

Distant Viewing : Tutoriel 1

Taylor Arnold

Distant Viewing Lab

tarnold2@richmond.edu

Affiches de films et analyse des couleurs



Distant Viewing
Lab



Introduction

Bienvenue

Ces diapositives accompagnent les tutoriels Python que nous avons conçus pour introduire les fondements théoriques et méthodologiques du distant viewing, ainsi qu'un aperçu des outils du Python.

Elles contiennent des notes, références et schémas supplémentaires que nous trouvons utiles lors des ateliers en présentiel. La plupart des notes techniques et du matériel se trouvent directement dans les notebooks.

Pour en savoir plus sur notre travail, rendez-vous sur le site du Distant Viewing Lab à l'adresse <https://distantviewing.org>, ou écrivez-nous directement à tarnold2@richmond.edu.



Un télescope surplombant Paris, en clin d'œil à la méthodologie du distant viewing.



Introduction

Distant Viewing Toolkit (DVT)

En 2017, nous avons commencé à développer un outil en Python, le « Distant Viewing Toolkit ». L'idée était de proposer des outils spécifiquement conçus pour l'analyse en sciences humaines de matériaux visuels.

A screenshot of a GitHub repository page for 'distant-viewing / dvt'. The page shows the README file for the 'dvt' repository. The README content includes:

- Repository name: distant-viewing / dvt
- Navigation links: Code, Issues, Pull requests, Discussions, Actions, Projects, Wiki, Security, Insights, Settings
- Repository path: dvt / README.md
- Commit message: statusrtha: Update README.md
- Preview section: Distant Viewing Toolkit for the Analysis of Visual Culture
- Version information: Python 3.7.9, Code 48.8.0, 48.8.0, 48.8.0, 48.8.0
- Description: The Distant Viewing Toolkit is a Python package that facilitates the computational analysis of still and moving images. The most recent version of the package focuses on providing a minimal set of functions that require only a small set of dependencies. Examples of how to make use of the toolkit are given in the following section.
- Setup information: For more information about setting up the toolkit on your own machine, please see [INSTALL.md](#). More information about the toolkit and project is available on the following pages:
- Links to related projects:
 - Example analysis using aggregated metadata: "[Visual Style in Two Network Era Sitcoms](#)"
 - Theory of the project: "[Distant Viewing: Analyzing Large Visual Corpora](#)"
 - Software Whitepaper: [A Python Package for the Analysis of Visual Culture](#)
- Contact information: If you have any trouble using the toolkit, please open a [GitHub issue](#). If you have additional questions or are interested in collaborating, please contact us at lamost2@richmond.edu and rlhan@richmond.edu.
- Section: Example Usage
- Usage description: To use the toolkit on a still image, we first use the `load_image` function to load the image in Python. Then, we create an annotation model; below we'll use an annotation that detects and identifies faces. Finally, we apply the annotation to the image and save the results.

Les trois projets issus du Distant Viewing Toolkit (DVT).



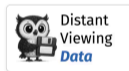
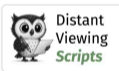
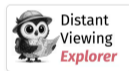
Introduction

Distant Viewing : Outils généraux

Ces dernières années, les outils généralistes ont rattrapé les besoins des chercheurs travaillant avec des données variées, sans qu'il soit nécessaire de construire des modèles d'apprentissage automatique à partir de zéro. Face à ces évolutions, nous avons fait évoluer le toolkit en trois sous-projets :

- **DV Explorer** : des outils graphiques dans le navigateur pour découvrir rapidement le fonctionnement de la vision par ordinateur.
- **DV Scripts** : une collection de scripts Python et de tutoriels montrant comment utiliser des outils généralistes pour des applications en humanités.
- **DV Data** : une collection croissante de matériaux visuels pour la recherche en humanités.

Ce tutoriel fait partie de DV Scripts. Les liens vers les trois projets se trouvent sur la page d'accueil du laboratoire.



Les trois projets issus du Distant Viewing Toolkit (DVT).



