message": "Invalid request."}, status=400)

```

E.5 Indexing annotations

In vhs/vhs-platform/vhsapp/utils/iiif/annotation.py, the function `index_anno` launches the indexation of the annotations, to go **from a .txt file to an annotated manifest**. A **GET request** is sent to retrieve the JSON content of the manifest, which is sent to the **annotation server** through a **POST request**.

```

1 def index_anno(manifest_url, wit_type, wit_id):
2     try:
3         manifest = requests.get(manifest_url)
4         manifest_content = manifest.json()
5     except Exception as e:
6         log(f"[index_anno]: Failed to load manifest for {wit_type} n°{wit_id}: {e}")
7
8         requests.post(f"{SAS_APP_URL}/manifests", json=manifest_content)
9
10    try:
11        requests.get(f"{VHS_APP_URL}/{APP_NAME}/iiif/v2/{wit_type}/{wit_id}/
12        populate/")
13    except Exception as e:
14        log(f"[index_anno]: Failed to index {wit_type} n°{wit_id}: {e}")

```

Liste des acronymes

- API *Application Programming Interface.* 17, 21–23, 26, 51, 53, 56–65, 85–87, 95–98, 105
- CNN Convolutional Neural Nets. 42
- CPU *Central Processing Unit.* 56
- CSV *Comma-separated values.* 6
- DIPS *DISHAS Interactive Parameter Squeezer.* 5
- DISHAS *Digital Information System for the History of Astral Sciences.* 4–7, 11
- DTI *DISHAS Table Interface.* 5
- EIDA *Editing and analysing hIstorical astronomical Diagrams with Artificial intelligence.* 3–8, 11, 12, 16, 25, 30–32, 34, 35, 37, 44–46, 48, 50–61, 63–65, 67, 68, 84–87, 95, 96, 100, 105
- GPU *Graphics Processing Unit.* 51, 56–60, 63–65, 85, 86
- HTML *HyperText Markup Language.* 6
- HTR *Handwritten Text Recognition.* 6
- HTTPS *Hypertext Transfer Protocol Secure.* 22
- HTTP *Hypertext Transfer Protocol.* 22, 58, 59, 62, 63, 85, 86
- IIIF *International Image Interoperability Framework.* 15, 21–27, 55, 57, 59, 62–64, 86, 87, 95, 96, 105
- IMAGINE Laboratoire d’Informatique Gaspard Monge. 6–8, 34, 35, 37, 46
- INHA Institut national d’histoire de l’art. 18
- ISCD Institut des sciences du calcul et des données. 6
- JSON *JavaScript Object Notation.* 22, 23, 59, 62, 63, 102
- REST *Representational state transfer.* 58–60
- SSH *Secure Shell.* 58, 85
- SVG *Scalable Vector Graphics.* 33
- SYRTE Systèmes de Référence Temps-Espace. 3, 6
- URI *Uniform Resource Identifier.* 22

- URL *Uniform Resource Locator*. 58, 95–97
- VHS Vision artificielle et analyse Historique de la circulation de l'illustration Scientifique. 6–8, 30, 32, 34, 36, 44–46, 48, 51–55, 57, 65, 83, 84, 86, 105
- XML *eXtensible Markup Language*. 6
- YOLO *You Only Look Once*. 44, 49, 62
- BnF Bibliothèque nationale de France. 16, 18, 21
- EnHerit *Enhancing Heritage Image Databases*. 7
- Rmn-Grand Palais Réunion des musées nationaux-Grand Palais. 17

Table des figures

1.1	Modèles lunaires de tradition arabe, byzantine et latine	11
2.1	Modèle de données IIIF pour la description d'un objet	23
2.2	Comparatif des ressources contenant les images de deux manifestes IIIF . .	25
3.1	Cas limites rencontrés lors de l'annotation des images du projet VHS . . .	36
4.1	Exemples tirés d'un jeu de données généré avec SynDoc	46
4.2	Diagrammes détectés par les modèles YOLOv5 et docExtractor dans deux pages du manuscrit hebr. Or. fol. 1054 de la Staatsbibliotek (Berlin). . . .	47
4.3	Objets détectés sur la page 6 du manuscrit 5 (Alm. 1) de la Gurukul Kangri Haridwar Collection	48
5.1	Interface administrateur de l'application EIDA	53
5.2	Schéma de l'architecture de l'application EIDA et de l'API extractorAPI . .	57
6.1	<i>Workflow</i> de traitement des sources du projet EIDA	68
B.1	Modèle de données de l'application VHS avant refonte	83
E.1	Capture d'écran du formulaire d'envoi d'une numérisation dans l'application EIDA	96

