

Hochschule Düsseldorf
Fachbereich Medien
Data Science, AI & IS

Hochschule Düsseldorf
University of Applied Sciences

Fachbereich Medien
Faculty of Media



Dokumentation

Thema: NLP-basiertes Imageretrieval
Modul: Advance in AI

von: Anton Zhevnov 882037

GitHub Link: [Click](#)

Prof: Dr. Dennis Müller
Abgabe: 19.02.2024 23:59

Ablauf von der Dokumentation

In dieser Dokumentation werden verschiedene Themen behandelt, die mit dem Projekt "NLP-basiertes Imageretrieval" verbunden sind. Durch Links können Sie zu den einzelnen Abschnitten navigieren.

- 01. Motivation
- 02. Code Aufbau
- 03. Anleitung
 - a. Vorbereitung
 - b. Ausführung
- 04. Technische Details
 - a. Clip
 - b. Annoy-Datenbank
- 05. Testergebnisse
- 06. Schwierigkeiten
- 07. Nutzung des Projekts
- 08. Quellen

Motivation

Im letzten Semester meines Studiengangs Daisy, im Fachgebiet Intelligente sichere Systeme und Big Data Engineering, haben wir darüber gesprochen, wie man große Mengen von Daten sortiert und auf diese zugreifen kann, wobei wir uns insbesondere mit Bildern beschäftigt haben. Unser Ziel war es, drei verschiedene Methoden zu entwickeln, um ein passendes Bild zu finden. Meine gewählten Methoden waren RGB und Hash. Dieses Projekt hat mir viel Spaß gemacht und das Ergebnis war sehr beeindruckend und interessant. Dadurch wurde mein Interesse geweckt, im Fach Advances in Intelligent Systems das Thema NLP-basiertes Imageretrieval zu wählen, da es sich nahtlos an mein Thema des letzten Semesters anschließt und es auf ein neues Level hebt.

Aber was genau ist NLP-basiertes Imageretrieval?¹

NLP-basiertes Imageretrieval bezieht sich auf den Einsatz von Natural Language Processing (NLP)-Techniken, um die Bildsuche und -abfrage zu verbessern. Normalerweise werden bei der Bildsuche traditionelle Methoden wie visuelle Merkmalsextraktion und -ähnlichkeit verwendet. Allerdings können diese Methoden Schwierigkeiten haben, semantische Bedeutungen oder Konzepte in Bildern zu erfassen, insbesondere wenn es darum geht, komplexe oder abstrakte Konzepte zu verstehen.

NLP-basiertes Imageretrieval versucht, diese Lücke zu überbrücken, indem es natürliche Sprache, also menschliche Sprache, verwendet, um die Anfragen für die Bildsuche zu formulieren oder um Bilder zu beschreiben. Dafür werden NLP-Techniken wie Textanalyse, semantische Analyse und Sprachmodellierung eingesetzt, um die Bedeutung von Textbeschreibungen oder Suchanfragen zu erfassen und mit den Bildinhalten abzugleichen. In meinem ersten Ansatz wollte ich die Methode X aus dem letzten Semester verbessern, indem ich sie mit Sprachverarbeitung kombiniere. Jedoch wäre dies zeitaufwendig und würde vom angestrebten Ergebnis ablenken. Daher bin ich zurück zur Aufgabenstellung gegangen und habe eine Methode gewählt, die auch von Professor Denis Müller empfohlen wurde: Contrastive Language-Image Pretraining (CLIP), zusammen mit der Datenbank Annoy. Die Funktionsweise von CLIP und der Annoy-Datenbank werde ich im technischen Teil genauer erläutern.

¹ <https://deeplobe.ai/text-based-image-retrieval-using-deep-learning/>

Code Aufbau

Ich habe versucht, den Code benutzerfreundlich zu gestalten, damit auch Nutzer, die wenig Erfahrung mit PCs oder Programmierung haben, ihn ohne Probleme nutzen können. Alle Eingaben können über die Konsole erfolgen. Ein einziger Eingriff im Code wäre nur beim Erstellen einer großen Datenbank mit mehr als 1000 Bildern erforderlich. Der Benutzer sollte dann in Zeile 152 den Wert "2" in "annoy_db.build(2)" auf die empfohlene Größe setzen. Dies wird explizit noch einmal in der Anleitung des Codes erwähnt. Hier werde ich alle meine Funktionen kurz erläutern und erklären. Ich werde versuchen, sie sehr leicht und genau zu erklären, damit jeder diese Funktionen verstehen kann. Ein paar Worte zum Aufbau des Codes: Die Funktionen sind auf Englisch, möglicherweise gibt es einige Funktionen, die auf Deutsch geblieben sind. Diese könnte ich einfach übersehen haben, aber das sollte nicht der Fall sein.

Ich beginne mit den Bibliotheken, die ich für die Erstellung meines Codes verwendet habe. Die "os", "clip" und "torch"-Bibliotheken werden für das encodieren von Bildern und Texten verwendet. "Annoy" und "sqlite3" sind die Datenbanken, die ich in meinem System verwendet habe. Die "time"-Bibliothek wurde für Zeitmessungen für die Bildersuche verwendet, da die Suchzeit bei meiner ersten Bibliothek, nicht Annoy, sehr lang war. "matplotlib.pyplot" wurde verwendet, um die ausgewählten Bilder anzuzeigen, und "shutil" zum Kopieren dieser Bilder in einen separaten Ordner im gewünschten Pfad.

Nun werde ich die Funktionen erklären und wie sie miteinander verbunden sind. Die Funktion "def show_image(image_path):" ist dafür verantwortlich, die Bilder anzuzeigen, wenn der Benutzer keine Kopie der ausgewählten Bilder in einem Ordner wünscht. Hier wird einfach der Pfad, der von der SQLite-Datenbank ausgegeben wird, durch "Image.open" gelesen, dann wird vom Benutzer Speicherort angegeben und im Panel bzw. Plot angezeigt.

Eine weitere wichtige Funktion ist "def find_matching_image_items(...)". Zuerst wird der Text in einen Vektor umgewandelt. Anschließend werden der Textvektor und der Bildervektoren ausgewählt und mit der Methode der Annoy-Bibliothek nach den ähnlichsten Nachbarn gesucht. Die Suchmethode, die die Annoy-Bibliothek verwendet, ist "search_k". Das Ganze wird auch zeitlich gemessen, damit ich bzw. der Nutzer einen Überblick darüber haben kann, wie lange es dauert, die ähnlichsten Nachbarn des Textvektors zum Bildervektoren zu finden und zurückzugeben. Derzeitiger Wert für die Berechnung liegt bei 100, "search_k_value=100", wird jedoch noch getestet. Aktuelle Informationen werden in der Zusammenfassung bereitgestellt.