Plan

- ▶ 1.1 Configuration
- ▶ 1.2 Jeu de données des affiches de films
- ▶ 1.3 Images numériques
- ▶ 1.4 Distant Viewing : théorie
- ▶ 1.5 Annoter la luminosité de l'image
- ▶ 1.6 Saturation et chroma
- ▶ 1.7 Couleur dominante
- ▶ 1.8 Conclusion et prochaines étapes



1.1 Configuration

Python et Colab

Ce tutoriel s'appuie sur l'environnement Google Colab, qui permet d'utiliser Python directement dans le navigateur, sans installer de logiciel sur sa machine. Colab est gratuit pour les petits projets, et des formules payantes donnent accès à davantage de ressources de calcul.

Nous utiliserons trois modules Python courants pour l'analyse de données : NumPy, Polars et plotnine.

Toutes les bibliothèques utilisées dans le tutoriel sont disponibles sous licence libre. Le code peut aussi être exécuté sur votre propre machine, moyennant une configuration un peu plus poussée.



Logos de Google Colab, Python, Pandas, NumPy et Matplotlib.



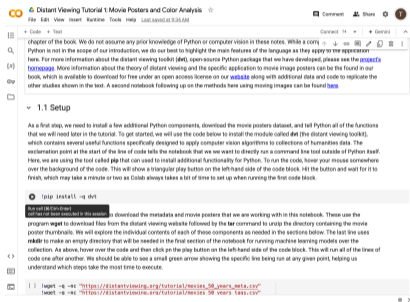
1.1 Configuration

Exécuter le notebook

Lorsqu'on ouvre le notebook du tutoriel dans Colab, une fenêtre similaire à celle de droite s'affiche dans le navigateur. Le notebook est consultable par tous ; pour exécuter le code, il faut se connecter à un compte Google.

Remarque : vous pouvez modifier le code et le texte localement dans votre navigateur. Ces modifications ne seront pas enregistrées pour les autres. N'hésitez donc pas à expérimenter avec le code à tout moment !

Une fois connecté à un compte Google, vous pourrez exécuter les blocs de code (les portions sur fond gris). Pour cela, passez la souris sur le fond gris et cliquez sur le bouton de lecture indiqué par la flèche rouge. Tout au long du tutoriel, il faut exécuter chaque bloc dans l'ordre.



Capture d'écran de l'interface Colab exécutant le code de configuration du DVT.



Plan

- ▶ 1.1 Configuration
- ▶ 1.2 Jeu de données des affiches de films
- ▶ 1.3 Images numériques
- ▶ 1.4 Distant Viewing : théorie
- ▶ 1.5 Annoter la luminosité de l'image
- ▶ 1.6 Saturation et chroma
- ▶ 1.7 Couleur dominante
- ▶ 1.8 Conclusion et prochaines étapes

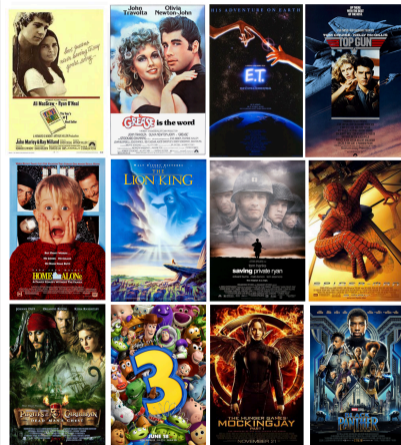


1.2 Jeu de données des affiches de films

Présentation du corpus

Dans ce tutoriel, nous allons explorer une collection d'affiches issues des 100 films les plus rentables de chaque année entre 1970 et 2019. Notre corpus comprend près de 5000 images (nous n'avons pas pu retrouver un petit nombre d'affiches anciennes).

Notre objectif est de comprendre comment la couleur et la composition des affiches (1) ont évolué dans le temps et (2) sont utilisées pour véhiculer (ou détourner) les conventions de genre. Nous voulons mobiliser des techniques computationnelles pour identifier et comprendre des motifs à travers les milliers d'images de notre collection.



Exemples d'affiches de films à travers les décennies et les genres du jeu de données.

