Distant Viewing Toolkit : Tutoriel 3

Taylor Arnold
Distant Viewing Lab
tarnold2@richmond.edu

Sitcoms de l'ère des grands réseaux et style visuel

Distant Viewing Lab

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 Distant Viewing Toolkit.

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.

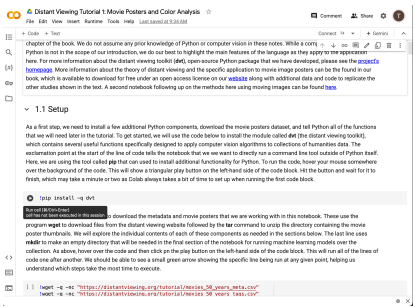
Pour en savoir plus sur notre travail, rendez-vous sur le site du Distant Viewing Lab à l'adresse https://distantviewing.org, ou écrivez-nous directement à tarnold2@richmond.edu et ltilton@richmond.edu.

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

Plan

Python et Colab

Comme pour le premier tutoriel, le deuxième 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.



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

Modules Python

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

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.

Plan

Présentation du corpus

Notre analyse portera sur deux sitcoms américains populaires diffusés du milieu des années 1960 au début des années 1970. Un bref résumé de chacun montre pourquoi ils sont souvent comparés et opposés :

Bewitched : la série tourne autour du mariage de Darrin Stevens, homme ordinaire, et de Samantha, une sorcière dotée de pouvoirs surnaturels. Samantha fait de son mieux pour mener une vie américaine « normale », mais des complications venues de son monde magique viennent régulièrement contrarier ses efforts.

I Dream of Jeannie : la série met en scène un génie féminin âgé de 2000 ans, surnommée Jeannie, découverte par l'astronaute Tony Nelson, qu'elle appelle son « maître ». Un ressort narratif récurrent : Tony tente d'accomplir son travail tout en cachant et en gérant les facéties magiques de Jeannie.

Bewitched (ABC; 1964–1972) *I Dream of Jeannie* (NBC; 1965–1970)

Extraction des données

Pour travailler de manière computationnelle sur ces séries, nous avons d'abord acheté des coffrets DVD contenant l'intégralité des 393 épisodes des deux séries. Nous avons ensuite extrait les fichiers vidéo des DVD, une opération désormais autorisée aux États-Unis dans le cadre du DMCA §1201 du code de la propriété intellectuelle américain.

La collection complète représente plus de 150 heures de matériau, soit plus de 13 millions de trames individuelles.

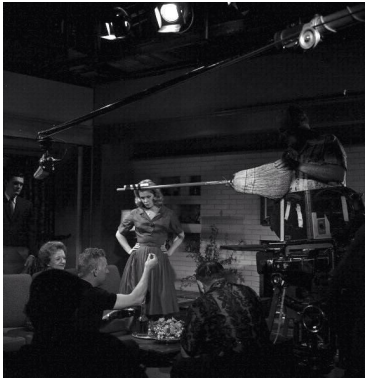
Coffrets DVD intégraux de Bewitched et I Dream of Jeannie.

Questions de recherche

Avant de passer à l'analyse computationnelle, prenons un instant pour considérer les questions de recherche qui ont motivé notre travail sur ces séries :

- Comment le cadrage d'un plan participe-t-il au rythme d'un épisode, et comment ces éléments varient-ils selon les séries, les époques ou d'autres caractéristiques ?
- Quelles formes visuelles et sonores des séries établissent la centralité et le rôle de chaque personnage, et comment ?
- Quelles formes visuelles et sonores des séries établissent les relations et l'intimité entre des paires de personnages, et comment ?

Réfléchissez à vos propres hypothèses sur ces questions. Pouvez-vous ajouter d'autres questions du même ordre auxquelles cette collection pourrait permettre de répondre ?



Tournage sur le plateau de Bewitched (1965).

Plan

Contexte qualitatif

Lorsqu'on mène des analyses computationnelles sur des matériaux visuels et audiovisuels, il est important de revenir fréquemment à des exemples précis pour s'assurer que notre analyse à grande échelle reste connectée à l'expérience humaine du visionnage.

Prenez le temps de regarder l'extrait de 45 secondes de Bewitched.

Après avoir visionné ce court extrait, vos idées sur les questions de recherche ont-elles évolué ? Souhaiteriez-vous ajouter de nouvelles questions ?

Bobines vidéo 2 pouces des UCLA Film & TV Archive.

Plan

Les trames comme images

Même si les fichiers vidéo sont stockés dans des formats très compressés, la meilleure manière de se représenter un fichier vidéo numérique est souvent comme une séquence de trames individuelles (accompagnée d'une bande son), chaque trame étant une image de même taille.

À titre d'exemple, l'image de droite a été construite en prenant une trame sur dix dans un extrait de 40 secondes au début de l'épisode « The Magic Cabin » de Bewitched (saison 2, épisode 16).

Séquence de trames individuelles extraites du fichier vidéo.