Die Funktion "def image_path_id" dient dazu, die Pfade der Bilder zu speichern. Da die Annoy-Bibliothek nur IDs und Vektoren speichern kann, wird hier SQLite für die Speicherung der Pfade verwendet. Das bedeutet, dass die Funktion die ID des Bildes nimmt, dieselbe ID in der Annoy-Datenbank hat und die ID zusammen mit dem Pfad speichert. Am Ende gibt SQLite den Pfad für die ID aus, die von "def find_matching_image_items" gefunden oder vorgeschlagen wurde.

Die Funktion "new_images_data" erstellt einen neuen Datensatz bzw. encodiert die Bilder, die vom Nutzer einmal eingegeben werden, in Form von Pfaden zu einem Ordner mit Bildern. Der Nutzer muss diese Funktion nicht unbedingt nutzen. Er kann entscheiden, ob er eine neue Datenbank mit eigenen Bildern erstellen möchte oder ob er bereits vorbereitete Datenbanken nutzen möchte. Hier kann der Nutzer seine ersten eigenen Eingaben machen. Bei der Eingabe von "Yes" kann der Nutzer eigene Bilder encodieren und nach den beschriebenen Bildern von eigenen Bildern suchen. Bei der Eingabe von "No" wird nach bereits erstellten Datenbanken gefragt. Wenn der Nutzer sich jedoch nicht sicher ist, kann er "Info" eingeben und erhält eine Erklärung, bevor er seine Entscheidung trifft.

Die nächste Funktion ist "def ask_save_to_folder". Hier kann der Nutzer am Ende entscheiden, ob er die Bilder in separaten Ordnern speichern möchte (bei "Yes") oder ob er die Bilder nur betrachten möchte (bei "No").

Die nächste Funktion "def create_and_copy_images_to_folder(paths)" basiert auf der Antwort "Yes" in der Funktion "def ask_save_to_folder()". Hier wird der Nutzer zunächst nach dem Speicherort für die Ordner gefragt und anschließend nach dem Namen der Ordner. So kann der Nutzer die Bilder sehr flexibel speichern und gleichzeitig den Überblick behalten, da er die volle Entscheidung hat. Wenn das Speichern erfolgreich war, erhält der Nutzer auch eine Benachrichtigung, dass alles "Top" gelaufen ist. Falls nicht, erhält er eine entsprechende Fehlermeldung. Diese kann entweder auf der Richtigkeit des Pfads oder auf dem Namen der Ordner basieren, der zum Beispiel doppelt vorkommt.

Nun möchte ich den Ablauf des Codes erklären. Es gibt verschiedene Versionen von Clip-Modellen, die alle auf der Clip-Website aufgelistet sind. Chat GPT hat mir auf die Frage "Welches Clip-Modell ist am besten für große Bildmengen geeignet?" das Modell "ViT-L/14@336px" vorgeschlagen, da es sich am besten geeignet erwiesen hat. Weitere Modelle, die ich getestet habe, waren: "ConvNext-XXLarge", "ViT-H/14" und "ConvNext-Large@320".

Hier wird der Benutzer nach dem Pfad gefragt, wenn er eigene Bilder encodieren und einstellen möchte. Dabei werden Anführungszeichen("") ignoriert, damit der Benutzer einfach den Pfad aus einem Ordner kopieren und einfügen kann. Anschließend wird nach dem Namen der Annoy- und SQLite-Datenbanken gefragt. Es ist möglich, verschiedene oder gleiche Namen für die Datenbanken zu verwenden, da das Programm automatisch ".ann" und ".db" anhängt. Dies geschieht, um die Verwendung des Systems unkompliziert zu gestalten.

Die Bilder können verschiedene Formate haben, und das Programm ist so konzipiert, dass alle gängigen Formate erkannt werden. Der Benutzer wird auch Benachrichtigungen über das "notification_interval" erhalten. Dies kann der Benutzer im Code selbst einstellen, aber für die Arbeitsweise des Programms ist es unerheblich, ob es auf 1, 1000 oder 1 Million eingestellt ist. In diesem Beispiel wird nach jeder Bildverarbeitung eine Benachrichtigung angezeigt, dass das Bild erfolgreich encodiert und gespeichert wurde.

In der "annoy_db"-Instanz werden Funktionen bereitgestellt, die mit dem Clip-Modell übereinstimmen, während "angular" für die Bilderkennung verwendet wird. Dann folgt die Erstellung der SQLite-Datenbank. Anschließend wird sichergestellt, dass alle Bilder nacheinander encodiert und gespeichert werden, und doppelte Bilder werden übersprungen, um sicherzustellen, dass der Fortschritt erhalten bleibt, selbst wenn das Programm unterbrochen und später wieder gestartet wird.

Am Ende wird dem Benutzer mitgeteilt, dass alles fertig ist und das System bereit für die Suche ist. Ein wichtiger Punkt ist hierbei die Einstellung bei "annoy_db.build(2)". Bei großen Datenmengen oder Bildern sollten die Bäume angepasst werden. Es wird empfohlen, durchschnittlich maximal 250 bis 400 Bilder pro Baum zu verwenden, damit die Suche schnell und präzise bleibt. Es ist zwar möglich, mehr Bilder pro Baum zu verwenden, aber dies wird nicht empfohlen. Das habe ich bereits auch am Anfang erwähnt.

Hier sind wir an dem Punkt angelangt, an dem der Nutzer mit "Nein" auf die erste Frage antwortet, was bedeutet, dass er einen bereits erstellten Annoy und SQLite Datenbanken nutzen möchte. In diesem Fall muss der Benutzer nun die Namen der Datenbanken eingeben, und die Datenbankdateien sollten in diesem Verzeichnis liegen, wo auch neue Datenbanken automatisch vom Programm gespeichert werden. Hier ist ein Beispiel für den Pfad: "C:\Users\EasyTony\Main10KAnn1000.db".