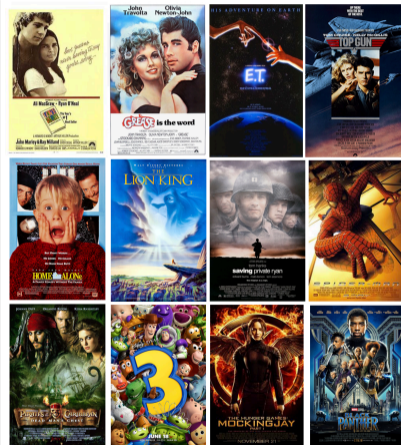


1.2 Jeu de données des affiches de films

Questions de recherche

Avant de nous lancer dans le travail computationnel, prenons un instant pour réfléchir aux questions de recherche qui sous-tendent cette étude :

- Quels sont les objectifs principaux d'une affiche de film ? Ces objectifs ont-ils évolué au fil du temps ?
- Quelles informations les concepteurs d'affiches cherchent-ils à transmettre à leur public ?
- Quelles informations déduisez-vous de la luminosité, de la saturation et des exemples d'affiches présentés à droite ?
- Quelles hypothèses peut-on formuler sur la relation entre la couleur, la composition d'une affiche et le genre du film ?



Exemples d'affiches de films à travers les décennies et les genres du jeu de données.



Plan

- ▶ 1.1 Configuration
- ▶ 1.2 Jeu de données des affiches de films
- ▶ **1.3 Images numériques**
- ▶ 1.4 Distant Viewing : théorie
- ▶ 1.5 Annoter la luminosité de l'image
- ▶ 1.6 Saturation et chroma
- ▶ 1.7 Couleur dominante
- ▶ 1.8 Conclusion et prochaines étapes



1.3 Images numériques

Représentation par pixels

Bien que cette image ait été prise à l'origine sur une pellicule couleur, la version dont nous disposons ici est un scan numérisé, stocké et affiché par un ordinateur.

À l'œil nu, nous identifions sans peine plusieurs éléments : le cheval, l'homme, le ciel, les nuages. Mais comment cette information est-elle comprise par l'ordinateur ?

Pour comprendre comment les images numériques sont stockées, nous allons zoomer dans l'image et examiner une version en basse résolution. Chaque point de l'image combinée est obtenu en définissant la quantité de lumière rouge, verte et bleue nécessaire à la reconstitution de la couleur.



Détail d'une photographie attribuée à Russell Lee, prise en août 1942.

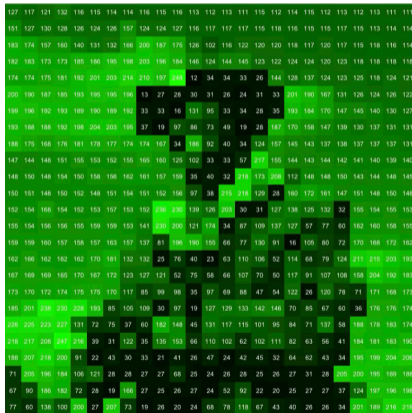


1.3 Images numériques

Lumière RGB

En isolant uniquement les valeurs des pixels verts, on dispose d'assez de place pour voir les nombres eux-mêmes. Comme dans le code Python, ces nombres vont de 0 (pas de vert) à 255 (vert maximal).

Notez qu'il y a beaucoup de lumière verte ici, alors même que rien dans l'image originale n'est ce qu'on appellerait habituellement « vert ».



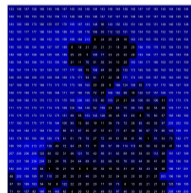
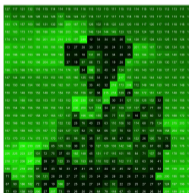
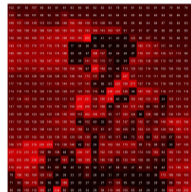
Visualisation des pixels verts.



1.3 Images numériques

Lumière RGB

Le vert se mélange au rouge et au bleu pour produire d'autres couleurs comme le marron, l'orange ou le blanc.