Plan

Identifier les ruptures manuellement

Comme l'indique le code, repérer les ruptures entre les plans est une étape fréquente et importante de l'analyse computationnelle des images animées.

En regardant les trames d'exemple à droite, parvenez-vous à identifier les ruptures de plan ? Notez qu'utiliser la luminosité comme dans l'exemple des notes serait délicat ici : au moins l'un des plans ne fait pas beaucoup varier la luminosité globale, et la séquence se termine par un fondu au noir.

Trames d'exemple pour repérer visuellement les ruptures de plan.

Sortie de l'algorithme

Voici les plans détectés par l'algorithme TransNetV2. Correspondent-ils aux ruptures que vous aviez repérées ?

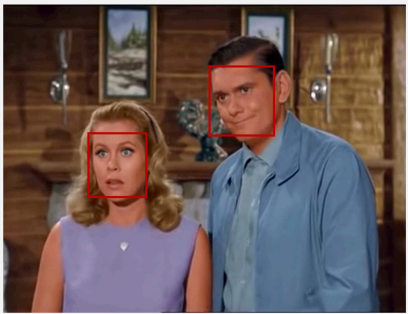
Plans détectés 1 à 6. Référence : Li, Yanghao, et al. (2021) « Benchmarking Detection Transfer Learning with Vision Transformers ».

Plan

Montage et mise en scène

Comme vous l'avez sans doute vu dans le premier tutoriel, repérer la position des visages dans une image fournit souvent une annotation utile pour de nombreuses applications.

Pour les images animées, la position des visages des personnages constitue souvent un point de départ pour comprendre le montage et la mise en scène des plans. C'est exactement le type de relations qui rejoignent nos questions de recherche.

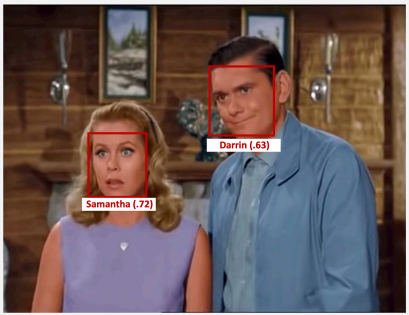


Introduction à la détection de visages dans les images animées.

Identifier les personnages

Comme décrit dans le notebook Python, lorsqu'on travaille avec des données cinématographiques ou télévisuelles, on cherche souvent à identifier quel personnage est associé à chacun des visages d'une trame.

En appliquant la technique vue dans l'exemple du notebook, voici les noms de personnages estimés et les scores de similarité pour les deux personnages.



Scores de reconnaissance faciale : Samantha (.72) et Darrin (.63). Référence : Cao, Qiong, et al. (2018). « VGGface2: A dataset for recognising faces across pose and age ».

Plan

Passage à l'échelle du corpus

Une fois qu'on comprend comment faire tourner les annotations sur un court extrait du corpus, l'étape suivante consiste à exécuter les algorithmes sur l'ensemble de la collection. Travailler sur de grands ensembles d'images animées peut demander beaucoup de ressources de calcul et/ou de temps : nous avons donc mis en cache les résultats pour que le notebook puisse les lire directement.

Une étape que nous escamotons ici est la méthode de sélection des seuils et des points de coupure lors du filtrage des données. Par exemple : à partir de quelle probabilité considère-t-on qu'un visage a vraiment été trouvé par l'algorithme ? À quel point un visage doit-il être proche d'un portrait pour qu'on accepte l'identification d'un personnage ? Si on fait tourner l'algorithme sur plusieurs trames d'un même plan, comment les agréger ? Répondre à ces questions exige de retourner abondamment aux matériaux originaux et de déterminer manuellement quels seuils fonctionnent le mieux pour notre application.

Une famille devant un téléviseur des années 1950.

Plan

Répondre aux questions de recherche

Dans cette section du notebook, nous voyons comment utiliser les données extraites pour aborder certaines des questions de recherche que nous avons posées plus tôt.

Notez que nous n'en présentons ici qu'un petit échantillon par rapport aux analyses contenues dans le chapitre du livre. Nous invitons les lecteurs intéressés à explorer les analyses plus longues qui y figurent.

Pouvez-vous imaginer des façons d'étendre cette analyse à l'aide de techniques plus récentes (LLM ?) ou en intégrant d'autres séries télévisées au corpus ?

Plateau 2 du General Service Studio où I Love Lucy a été tournée, avec le dispositif à trois caméras utilisé sur scène. American Cinematographer.

Plan

Pour aller plus loin

Plusieurs directions s'ouvrent à vous après ce deuxième tutoriel :

- Pour voir comment compléter l'analyse de ce deuxième tutoriel, consultez le chapitre 5 de Distant Viewing.
- Si vous débutez en programmation, suivez les liens dans le notebook pour découvrir Python.
- Pour ceux qui ont sauté le premier tutoriel, il peut être utile d'y revenir maintenant afin de mieux comprendre le fonctionnement des images numériques et ses implications pour les fondements théoriques du distant viewing.

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