Dann wird der "Such-Loop" gestartet. Hier kann der Nutzer entweder das Programm schließen, Informationen durch Eingabe von "Info" erhalten oder einfach eigene Bildbeschreibungen eingeben. Die Beschreibungen sollten idealerweise auf Englisch verfasst werden, da Sätze in anderen Sprachen schlechter erkannt werden und daher zu sehr schlechten Ergebnissen führen können. Wenn jedoch Englisch verwendet wird, treten solche Probleme nicht auf. Die Ergebnisse liegen mindestens bei 80 bis 90% Genauigkeit, basierend auf Tests mit über 50 Sätzen. Deutsch und Japanisch leichter Sätze wie "Flugzeug steht am Flughafen", klappt gut, aber schwerere Sätze nicht gut bis sehr schlecht bzw 0% Genauigkeit. Das liegt daran, dass das Modell auf englisch trainiert wurde.

Nachdem der Nutzer die Beschreibung eingegeben hat, kann er sich wünschen, 1 bis 10 Bilder auszugeben, die angezeigt oder gespeichert werden sollen. Dies kann auch auf 20 oder 50 erweitert werden, indem die Zeile 184 angepasst wird sowie die dazugehörige Abfrage in Zeile 183. Das Programm bestätigt dann, wie viele Bilder der Nutzer angefordert hat, und gibt diese aus. Es fragt auch den Benutzer, ob er die Bilder in einem Ordner speichern oder sie einfach nur anzeigen möchte. Wenn der Benutzer sie nur anzeigen möchte, ohne sie zu speichern, also "No", muss er den Pfad angeben, in dem die Bilder gespeichert sind. Diese müssen auf dem PC oder einem USB-Stick vorhanden sein, da sonst eine Anzeige nicht möglich ist. Bei der Auswahl von "Yes" entscheidet sich der Benutzer dazu, einen neuen Ordner für die Bilder anzulegen und diesem einen Namen zuzuweisen. Anschließend wird der Prozess wiederholt, und der Nutzer kann eine weitere Beschreibung eingeben. Das Programm läuft so lange, bis der Benutzer "exit" eingibt.

Anleitung

a. Vorbereitung

Ich werde es für **Anaconda Spyder** erklären.

- 1. Anaconda herunterladen:** Gehen Sie auf die Anaconda-Website (<https://www.anaconda.com/products/distribution>) und laden Sie die Version für Ihr Betriebssystem herunter (Windows, macOS, Linux).
- 2. Spyder in Anaconda installieren:** Nachdem Sie Anaconda erfolgreich installiert haben, können Sie Spyder über die Anaconda Navigator-Anwendung installieren.
- 3. Anaconda Prompt öffnen:** Die Anaconda Prompt ist eine Befehlszeile Anwendung, die Ihnen Zugriff auf Anaconda- und Python-spezifische Befehle gibt.
- 4. Geben Sie die folgenden Befehle** in die Anaconda Prompt ein oder kopieren Sie sie und fügen Sie sie mit einem Rechtsklick ein:

```
pip install git+https://github.com/openai/CLIP.git  
pip install torch Pillow annoy numpy
```

- 5. Sobald Sie die Befehle** eingegeben haben und keine Fehlermeldungen auftreten, wurden die Pakete erfolgreich installiert.
- 6. Geben Sie “spyder”** in die Anaconda Prompt ein und drücken Sie die Eingabetaste, um Spyder zu öffnen.
- 7. Kopieren Sie den Code von GitHub** und fügen Sie ihn in Spyder ein, oder laden Sie ihn herunter und öffnen Sie ihn in Spyder. GitHub Link:
- 8. Wenn Sie keine spezielle Datenbank** haben, können Sie mit Ihrem Projekt starten.
- 8.1 Wenn Sie eine Datenbank** haben, kopieren Sie sie und fügen Sie sie an den angegebenen Ort ein:
Pfad: Dieser PC - Datenträger - Benutzer - Ihr Benutzername - Rechtsklick auf ein leeres Feld - Einfügen.
- 9. Starten Sie den Code in Spyder.** Beim ersten Mal wird die Annoy-Bibliothek heruntergeladen.
- 9.1. Nun können Sie die Eingaben** entsprechend den Anforderungen machen.

Wichtig zu beachten: Wenn Sie Bilder in die Bibliothek laden möchten, müssen Sie die Bäume entsprechend einstellen (Zeile 116):

```
annoy db.build(Hier eingeben Zahl von unten)  
annoy_db.save(annoy_db_path)
```

Es sollten pro Baum maximal 250 bis 400 Bilder sein. Man kann zwar mehr pro Baum machen, aber dann könnten die Ergebnisse schlechter sein bzw. man muss die Bilder, die man sucht, genauer beschreiben. Deswegen empfehle ich 100 bis 250 Bilder bis zu 100.000 Bildern und ab 100.000 250 bis 400 Bilder pro Baum. Bitte keine Komma-Zahlen verwenden, runden Sie auf ganze Zahlen.

Es ist wichtig zu beachten, dass ohne CUDA (Grafikkarte) die Berechnung auf der CPU durchgeführt wird, was pro 900 bis 1300 Bilder etwa eine Stunde dauern kann.

