

Bildverarbeitung und Computer Vision. Übungen zur Programmierung in den Bildwissenschaften

WiSe 2021/2022

Michael Schlee

Übungsblatt 7

Es handelt sich bei der Kombination dieser zwei folgenden Übungsaufgaben auf eine direkte Vorbereitung auf die Prüfungsleistung selbst. In der kommenden Übung werden (wegen der Abgabepflicht) selbstverständlich keine Musterlösungen zur Verfügung gestellt. Jedoch können Ansätze, Probleme sowie Codeabschnitte verglichen und besprochen werden. Sollte keinerlei oder nur wenig Zeit schlussendlich darauf verwendet werden, so wird die restliche Zeit der Übung genutzt um weitere Übungsaufgaben und Problemstellungen gemeinsam zu lösen. Nutzen Sie die Übungen also gerne, um über jegliche Probleme mit diesen zwei Aufgaben zu sprechen. Diese zwei Übungsaufgaben sind die letzten Aufgaben mit Abgabepflicht. Ebenso werden die folgenden Übungen sich mit Priorität um Probleme und Fragen zu diesen zwei Aufgaben beschäftigen. Viel Spaß beim Üben!

1 Colour Stylometry - ohne Keras/Tensorflow - Abgabe (bis 05.02.) für die Prüfungszulassung notwendig.

Die Stilometrie ist eine Forschungsmethode, die unter anderem von Digital Humanities-Wissenschaftlern verwendet wird, um den Computer darauf zu trainieren, Stilmarker in verschiedenen Artefakten zu erkennen und zu identifizieren, wie z. B. Manuskripte, Texte und - im Fall dieser Aufgabe - große Kunstwerke. Daraus folgt also, dass es erkennbare Merkmale von Gemälden gibt, die Aufschluss darüber geben, ob ein Gemälde ein Rembrandt oder ein DaVinci ist, da diese Künstler in unterschiedlichen regionalen Kulturen und in verschiedenen Epochen gearbeitet haben. In dieser Aufgabe werden Sie einige Gemälde auf diese kulturellen "Stichwörter" hin untersuchen und bestimmen, welcher Stil dem von drei "geheimnisvollen" Gemälden am ähnlichsten ist.

Für diese Aufgabe der Art Stylometry werden Sie zwei Skripts erstellen:
Ein Skript "trainiert" Ihr System auf die Unterschiede in der Farbwahl zwischen den künstlerischen Werken von zwei Meisterkünstlern: Rembrandt Harmenszoon van Rijn (Rembrandt) und Leonardo di ser Piero da Vinci (da Vinci). Ein zweites Skript verwendet die Trainingsdaten aus dem ersten Skript, um die Durchschnittswerte für die Farbwerte in den Trainingsdaten mit den Farbwer-

ten die in einem "mysteriösen" Gemälde enthalten sind zu vergleichen. Sobald der Vergleich durchgeführt wurde, wird das zweite Skript die Trainingsarbeit identifizieren, die dem "geheimnisvollen" Gemälde in der Farbzusammensetzung am ähnlichsten ist.

Skript 1-Trainingsskript

Erstellen Sie ein Skript, das jedes der sechs Bilder (die drei DaVinci-Gemälde und die drei Rembrandt-Gemälde, die in der Zip-Datei mit den Bildern enthalten sind - siehe StudIP) und eine Schleife zum Akkumulieren der gesamten Rot-, Grün- und Blauwerte (RGB) jedes Pixels in entsprechenden Akkumulator-Variablen. Nach Ausführung der Schleife werden die Durchschnittswerte (akkumulierte Gesamt Beträge/Anzahl der Pixel) für Rot, Grün und Blau in einer Textdatei für jedes der sechs Bilder gespeichert werden. Laden Sie hierfür die Bilder mittels einer der Ihnen bekannten Bibliotheken, wandeln Sie in NumPy Arrays und durchlaufen diese entsprechend. Ihr Trainingsskript sollte diese Aufgaben erfüllen:

- a) Durchlaufen Sie für jedes Gemälde die Pixel und analysieren Sie jedes Bild auf diese drei Merkmale: Durchschnittswert, berechnet aus den jeweiligen Gesamtwerten für Rot, Grün und Blau über alle Pixel
- b) Speichern Sie den Dateinamen und die Durchschnittswerte für jedes der sechs Gemälde in einer Textdatei zur Verwendung in Skript 2.