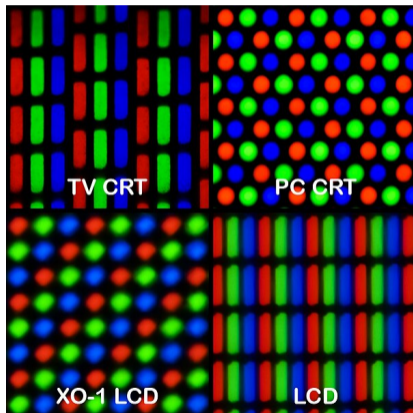
Visualisation des pixels rouges, verts et bleus.



1.3 Images numériques

Lumière RGB

En général, surtout sur les écrans modernes, les pixels sont trop petits pour être discernables individuellement. Mais si l'on zoomait suffisamment, on verrait apparaître les lumières rouge, verte et bleue, comme sur l'image de droite.



Visualisation des pixels RGB sur différents types d'écrans CRT et LCD.



Plan

- ▶ 1.1 Configuration
- ▶ 1.2 Jeu de données des affiches de films
- ▶ 1.3 Images numériques
- ▶ **1.4 Distant Viewing : théorie**
- ▶ 1.5 Annoter la luminosité de l'image
- ▶ 1.6 Saturation et chroma
- ▶ 1.7 Couleur dominante
- ▶ 1.8 Conclusion et prochaines étapes



1.4 Distant Viewing : théorie

Propriétés mimétiques

Notez qu'il est impossible de comprendre que les nombres dans le coin supérieur gauche représentent l'œil d'un cheval tant que ces nombres ne sont pas replacés dans le contexte de l'image entière.

Rien, dans l'image numérique, n'indique explicitement qu'il s'agit d'un cheval. Si nous reconnaissons un cheval, c'est parce que la forme ressemble à celle d'un cheval tel que nous le voyons de nos propres yeux.

Pour reprendre le vocabulaire de la sémiotique, les images numériques (comme leurs équivalents analogiques) produisent du sens par leurs propriétés mimétiques. Elles transmettent de l'information en partageant des similarités avec les choses qu'elles représentent.

30	29	31	35	72	112	132	97	71	58	48
29	25	23	24	26	41	107	107	152	73	61
20	23	24	32	23	17	9	131	103	100	58
25	28	27	19	27	28	21	20	84	104	121
36	34	26	26	28	35	28	34	17	33	68
27	28	28	23	20	25	28	23	30	34	28
27	27	23	28	23	28	23	27	37	41	28
38	34	25	22	20	23	34	33	40	35	38
39	23	24	28	26	31	37	39	43	29	31
20	29	29	33	31	34	31	41	35	34	32
51	34	32	30	43	45	48	40	31	32	38



Zooms successifs sur l'œil d'un cheval, montrant les tableaux de pixels à côté des représentations en niveaux de gris.



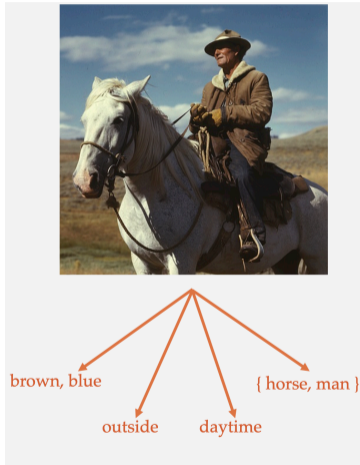
1.4 Distant Viewing : théorie

Annotations sémantiques

Puisque les images numériques n'encodent pas explicitement leurs éléments sémantiques, nous devons construire des représentations des informations qui nous intéressent, dans le cadre même du processus analytique.

Par exemple, à droite, on voit plusieurs façons de résumer l'information contenue dans notre image à travers des annotations structurées.

Aucune annotation ne peut représenter parfaitement l'image. On perd toujours de l'information présente dans l'image, et on fait des hypothèses (et parfois des erreurs) dans la manière dont on construit ces annotations.



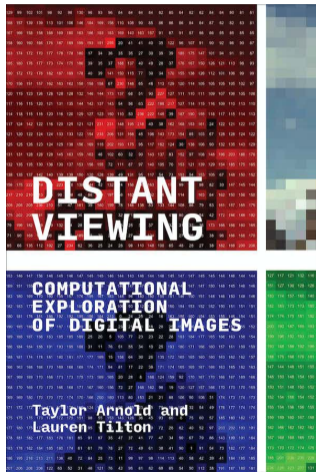
Exemples d'annotations sémantiques.