b. Ausführung

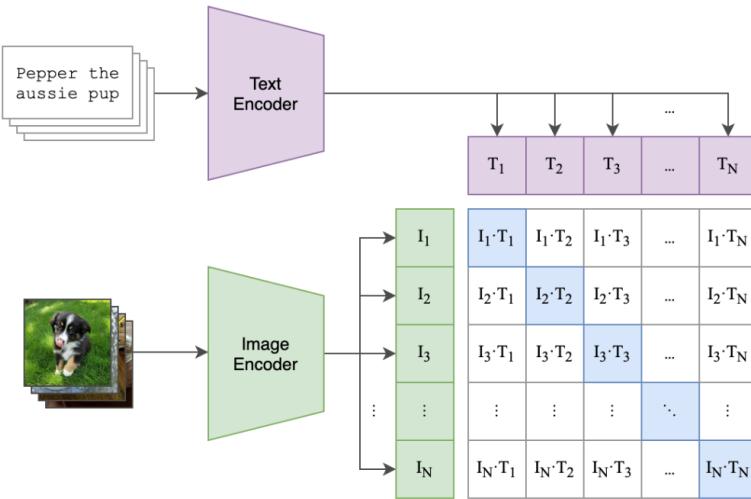
- 1. Was wollen Sie machen?**
 - a. Mit "Yes" können Sie eigene Bilder für durchsuche basieren auf Text dafür vorbereiten.
 - b. Mit "No" können Sie die bereits vorbereitete Datenbank basierend auf Ihrer Beschreibung durch suchen. Dafür müssen sie die Bilder auf externe oder interne Speicherplatz haben und vorbereitete Dateien mit ".ann" und ".db" in den folgenden Pfad einfügen.
"Lokale Datenträger(C):\Users\Username\" in Ordner Username einfügen.
 - c. Mit "Info" bekommen Sie kurze Erklärungen und Sie können sich danach erneut entscheiden.
- 2. Eingabe von Pfad und Datenbanken.**
 - a. Wenn Punkt 1. "Yes"
Pfad eingeben wo ihre Bilder gespeichert sind.
Die Name für Annoy Datenbank eingeben
Die Name für SQLite Datenbank eingeben
**Die Namen können identisch sein, aber dürfen nicht doppelt auf ihren Rechner vorkommen.*
Warten bis das Programm Hinweise gibt, dass Encoding und Speichern fertig sind:
"The images were successfully added."
 - b. Wenn Punkt 1. "No"
Die Name für Annoy Datenbank eingeben
Die Name für SQLite Datenbank eingeben
*Davor muss man diese Dateien auch unter Richtigen Pfad speichern (siehe "1.No")
- 3. Das gesuchte Bild beschreiben.**
 - a. Sie können hier die Beschreibung schreiben. Empfehlung, schreiben Sie die Beschreibung auf Englisch für genauere Resultate, andere Sprachen wie Japanisch und Deutsch funktionieren nur bei leichten Sätzen wie "Flugzeug am Flughafen".
 - b. Mit "Info" bekommen Sie eine kurze Erklärung.
 - c. Mit "Exit" beenden Sie das Programm.
- 4. Menge an Bilder/Resultate**
Hier können Sie aussuchen, wie viele Resultate sie sich wünschen, 1 bis 10.
- 5. Speichern oder Anzeigen**
 - a. Bei "Yes" werden die Resultate in separate Ordner gespeichert. Die Ordner Speicherort sowie auch Ordnername können Sie selber festlegen.
 - b. Bei "No" müssen Sie Pfad eingeben, wo die Bilder gespeichert sind, auch USB Stick sein. Danach werden Bilder im Spyder bei der Abteilung "Plot" angezeigt.

Sie kommen wieder zu Punkt 3, wenn Sie Datenbanken wechseln oder selber erstellen wollt müsst ihr "Exit" eingeben und das Programm erneut Starten

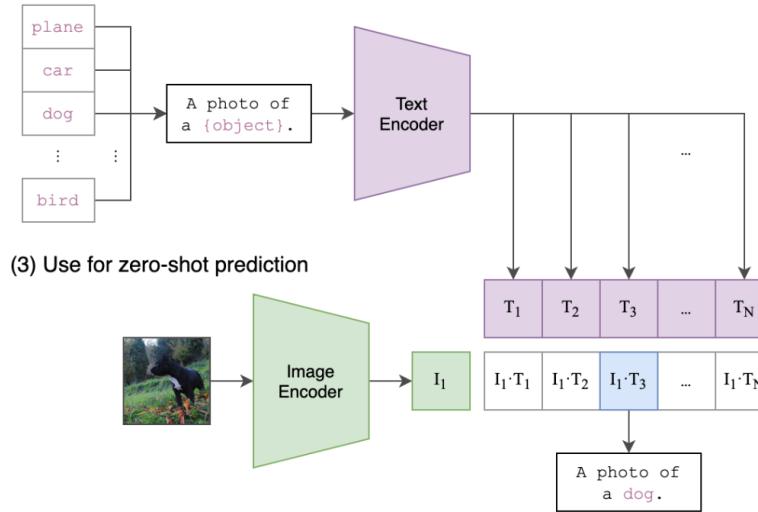
Technische Details

a. Contrastive Language-Image Pretraining (CLIP)

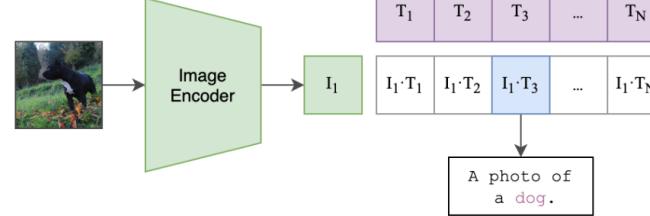
(1) Contrastive pre-training



(2) Create dataset classifier from label text



(3) Use for zero-shot prediction



CLIP wird zunächst auf großen Datensätzen von Bildern und Texten vortrainiert. Dabei lernt das Modell, semantische Repräsentationen für sowohl Bilder als auch Texte zu erstellen.

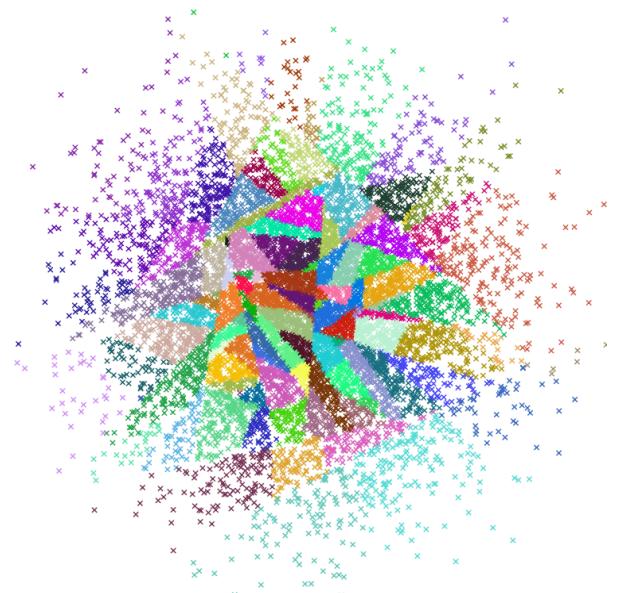
CLIP verwendet separate Embedding-Funktionen für Bilder und Texte, um sie in Vektorräume einzubetten. Diese Funktionen transformieren Bilder und Texte in dichte Vektoren mit einer bestimmten Anzahl von Dimensionen, wobei jeder Eintrag im Vektor ein Merkmal repräsentiert. Nachdem Bilder und Texte eingebettet wurden, wird die Ähnlichkeit zwischen ihnen im gemeinsamen Raum der Merkmale berechnet. Dies kann beispielsweise durch die Berechnung des Skalar. Aufgabenbereich ist zum Beispiel mein Projekt.²

b. Der Annoy-Datenbank

Die Annoy-Datenbank ist eine Bibliothek für Approximation von nächsten Nachbarn, entwickelt von Spotify. Sie bietet effiziente Suchzeiten für große Datensätze, indem sie eine spezielle Indexstruktur verwendet. Annoy ist darauf spezialisiert, Näherungslösungen für das NN-Problem bereitzustellen und ist in verschiedenen Programmiersprachen wie Python, C++ und Java verfügbar. Durch die Optimierung von Speicher- und Rechenprozessen ermöglicht Annoy schnelle Suchzeiten, während die Genauigkeit erhalten bleibt. Sie können Ihre Daten in die Annoy-Indexstruktur laden und dann schnell Nachbarn für neue Punkte finden. Insgesamt ist Annoy eine leistungsstarke Lösung für das NN-Problem, besonders nützlich bei der Verarbeitung großer Datenmengen.³

