



Cas d'étude | Image Seeker

07.04.2023

Réalisé par

Frédéric Chen

Thomas Danguilhen

Céline Goncalves

Emil Răducanu

Arthur Sarmini Det Satouf

Encadré par

Youcef Sklab

Lylia Alouache

Souhila Arib

Table des matières

I - Introduction.....	2
II - Moteur de recherche d'images Clip.....	3
1. Présentation du jeu de données.....	3
2. Présentation du modèle CLIP.....	3
3. Présentation de l'implémentation.....	5
4. Résultat.....	7
III - Conclusion.....	8
IV - Bibliographie.....	9

I - Introduction

L'intelligence artificielle (IA) est un domaine en constante évolution, et l'une des avancées les plus remarquables de ces dernières années est l'apprentissage multimodal. Ce domaine de recherche se concentre sur la fusion de différentes sources de données, notamment le traitement de texte et les images, en utilisant des techniques d'apprentissage automatique.

Le modèle CLIP d'OpenAI est au centre des travaux dans ce domaine, grâce à sa capacité à intégrer des contenus multimédias dans un même espace latent en utilisant une technique d'apprentissage contrastive. Les modèles multimodaux comme CLIP ont permis des avancées significatives dans la classification d'images, le transfert learning, la génération de données synthétiques et la recherche sémantique.

Dans le cadre de ce rapport, nous nous intéresserons à cette dernière application et verrons comment elle peut être utilisée dans un workflow de vision par ordinateur. Plus précisément, nous nous concentrerons sur le développement d'une application Streamlit permettant de faire de la recherche d'images à partir de requêtes textuelles en utilisant des outils tels que Pinecone et le modèle CLIP d'OpenAI.



<https://github.com/Danguilhen/clip-image-search-engine.git>



<https://www.kaggle.com/code/dann12/notebook104d612fb3>

II - Moteur de recherche d'images Clip

1. Présentation du jeu de données

L'ensemble du jeu de données provient de ImageNet qui l'un des plus grands ensembles de données d'images annotées disponible publiquement. Elle contient plus de 14 millions d'images appartenant à plus de 20 000 catégories d'objets différents. Pour ce projet, nous avons seulement utilisé une partie d'ImageNet, qui est ImageNet 2012, qui est un ensemble de données d'image contenant 1 000 images. Ces images appartiennent à tous types de catégories (animaux, objets, personnes etc.)

2. Présentation du modèle CLIP

Le modèle CLIP (Contrastive Language-Image Pre-Training) est un modèle open-source de réseau de neurones profonds créé par OpenAI qui permet de traiter des images et du texte simultanément. Il s'agit d'un modèle multimodal, capable de comprendre la relation sémantique entre les mots et les images. Le modèle a été entraîné à partir d'un large ensemble de données multimodales, en utilisant une technique d'apprentissage automatique dite "contrastive" qui permet de placer les images et les textes dans un même espace latent.

L'objectif du modèle CLIP est de permettre aux machines de comprendre la signification des images et du langage naturel de la même manière que les humains. Grâce à sa capacité à lier des images à des descriptions textuelles, le modèle peut être utilisé pour une variété de tâches, telles que la classification d'images, la génération de légendes pour les images, la recherche d'images par le texte, etc.

Une autre caractéristique importante du modèle CLIP est qu'il utilise l'apprentissage zero-shot, ce qui signifie qu'il peut généraliser sur des étiquettes non vues sans avoir été spécifiquement entraîné pour les classer. Contrairement aux modèles traditionnels d'ImageNet qui sont entraînés à reconnaître des classes spécifiques, le modèle CLIP est libre de cette limitation.

Le modèle CLIP a été entraîné sur un ensemble de données massif appelé ImageNet, qui contient plus de 1,28 million d'images annotées avec des catégories sémantiques.

Le modèle CLIP est une avancée majeure dans le domaine de l'IA multimodale, et il a permis de nombreuses applications dans divers domaines tels que la reconnaissance d'objets, la création de modèles de langage naturel, la traduction automatique, la synthèse d'images, etc.