Principes du Distant Viewing

La théorie du distant viewing combine ces observations avec des travaux en sémiotique et en culture visuelle pour proposer une théorie de l'exploration computationnelle des images numériques par la vision par ordinateur, et expliquer pourquoi cette approche est nécessaire.

- La nature mimétique des images impose que l'analyse computationnelle des images numériques commence par l'extraction de métadonnées sémantiques (« annotations ») via la vision par ordinateur.
- Le processus de construction d'annotations est une forme de regard médiatisée par le calcul.
- L'application de la vision par ordinateur n'est pas un processus neutre. Elle engage des manières de voir socialement situées, qui sont inscrites dans les algorithmes et dans les façons dont ils sont utilisés.



Couverture de Distant Viewing: Computational Exploration of Digital Images (MIT Press, 2023).



Plan

- ▶ 1.1 Configuration
- ▶ 1.2 Jeu de données des affiches de films
- ▶ 1.3 Images numériques
- ▶ 1.4 Distant Viewing : théorie
- ▶ **1.5 Annoter la luminosité de l'image**
- ▶ 1.6 Saturation et chroma
- ▶ 1.7 Couleur dominante
- ▶ 1.8 Conclusion et prochaines étapes



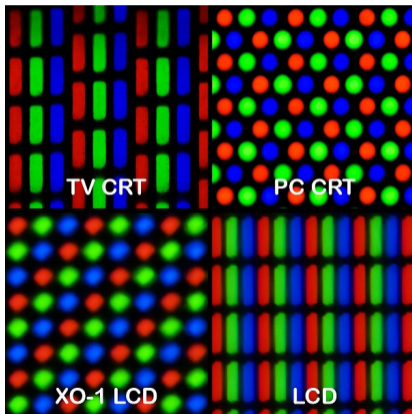
1.5 Annoter la luminosité de l'image

Mesurer la luminosité

Commençons par une annotation relativement simple, qui mesure la luminosité de chaque affiche.

- Comment la luminosité des affiches peut-elle évoluer dans le temps ? Quels facteurs pourraient expliquer ces changements ?
- Comment la luminosité peut-elle se rattacher au genre d'un film ? Quels genres pensez-vous être les plus lumineux ? Lesquels seraient les plus sombres ?

Comment mesurer la luminosité d'une image ? Rappelons que les intensités des pixels indiquent la quantité de lumière rouge, verte et bleue nécessaire pour produire une couleur donnée. En faisant la moyenne des valeurs de pixels sur toute l'image, on obtient une indication de la quantité totale de lumière utilisée, et donc une manière de mesurer sa luminosité.



Visualisation des pixels RGB sur différents types d'écrans CRT et LCD.



Plan

- ▶ 1.1 Configuration
- ▶ 1.2 Jeu de données des affiches de films
- ▶ 1.3 Images numériques
- ▶ 1.4 Distant Viewing : théorie
- ▶ 1.5 Annoter la luminosité de l'image
- ▶ **1.6 Saturation et chroma**
- ▶ 1.7 Couleur dominante
- ▶ 1.8 Conclusion et prochaines étapes



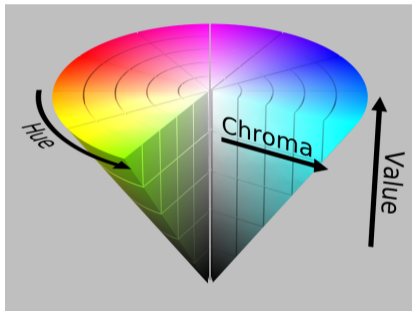
1.6 Saturation et chroma

Représentations alternatives de la couleur

À droite, une visualisation de la couleur sous la forme teinte-saturation-valeur (HSV), une alternative à la représentation RGB que nous avons utilisée jusqu'ici. La saturation vaut 0 au centre du cylindre (nuances de gris) et 1 sur sa surface extérieure. La valeur est une autre mesure de la luminosité.