² <https://github.com/openai/CLIP>

³ <https://github.com/spotify/annoy>

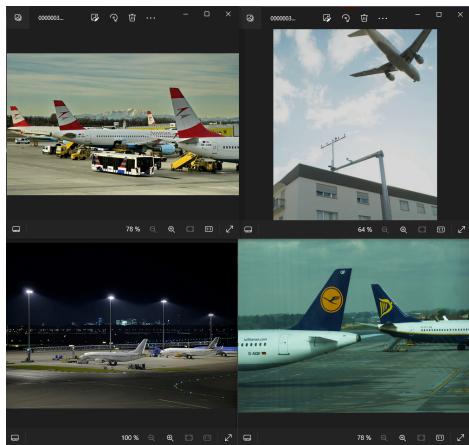


Testergebnisse

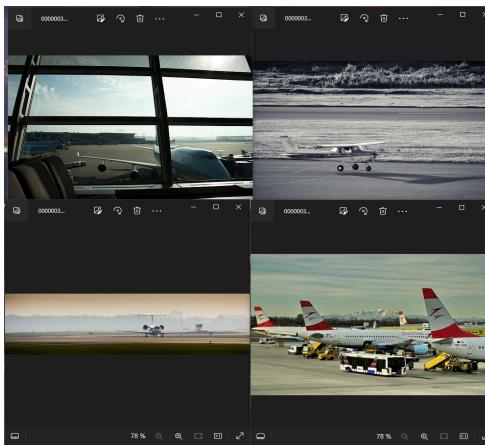
Der Clip hat verschiedene Modelle, die ich von Anfang an ausprobiert habe, um das beste Modell zu finden. Das Modell ViT-L/14@336px fand ich am besten. Es benötigt zwar mehr Speicher und Rechenkapazität, aber die Genauigkeit ist deutlich besser als bei anderen Modellen. In diesem Teil werde ich auf Tests und Anpassungen eingehen.

Datenbanken: Main10KAnn10.ann/.db

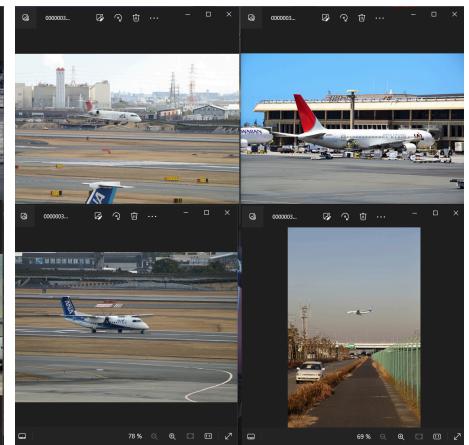
Details: ca. 10.000 Bilder; Bäume 1000; ca. 10 Bilder pro Baum



4



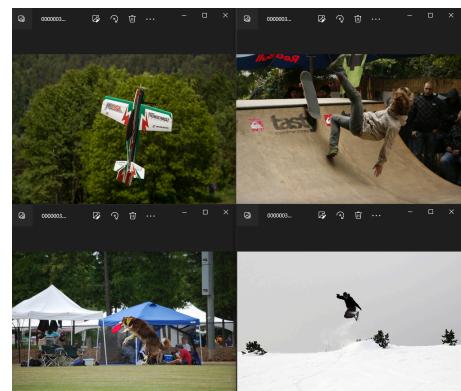
5



6

Also, hier sehen wir, dass Deutsch, Englisch und Japanisch sehr gut funktionieren. Interessant ist auch, dass die Bilder immer unterschiedlich sind, obwohl der Inhalt des Textes gleich bleibt.

Es ist wichtig, den genauen Text einzugeben, um das passende Bild zu erhalten. Wenn der Text nicht genau ist, können auch sehr schlechte Ergebnisse erzielt werden. Der Clip benötigt eine fast genaue Beschreibung, denn je mehr Informationen er aus dem Text erhält, desto besser kann er sie einordnen. Beispiel, Eingabe Deutsch: “Flugzeug”



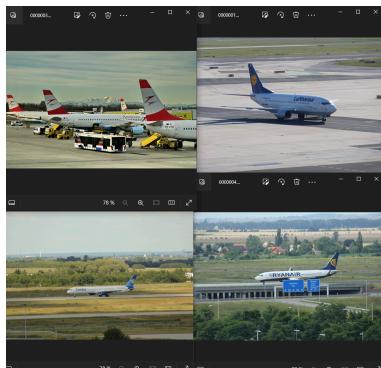
⁴ Passagierflugzeug am Flughafen

⁵ Passenger airplane at the airport

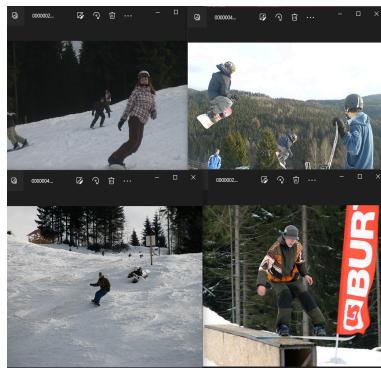
⁶ 空港の旅客機

Datenbanken: Main123K.ann/.db

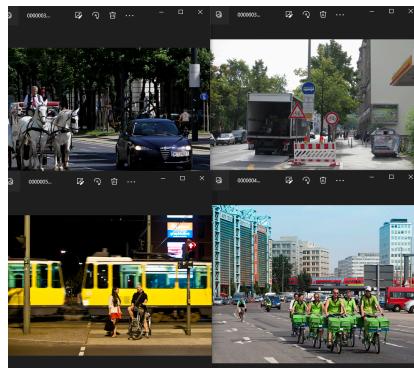
Details: ca. 123.000 Bilder; Bäume 500; ca. 250 Bilder pro Baum