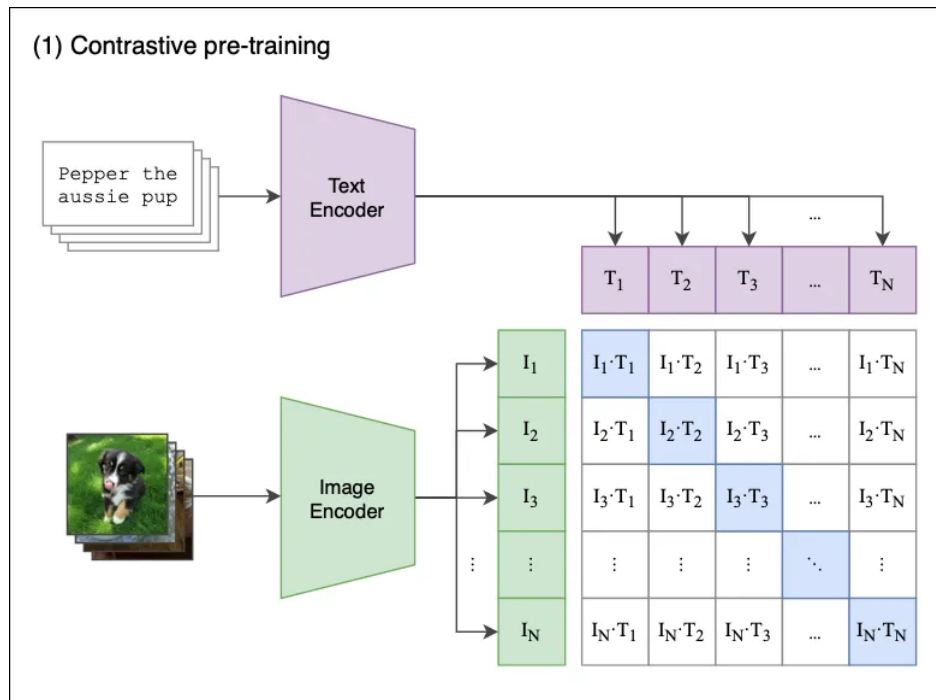


Figure 1: Contrastive Pre-training step of **CLIP**

3. Présentation de l'implémentation

Solutions techniques

- Pre-commit
- Sphinx
- PEP 008
- PEP 257
- Github workflow
- Readthedocs
- Pinecone
- Streamlit
- CLIP (clip-vit-base-patch32, Hugging Face)
- PyInstaller

Pre-commit est un outil qui vous permet d'automatiser les contrôles sur votre code avant de faire un commit. Cela aide à garantir que votre code respecte les normes de codage et les meilleures pratiques, et peut détecter les erreurs courantes avant qu'elles ne deviennent de plus gros problèmes.


Sphinx est un générateur de documentation populaire pour les projets Python. Il vous permet d'écrire la documentation dans un langage de balisage simple, et il générera une documentation de qualité professionnelle dans divers formats, tels que HTML, PDF, etc.

PEP 008 est un guide de style pour le code Python. Il fournit des recommandations pour écrire du code lisible et maintenable, et il est largement suivi par la communauté Python. En respectant le PEP 008, vous pouvez vous assurer que votre code est cohérent et facile à comprendre.

PEP 257 est un guide pour écrire des docstrings en Python. Il fournit des recommandations sur les informations à inclure dans les docstrings et sur la manière de les formater. Cela aide à garantir que la documentation de votre code est cohérente et fournit toutes les informations dont les utilisateurs ont besoin.

Github est une plateforme populaire pour héberger et gérer des référentiels de code. Le workflow Github fait référence au processus d'utilisation de Github pour gérer le code de votre projet, y compris la création de problèmes, l'attribution de tâches et la gestion des demandes de tirage.

Readthedocs est une plateforme qui vous permet d'héberger et de construire la documentation de votre projet. Il s'intègre à Github, de sorte que chaque fois que vous



apportez des modifications à votre code, il reconstruira automatiquement votre documentation. Cela facilite la mise à jour et l'accessibilité de votre documentation pour les utilisateurs.

Pinecone est une plateforme d'indexation de vecteurs utilisée avec le modèle CLIP. Elle permet de stocker et d'indexer les vecteurs de représentation d'images et de texte précalculés par le modèle CLIP, afin de faciliter et d'accélérer la recherche d'images à l'aide de requêtes textuelles. Cela permet d'obtenir des résultats de recherche rapides et précis pour une expérience utilisateur satisfaisante.