Le chroma est une représentation alternative de la saturation. On peut le visualiser géométriquement en pinçant la base du cylindre HSV pour former un cône.

Nous travaillons avec le chroma dans notre analyse parce que nous pensons qu'il rend mieux compte de la richesse d'une couleur. Le choix entre ces deux mesures illustre bien le type de décisions subjectives, parfois subtiles, qui influencent les applications du distant viewing et qu'il faut sans cesse expliciter.



Visualisations des espaces colorimétriques sous forme de cylindre HSV et de cône Chroma.



Plan

- ▶ 1.1 Configuration
- ▶ 1.2 Jeu de données des affiches de films
- ▶ 1.3 Images numériques
- ▶ 1.4 Distant Viewing : théorie
- ▶ 1.5 Annoter la luminosité de l'image
- ▶ 1.6 Saturation et chroma
- ▶ **1.7 Couleur dominante**
- ▶ 1.8 Conclusion et prochaines étapes



1.7 Couleur dominante

Teinte et genre

La teinte (hue) d'une couleur correspond au nom courant qu'on lui donnerait usuellement : rouge, vert, violet, etc. Nous représenterons la teinte par un nombre entre 0 et 1, correspondant aux positions sur la roue chromatique de droite.

Notez que la teinte est une mesure circulaire : les valeurs proches de 0 et celles proches de 1 correspondent à des nuances de rouge très similaires.

En regardant les affiches et en réfléchissant à la signification culturelle de la couleur, prenez un instant pour vous interroger avant de découvrir les résultats agrégés :

- Quels liens vous attendez-vous à voir entre des couleurs précises et certains genres ?
- Pensez-vous que certaines couleurs soient des indicateurs fréquents d'un genre particulier ?
- Quelles couleurs vous semblent les plus utilisées sur les affiches de films ?



Une roue chromatique standard, avec les valeurs de teinte allant de 0 à 1.



Plan

- ▶ 1.1 Configuration
- ▶ 1.2 Jeu de données des affiches de films
- ▶ 1.3 Images numériques
- ▶ 1.4 Distant Viewing : théorie
- ▶ 1.5 Annoter la luminosité de l'image
- ▶ 1.6 Saturation et chroma
- ▶ 1.7 Couleur dominante
- ▶ 1.8 Conclusion et prochaines étapes



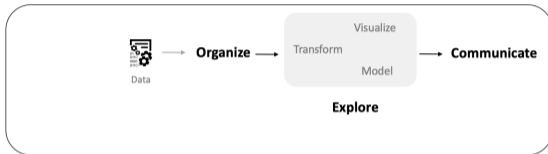
1.8 Conclusion et prochaines étapes

Conclusion

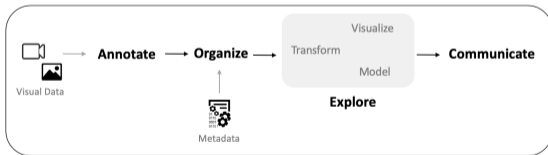
En prenant un peu de recul par rapport aux méthodes spécifiques, on voit que le distant viewing décrit une manière particulière de travailler les données quand celles-ci proviennent d'un fonds visuel.

Plus précisément, il faut créer des annotations structurées qui représentent (imparfaitement) les traits qui nous intéressent, avant même de passer à la visualisation, au nettoyage des données ou à toute autre forme de modélisation.

Ce schéma se poursuivra dans le second tutoriel.



Data Science Pipeline (Structured Data)



Pipeline for Distant Viewing

Pipeline classique de la science des données (en haut) comparé au pipeline du distant viewing.



1.8 Conclusion et prochaines étapes

Pour aller plus loin

Plusieurs directions s'ouvrent à vous après ces notes :

- Pour voir comment mener à bien l'analyse présentée ici, consultez le chapitre 3 de Distant Viewing.
- Si vous débutez en programmation, suivez les liens dans le notebook pour découvrir Python.
- Pour en savoir plus sur les algorithmes de vision par ordinateur et le travail sur les images animées, consultez les tutoriels 2 et 3.

Merci d'avoir pris le temps de nous suivre. N'hésitez pas à nous contacter pour toute question ou tout retour !