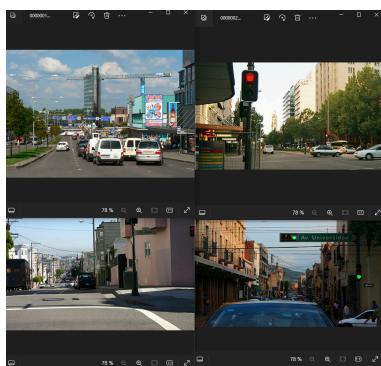
7



8



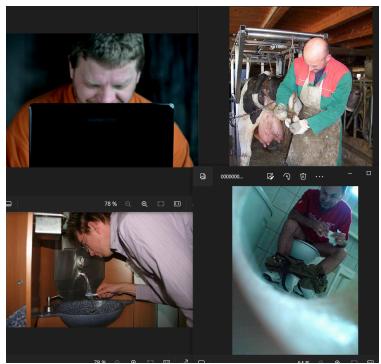
9



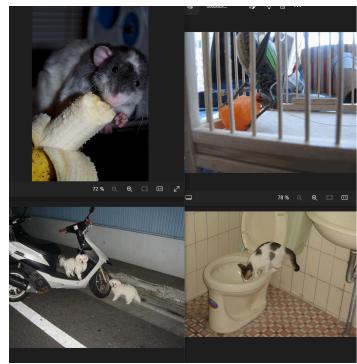
10

Es ist erneut offensichtlich, dass bessere Ergebnisse erzielt werden, wenn die Eingabe in englischer Sprache erfolgt, obwohl der Kontext derselbe bleibt. Deshalb ist es gelegentlich nützlich, selbst wenn die Ergebnisse nicht gut sind, die Eingabe ins Englische zu übersetzen. Ich werde nun weitere Tests mit japanischer Sprache durchführen, da meiner Meinung nach selbst bei meinem ersten Test die japanische Eingabe bessere Ergebnisse erzielt hat als die deutsche oder englische Eingabe. Die Übersetzung wird mit ChatGPT erfolgen.

Test Komplexe Sätze



11



12



13

⁷ Passagierflugzeug am Flughafen

⁸ Snowboarden in den Bergen

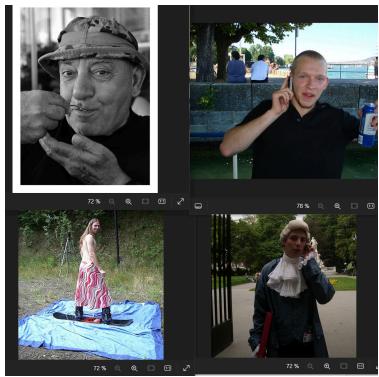
⁹ Das Auto fährt durch die Stadt

¹⁰ The car drives through the city.

¹¹ Der Mann arbeitet auf den Feldern.

¹² 男性は畑で働いています

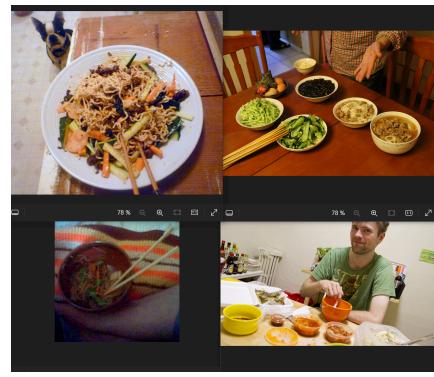
¹³ The man works in the fields.



14



15



16

Schwierigkeiten

Anschließend möchte ich über die Schwierigkeiten sprechen. Die größte Herausforderung bestand darin, alles in einem System zusammenzuführen und die Steuerung ausschließlich über die Konsole zu ermöglichen. Mein Programm bestand aus 5 separaten Systemen: Bilder encodieren in Vektoren und speichern, Pfad der kodierten Bilder in SQLite speichern, das ähnlichste Bild anhand seiner ID auswählen, diese ID in SQLite eingeben und den entsprechenden Pfad erhalten, dann den Ordner mit den Bildern suchen und dort den Pfad kopieren und das Bild anzeigen. Es war also sehr kompliziert und benutzerunfreundlich, da die Eingaben im Code und nicht in der Konsole erfolgten. Daher musste man den Code jedes Mal anhalten, die Eingabe ändern und dies in jedem System tun. Anschließend wurde mir gesagt, dass ich dies benutzerfreundlicher gestalten und die Eingaben nur über die Konsole stattfinden lassen soll, und es sollte ein einziges Programm mit diesen 5 Systemen sein. Der größte Bereich der Herausforderung war dann auch die Aufgabenverteilung und die Übersicht im Code. Ich habe es geschafft, alles zu erledigen, außer den Code sauber zu halten. Man könnte meinen, er sieht gut aus, aber ich finde trotzdem, dass er nicht sehr "sauber" aussieht. Doch um ihn zu verbessern, hatte ich weder das Wissen noch ehrlich gesagt die Kraft. Bei jeder Änderung hatte mein Code einen Fehler bekommen, was alles noch mehr durcheinander brachte.

Aktuelle Problem

Heute, am 19.02. um 20:00 Uhr, konnte ich beim Testen meiner Präsentation feststellen, dass die "Show" Funktion auf meinem Laptop überhaupt nicht funktioniert hat. Die Konsole ist ständig abgestürzt, obwohl sie auf meinem PC einwandfrei lief. Warum das so ist und warum mein Laptop damit Probleme hat, konnte ich nicht herausfinden. In der Präsentation werde ich ein YouTube-Video zeigen: <https://youtu.be/BtNSyTfDLnY>. So kann ich trotzdem diese Funktion demonstrieren. Fehler Screenshot: Letzte Seite.

¹⁴ Der Mann isst Nudeln

¹⁵ その男性は麺を食べます

¹⁶ The man eats noodles.

Nutzung des Projekts

Das Programm hat aus meiner Sicht viele verschiedene Anwendungsbereiche, die auch von großen Unternehmen wie Google praktiziert werden. Wenn man einen Text eingibt, werden einem Bilder vorgeschlagen, die zum jeweiligen Text passen können. Ein Fotograf könnte dieses Programm verwenden, um Bilder zu speichern und sie später schneller zu finden. Man kann dieses Programm auch zur Sortierung verwenden, wenn man zum Beispiel Flugzeugbilder und Snowboardbilder herausfiltern möchte. Dabei werden verschiedene Bilder verwendet, die alle zufällig sind. Wenn ein normaler Mensch beispielsweise eine Galerie mit Urlaubsbildern in den Bergen sortieren möchte, kann er dieses Programm verwenden. Wenn wir über eine große Menge an Bildern, bis zu 50.000, sprechen, kann jeder PC diese innerhalb von maximal 10 Stunden codieren, ohne dass eine Grafikkarte benötigt wird. Also bietet dieses Programm verschiedene Möglichkeiten. Es ist jedoch nicht geeignet, um ein bestimmtes Bild genau zu finden. Wenn eine große Auswahl an Bildern vorliegt und das Bild zu 80-90% genau beschrieben wird, kann man jedoch ein gewünschtes Ergebnis erhalten.