Streamlit est une bibliothèque open-source qui permet de créer facilement des applications web interactives pour l'analyse de données et l'apprentissage automatique. Elle offre une grande variété de fonctionnalités pour le développement d'applications web, tout en étant simple et rapide à utiliser.

PyInstaller est une bibliothèque open-source qui permet de créer des exécutables autonomes pour des programmes Python, sans avoir besoin d'installer Python ou les bibliothèques de dépendance. Elle est simple à utiliser et prend en charge différents systèmes d'exploitation. PyInstaller est utile pour les développeurs qui souhaitent distribuer leurs applications facilement et protéger leur code source.

Application Streamlit

Notre application Streamlit permet de rechercher des images en fonction d'une requête texte en utilisant le modèle CLIP d'OpenAI.

Le programme commence par importer les bibliothèques nécessaires, y compris `ipoly` pour charger les données d'images, la bibliothèque `Numpy` pour la manipulation de matrices, la bibliothèque `Pinecone` pour la recherche d'images, la bibliothèque `Tensorflow` pour le calcul de l'embedding, et la bibliothèque `Streamlit` pour l'interface utilisateur.

Ensuite, la fonction `"text_embedding"` est définie pour créer l'embedding d'un texte donné en utilisant le modèle CLIP. La fonction `"image_embedding"` est également définie pour créer l'embedding d'un lot d'images en utilisant le même modèle.

La fonction `"connect_pinecone"` est utilisée pour se connecter à Pinecone, une plateforme de recherche d'images, et créer un index pour les images chargées.

La fonction `"get_images"` est définie pour effectuer la recherche d'images à partir d'une requête texte donnée. Cette fonction crée un embedding pour le texte donné, effectue une requête à l'index créé, et renvoie les images les plus similaires.

Nous avons, par ailleurs, implémenté l'entraînement de CLIP sur une base de données comprenant plus de 14 000 images représentant 100 sports différents.

Enfin, l'interface utilisateur Streamlit est définie avec trois champs de saisie pour l'API token de Picone, l'environnement cloud et la requête texte, et un bouton pour lancer la recherche. Si les images les plus similaires sont trouvées, elles sont affichées dans l'interface utilisateur. Sinon, un message d'erreur est affiché.

4. Résultat

Image seeker

Enter your API token

.....



Enter your cloud environment ex: eu-west1-gcp

eu-west1-gcp

Enter a small text

a boat on the river

Validate



Figure 2 : visuel de notre application

III - Conclusion

En conclusion, notre solution permet de créer une application intéressante et utile à l'aide de Streamlit, qui utilise le puissant modèle CLIP pour trouver des images en fonction des requêtes textuelles saisies par l'utilisateur. Cependant, l'utilisation de modèles d'apprentissage profond tels que CLIP nécessite des ressources informatiques importantes, ce qui peut limiter l'accès pour certains utilisateurs. De plus, la configuration et l'authentification du compte Pinecone peuvent poser des difficultés potentielles pour certains utilisateurs, tout comme la récupération des images à partir des sources. Il est également important de noter les limites et les biais possibles de ces modèles, ainsi que les implications éthiques potentielles de leur utilisation dans des applications critiques.

Bien que nous ayons rencontré quelques difficultés de compréhension au départ, nous avons réussi à répondre aux attentes du sujet et en les surpassant en ré-entraînant le modèle CLIP. En effet, cela a impliqué de comprendre les exigences du projet ainsi que le fonctionnement de nouveaux outils tels que CLIP et Pinecone.

Finalement, ce projet nous a permis d'approfondir nos connaissances en deep learning et de découvrir un nouveau modèle à l'état de l'art, tout en nous donnant une base pour développer des applications plus complexes à l'aide de bibliothèques open source disponibles.

IV - Bibliographie

[2] <https://docs.pinecone.io/docs/quickstart>

[3] <https://docs.pinecone.io/docs/image-similarity-search>

[4] <https://medium.com/voxel51/finding-images-with-words-92b078314ed1>

[5] <https://github.com/openai/CLIP>

[6] <https://streamlit.io/>