Table des matières

Résumé	iii
Remerciements	v
Introduction	xiii
I Construire un corpus de numérisations pour le traitement par vision artificielle	1
1 Le projet EiDA	3
1.1 Contexte et objectifs du projet	3
1.1.1 Un projet de recherche en humanités numériques	3
1.1.2 Avant EiDA : le projet DISHAS	5
1.1.3 Acteurs du projet	6
1.2 Sources primaires	8
1.2.1 L'astronomie ptoléméenne : naissance et diffusion	8
1.2.2 Les diagrammes	10
2 Images et interopérabilité	15
2.1 L'image comme source	15
2.1.1 Construire un corpus d'images : enjeux et méthodes	15
2.1.2 Images et données ouvertes	19
2.2 Le standard IIIF	21
2.2.1 IIIF et les données ouvertes	21
2.2.2 IIIF, un modèle universel ?	24
3 Corpus historiques et jeux de données	29
3.1 Dimensions et cadre	29
3.1.1 La tentative de l'exhaustivité	29
3.1.2 Automatiser le traitement de corpus massifs	31
3.2 Objectifs scientifiques et possibilités numériques	34

3.2.1	Dialogue entre chercheurs et ingénieurs	34
3.2.2	Définir les pratiques	35
II	De l'image à l'objet : intégrer l'apprentissage profond au traitement des sources historiques	39
4	L'apprentissage profond	41
4.1	Réseaux de neurones et <i>computer vision</i>	41
4.1.1	Des « neurones » pour le <i>deep learning</i>	41
4.1.2	Réseaux de neurones à convolution	42
4.2	Modèles de vision <i>off-the-shelf</i>	43
4.2.1	YOLOv5, docExtractor et autres modèles	43
4.2.2	Entraînement et <i>fine-tuning</i> d'un modèle de détection	46
5	Construire une plateforme pour la détection	51
5.1	Penser une application pour la détection d'objets	51
5.1.1	Une interface pour le traitement des sources	51
5.1.2	Décrire les sources : images et objets	53
5.2	Annoter sur un GPU : extractorAPI	56
5.2.1	Architecture du système	56
5.2.2	Cahier des charges et analyse des besoins	58
5.2.3	Développement de l'API	60
6	Computer vision et pratiques des chercheurs	67
6.1	<i>Workflow</i> de traitement des sources	67
6.1.1	Chaîne de traitement des sources historiques	67
6.1.2	Échanges et transformation des données	68
6.1.3	Créer des interfaces pour les étapes manuelles	68
6.2	Médiation et documentation	68
6.2.1	Échanges avec les chercheurs	68
6.2.2	Documentation et Wiki : penser le réemploi du code	69
III	Perspectives pour le traitement des sources : vers un outil pour l'édition et la recherche	71
7	Éditer des diagrammes	73
7.1	Édition critique des diagrammes astronomiques	73
7.1.1	Historique	73
7.1.2	Éditer les diagrammes en format numérique	73

TABLE DES MATIÈRES	109
7.2 De l'image aux vecteurs	73
7.2.1 Automatiser la transcription	73
7.2.2 Données d'entraînement et annotations pour la vectorisation	73
8 Regroupement par similarité et <i>clustering</i>	75
8.1 <i>Similarity retrieval</i> et navigation des corpus	75
8.1.1 Principe et méthodes	75
8.1.2 Fouiller des corpus par l'image	75
8.2 Le <i>clustering</i> comme outil pour les chercheurs	75
8.2.1 Sous-section 1	75
8.2.2 Sous-section 2	75
Conclusion	77
Annexes	81
A Prepare custom data for training	81
A.1 What is YOLOv5?	81
A.2 Training dataset	81
A.3 Config file	82
A.4 Notes	82
B Modèles de données EIDA/VHS	83
B.1 Modèle de données initial de l'application VHS	83
B.2 Modèle de données de l'application VHS/EIDA	84
C extractorAPI : cahier des charges	85
C.1 Fonctionnalités	85
C.1.1 <i>Workflow</i>	85
C.1.2 Outils	86
C.1.3 Échange de données	87
C.1.4 Sécurité	87
C.1.5 Interactions	87
D Script YOLOv5 pour lancer la détection d'objet	89
E Annotate images with extractorAPI	95
E.1 Workflow	95
E.2 Settings	95
E.2.1 EiDA	95
E.2.2 extractorAPI	96

E.3	Sending for detection	96
E.3.1	IIIF manifest	96
E.3.2	annotate_wit function	96
E.3.3	Automating the annotation request	97
E.4	Receiving annotations	100
E.5	Indexing annotations	102
Acronymes		103
Table des figures		105
Table des matières		107