Weitere Schritte

Als nächstes könnte man dem Modell eine benutzerfreundliche Schnittstelle geben, damit der Benutzer nicht alles in der Konsole eingeben muss, sondern dies in einem Programm tun kann. Man könnte auch zusätzliche Informationsquellen integrieren und es dem Benutzer ermöglichen, die Anzahl der Bäume manuell oder benutzerdefiniert einzugeben. Anschließend könnte man das Programm als Beta-Version veröffentlichen und sehen, was dem Programm noch fehlt. Der erste Schritt wäre, herauszufinden, warum mein Laptop abstürzt, wenn ich "Bilder anzeigen" in Spyder verwende, und die Ursache für diesen Fehler zu finden und zu beheben.

Eigene Test

Hier ist der Google Drive-Link: [Klick](#)

Dort finden Sie einige Bilder, die heruntergeladen und dann codiert werden können. Sie können damit herumspielen und sie selbst testen.

Quellen:

Dokumentation:

<https://deeplobe.ai/text-based-image-retrieval-using-deep-learning/>

<https://github.com/openai/CLIP>

<https://github.com/spotify/annoy>

Grammatik- und Rechtschreibfehler; Google Drive und ChatGPT.

Chat GPT:

You
Korrigiere mein Text überprüfe Grammatik und Rechtschreibung
Im letzten Semester meines Studiengangs Daisy, im Fachgebiet Intelligente sichere Systeme und Big Data Engineering, haben wir darüber gesprochen, wie man große Mengen von Daten sortiert und auf diese zugreifen kann, während man sie insbesondere mit Bildern beschäftigt haben. Unser Ziel war es dabei, verschiedene Methoden zu entwickeln, um ein passendes Bild zu finden. Drei gewählte Methoden waren RGB und Hash. Dieses Projekt hat mir viel Spaß gemacht, und das Ergebnis war sehr beeindruckend und interessant. Dadurch wurde mein Interesse geweckt, im Fach Advances in Intelligent Systems das Thema NLP-basiertes Imageretrieval zu wählen, da es nichts an mein Thema des letzten Semesters anschließt und es auf ein neues Level hebt.

ChatGPT

Im letzten Semester meines Studiengangs Daisy, im Fachgebiet Intelligente sichere Systeme und Big Data Engineering, haben wir darüber gesprochen, wie man große Mengen von Daten sortiert und darauf zugreifen kann, wobei wir uns insbesondere mit Bildern beschäftigt haben. Unser Ziel war es, drei verschiedene Methoden zu entwickeln, um ein passendes Bild zu finden. Mehr gewählte Methoden waren RGB und Hash. Dieses Projekt hat mir viel Spaß gemacht, und das Ergebnis war sehr beeindruckend und interessant. Dadurch wurde mein Interesse geweckt, im Fach Advances in Intelligent Systems das Thema NLP-basiertes Imageretrieval zu wählen, da es nichts an mein Thema des letzten Semesters anschließt und es auf ein neues Level hebt.

You
Korrigiere mein Text überprüfe Grammatik und Rechtschreibung
Anschließend möchte ich über Schwierigkeiten sprechen. Die größten Herausforderungen bestanden darin, alles in einem System zusammenzuführen und die Steuerung ausschließlich über die Konsole zu ermöglichen. Mein Programm bestand aus 5 separaten Systemen: Bilder encodieren in Vektoren und speichern, Pfad der encodierten Bilder in SQLite speichern Anhang der ID, das ähnliche Bild anhand seiner ID auswählen, diese ID wieder in SOLITE eingegeben und den entsprechenden Pfad erhalten, dann den Ordner mit den Bildern suchen und dort den Pfad kopieren und das Bild anzeigen. Es war also sehr kompliziert und umgekehrt zu benutzerfreundlich, da die Eingaben in Code und nicht in der Konsole erfolgten. Daher musste man Code jedes Mal anhalten, die Eingabe ändern und diese in jedem System tun. Anschließend wurde mir gesagt, dass ich dies benutzerfreundlicher gestalten und die Eingaben nur über die Konsole stattfinden lassen soll, und es sollte ein einziges Programm mit diesen 5 Systemen sein. Der größte Bereich der Herausforderung war dann auch die Aufgabenverteilung und die Übersicht im Code. Ich habe es geschafft, alles zu erledigen, außer den Code sauber zu halten. Man könnte meinen, haben andere Studenten gesagt, er sieht gut aus; aber ich finde trotzdem, dass er nicht sehr "sauber" aussieht. Doch um ihn zu verbessern, hätte ich weder das Wissen noch ehrlich gesagt die Kraft. Bei jeder Änderung hatte mein Code einen Fehler bekommen, was alles noch mehr durcheinander brachte.

ChatGPT

Anschließend möchte ich über Schwierigkeiten sprechen. Die größten Herausforderungen bestanden darin, alles in einem System zusammenzuführen und die Steuerung ausschließlich über die Konsole zu ermöglichen. Mein Programm bestand aus 5 separaten Systemen: Bilder codieren in Vektoren und speichern, Pfad der codierten Bilder in SQLite speichern Anhang der ID, das ähnliche Bild anhand seiner ID auswählen, diese ID wieder in SOLITE eingegeben und den entsprechenden Pfad erhalten, dann den Ordner mit den Bildern suchen und dort den Pfad kopieren und das Bild anzeigen. Es war also sehr kompliziert und keineswegs benutzerfreundlich, da die Eingaben im Code und nicht in der Konsole erfolgten. Daher musste man den Code jedes Mal anhalten, die Eingabe ändern und dies in jedem System tun. Anschließend wurde mir gesagt, dass ich dies benutzerfreundlicher gestalten und die Eingaben nur über die Konsole stattfinden lassen soll, und es sollte ein einziges Programm mit diesen 5 Systemen sein. Der größte Bereich der Herausforderung war dann auch die Aufgabenverteilung und die Übersicht im Code. Ich habe es geschafft, alles zu erledigen, außer den Code sauber zu halten. Andere Studenten mögen meinen, er sieht gut aus, aber ich finde trotzdem, dass er nicht sehr "sauber" aussieht. Doch um ihn zu verbessern, hätte ich weder das Wissen noch ehrlich gesagt die Kraft. Bei jeder Änderung hatte mein Code einen Fehler bekommen, was alles noch mehr durcheinander brachte.