SKript 2 - Spekulationsskript Nachdem das Trainingsskript abgeschlossen ist und eine geeignete Ausgabedatei erzeugt wurde, erstellen Sie nun ein Skript, das diese Aufgaben übernimmt:

- a) Laden Sie die Daten aus der erzeugten Trainingsskriptdatei ein
- b) Laden Sie das "Mystery-Portrait" und analysieren Sie es auf die gleiche Weise, wie Sie es für die Trainingsbilder in Skript 1 gemacht haben.
- c) Vergleichen Sie jedes der Merkmale im Mystery-Porträt (durchschnittliches Rot, durchschnittliches Grün, und durchschnittliches Blau) mit diesen Durchschnittswerten der sechs Trainingsporträts, indem Sie eine einfachen Abstandsformel auf jeder Dimension durchführen:
$$\text{Abstand} = | \text{trainierter Wert} - \text{Mystery-Wert} | / \text{Mystery-Wert}.$$
- d) Bestimmen Sie, welches das "nächstgelegene" Porträt zum Mystery-Porträt ist, indem Sie alle der berechneten Abstände addieren und die kleinste Summe aus allen sechs Porträts wählen.
- e) Zeigen Sie sowohl das geheimnisvolle Porträt als auch das nächstgelegene Trainingsporträt an und geben Sie die zugehörigen Werte für beide aus.

Wenn Sie beide Skripts abgeschlossen haben, zippen Sie NUR die beiden Python-Skripte selbst (nicht die Trainings- oder Mystery-Bilder und auch nicht Ihre

Trainingsdaten-Ausgabedatei) und reichen Sie diese bis zum Abgabetermin per Mail ein. Wenn ich Ihre Dateien zum Test verwende, stelle ich sicher, dass sich alle Trainings- und Testbilder im selben Verzeichnis befinden wie Ihre Skripts. Ich werde Ihr Skript 1 ausführen, um Ihre Trainingsdaten zu erzeugen, und dann unmittelbar danach lasse ich Ihr Skript 2 auf einem der Testbilder laufen, um Ihre Ausgabe zu betrachten, die mir das Trainingsbild zeigen sollte, das in der Farbauswahl dem von mir gewählten Testbild am ähnlichsten ist. Ich kann eines der Testbilder verwenden die Ihnen zum Testen zur Verfügung gestellt wurden, oder ich verwende ein ganz anderes Testbild Bild für die Einstufung.

2 Keras DL Ansatz für Da Vinci & Rembrandt - Abgabe (bis 12.02.) für die Prüfungszulassung notwendig.

Verwenden Sie nun im Vergleich einen Deep Learning Ansatz mit Keras, um die Mystery-Bilder zu laden und eine Vorhersage für diese zu treffen. Sie können hierfür die bereits in der letzten Übung zur Verfügung gestellten Code-Teile benutzen und entsprechend anpassen und erweitern. Nutzen Sie dafür die per Owncloud zur Verfügung gestellten Datensets "Rembrandt_DaVinci_wikiart" die von hier und hier entnommen wurden.

Sie können diese beiden Datensets beliebig erweitern oder kürzen. Deren Verteilung als Grundlage dient an dieser Stelle nur der Entlastung. Versuchen Sie auch beim eigentlichen Training eine möglichst hohe Genauigkeit zu erzielen und notieren Sie welche Einstellungen für Sie am Besten funktioniert haben. Dokumentieren Sie mindestens 3 verschiedene Lernansätze in verwendeten Parametern sowie zugehörigen Genauigkeiten. Zudem verwenden Sie die Log-Möglichkeit von Tensorboard für ihren finalen Code mit der besten Genauigkeit. Diese Logs geben Sie bitte gemeinsam mit dem finalen Code und der beschriebenen (sehr kurzen!) Dokumentation ab.