You
Korrigiere mein Text überprüfe Grammatik und Rechtschreibung
Das Programm hat aus meiner Sicht viele verschiedene Anwendungsbereiche, die auch von großen Unternehmen wie Google benutzt werden. Wenn man einen Text eingibt, werden Bilder vorgeschlagen, die zum jeweiligen Text passen können. Ein Fotograf könnte dieses Programm verwenden, um Bilder zu speichern und später schneller zu finden. Man kann dieses Programm auch zur Sortierung verwenden, wenn man zum Beispiel Flugzeugbilder und Snowboardbilder herausfinden möchte. Dabei werden verschiedene Bilder verwendet, die alle zufällig sind. Wenn ein normaler Mensch beispielsweise eine Galerie mit Urlaubsbildern in den Bergen sortieren möchte, kann er dieses Programm verwenden. Wenn wir über eine große Menge an Bildern, bis zu 10.000 Bilder sprechen, kann jeder PC diese innerhalb von ca. 10 Stunden encodieren, ohne dass eine Grafikkarte benötigt wird. Also bietet dieses Programm verschiedene Möglichkeiten. Es ist jedoch nicht geeignet, um ein bestimmtes Bild genau zu finden. Wenn eine nicht große Auswahl von Bildern vorliegt (max. 100) und das Bild zu 80-90% genau auf Englisch beschrieben wird, kann man jedoch ein gewünschtes Ergebnis erhalten.

ChatGPT

Das Programm hat aus meiner Sicht viele verschiedene Anwendungsbereiche, die auch von großen Unternehmen wie Google genutzt werden. Wenn man einen Text eingibt, werden Bilder vorgeschlagen, die zum jeweiligen Text passen können. Ein Fotograf könnte dieses Programm verwenden, um Bilder zu speichern und später schneller zu finden. Man kann dieses Programm auch zur Sortierung verwenden, wenn man zum Beispiel Flugzeugbilder und Snowboardbilder herausfinden möchte. Dabei werden verschiedene Bilder verwendet, die alle zufällig sind. Wenn ein normaler Mensch beispielsweise eine Galerie mit Urlaubsbildern in den Bergen sortieren möchte, kann er dieses Programm verwenden. Wenn wir über eine große Menge an Bildern sprechen, bis zu 10.000 Bildern, kann jeder PC diese innerhalb von ca. 10 Stunden encodieren, ohne dass eine Grafikkarte benötigt wird. Also bietet dieses Programm verschiedene Möglichkeiten. Es ist jedoch nicht geeignet, um ein bestimmtes Bild genau zu finden. Wenn eine nicht große Auswahl von Bildern vorliegt (max. 100) und das Bild zu 80-90% genau auf Englisch beschrieben wird, kann man jedoch ein gewünschtes Ergebnis erhalten.

Code: Frage-Forums, Reddit, GitHub, Bibliotheken. Beispiele bzw ansätze der Chat GPT vorgeschlagen und auch Fehlermeldungen.

Fehlermeldung.

```
□ Console 1/A X
[...]
linked into the process, e.g. by avoiding static linking of the OpenMP runtime in any library. As an unsafe, unsupported, undocumented workaround you can set the environment variable KMP_DUPLICATE_LIB_OK=TRUE to allow the program to continue to execute, but that may cause crashes or silently produce incorrect results. For more information, please see http://www.intel.com/software/products/support/.

Fatal Python error: Aborted

Main thread:
Current thread 0x00003e4c (most recent call first):
  File "<_array_function_> internals", line 180 in dot
  File "C:\ProgramData\anaconda3\lib\site-packages\matplotlib\transforms.py", line 2441 in get_affine
  File "C:\ProgramData\anaconda3\lib\site-packages\matplotlib\transforms.py", line 2442 in get_affine
  File "C:\ProgramData\anaconda3\lib\site-packages\matplotlib\patches.py", line 590 in draw
  File "C:\ProgramData\anaconda3\lib\site-packages\matplotlib\artist.py", line 72 in draw_wrapper
  File "C:\ProgramData\anaconda3\lib\site-packages\matplotlib\figure.py", line 3124 in draw
  File "C:\ProgramData\anaconda3\lib\site-packages\matplotlib\artist.py", line 72 in draw_wrapper
  File "C:\ProgramData\anaconda3\lib\site-packages\matplotlib\artist.py", line 95 in draw_wrapper
  File "C:\ProgramData\anaconda3\lib\site-packages\matplotlib\backend_bases.py", line 2338 in print_figure
  File "C:\ProgramData\anaconda3\lib\site-packages\IPython\core\pylabtools.py", line 152 in print_figure
  File "C:\ProgramData\anaconda3\lib\site-packages\IPython\core\formatters.py", line 338 in __call__
  File "C:\ProgramData\anaconda3\lib\site-packages\IPython\core\formatters.py", line 221 in catch_format_error
  File "C:\ProgramData\anaconda3\lib\site-packages\decorator.py", line 232 in fun
  File "C:\ProgramData\anaconda3\lib\site-packages\IPython\core\formatters.py", line 177 in format
  File "C:\ProgramData\anaconda3\lib\site-packages\IPython\core\display_functions.py", line 298 in display
  File "C:\ProgramData\anaconda3\lib\site-packages\matplotlib_inline\backend_inline.py", line 90 in show
  File "C:\ProgramData\anaconda3\lib\site-packages\matplotlib\pyplot.py", line 445 in show
  File "d:\main nlp (with with saddle and show).py", line 17 in show_image
  File "d:\main nlp (with with saddle and show).py", line 210 in <module>
  File "C:\ProgramData\anaconda3\lib\site-packages\spyder_kernels\py3compat.py", line 356 in compat_exec
  File "C:\ProgramData\anaconda3\lib\site-packages\spyder_kernels\customize\spydercustomize.py", line 469 in exec_code
  File "C:\ProgramData\anaconda3\lib\site-packages\spyder_kernels\customize\spydercustomize.py", line 736 in _exec_cell
  File "C:\ProgramData\anaconda3\lib\site-packages\spyder_kernels\customize\spydercustomize.py", line 685 in runcell
  File "C:\Users\zhevn\AppData\Local\Temp\ipykernel_2544\1448990993.py", line 1 in <module>

Restarting kernel...
```