Meine Vorgehensweise

Das Auswählen eines Datensets ist der erste und wahrscheinliche wichtigste Schritt beim Programmieren einer KI, denn mit der Qualität des Datensets steigt und fällt die Qualität der KI. Zudem muss das Datenset auch die genauen Daten enthalten, die man benutzen will. Auch die Grösse eines Datensets ist entscheidend. Man sollte ein Datenset wählen, welches gross genug ist, um die KI zu trainieren. Zu grosse Datensätze gibt es keine, solange der Computer freien Speicherplatz hat. Der meiner Meinung nach beste Ort, um an solche Datensets zu gelangen ist eine Website namens Kaggle. Auf dieser Website können alle Personen mit Zugang zum Internet, Datensätze veröffentlichen und herunterladen. Die Seite hat auch Wettbewerbe mit Preisgelder, Kurse und Foren für Diskussionen rund um das Thema KI. Wer sich für Deeplearning und Datascience interessiert sollte diese Website sicher einmal auschecken. Ich habe mir zuerst überlegt, ein eigenes Datenset zu kreieren, da ich kein Datenset gefunden habe, welches meinen Vorstellungen entsprach. Glücklicherweise hat der Nutzer NikiTricky aus Bulgarien ein Datenset erstellt, welches genau meinen Erwartungen entsprach. Das Datenset heisst Street View Panoramas und enthält 187177 Bilder mit den dazugehörigen Koordinaten und einem Aufnahmedatum, welches ich jedoch nicht benutzte. NikiTricky erstellte das Datenset laut eigenen Angaben, indem er ein Programm schrieb, welches bei zufälligen Koordinaten testet, ob es ein Google Streetview Bild gibt. Wenn ein Bild existiert, wird dieses heruntergeladen und die Koordinaten mit dem Aufnahmedatum werden in einer CSV-Datei gespeichert. Zudem wird dem Bild noch ein Name gegeben, damit man die Koordinaten und das Bild nicht mit anderen Koordinaten und Bilder verwechselt.

Nachdem man alle Daten heruntergeladen hat, muss man die Daten so Strukturieren, dass man zum Trainieren möglichst viel Zeit einsparen kann. Ich habe bei meiner KI die Bilder in einer CSV-Datei gespeichert, damit ich für das Trainieren der KI viel schneller zugriff auf die Bilder habe und nicht jedes Mal wieder alle Werte aller Pixel mit Pillow herholen muss. Zudem muss man den Datensatz in zwei Teile teilen. Den Trainings und der Test-teil. Dies ist wichtig, weil es passieren kann, dass sich die KI zu stark am Datenset anpasst und dann nur noch bei Bildern aus dem Datensatz ein akzeptables Resultat ausgibt. Damit man also einen Vernünftigen Test zur Genauigkeit der KI machen kann, müssen Daten verwendet werden, die nicht zum trainieren benutzt wurden.

Nachdem die Daten im passenden Format waren, konnte ich bereits mit dem Programmieren der KI beginnen. Ich entschied mich dafür die KI in Python zu programmieren, da ich die Sprache bereits gut beherrsche und es einige gute Bibliotheken für Deeplearning gibt, die effizient sind und einfach zu benutzen. Ich habe mich dazu entschieden Pytorch für mein Projekt zu benutzen, da es vielerorts empfohlen wird und auch von vielen grossen Firmen benutzt wird. Da ich schon einmal eine KI programmiert habe, konnte ich einen Teil meines Codes kopieren und musste daher nicht den gesamten Code von Grund an schreiben. Ich musste lediglich die Struktur des Neuronalen Netzwerks anpassen und die Daten im Korrekten Format an die KI füttern. Dies hat jedoch eine Weile gedauert, da ich vor diesem Projekt noch nie mit CSV-Dateien gearbeitet habe und auch noch ein Problem hatte. Das Problem war, dass ein Bild nicht existiert hat. Somit war eine Liste mit Daten Null und ich bekam immer wieder eine Fehlermeldung. Als ich das Problem erkannte, konnte ich die entsprechende Zeile auch nicht so einfach löschen, da das Dokument zu gross war für verschiedene Texteditoren. Ich musste also ein eigenes Programm schreiben, um eine spezifische Zeile meines Datensets zu löschen. Danach funktionierten der Code und ich hatte endlich mein erstes trainiertes Model.

Um das Resultat der KI zu verbessern habe ich mir überlegt, eine zweite KI zu programmieren, die die Klimazone des Aufnahmeortes erkennen kann. Dazu musste ich mich in Shapefiles einlesen und ein Programm programmieren, welches die Klimazone erkennen kann. Danach kam mir noch die Idee, dass ich die Distanzen zu den verschiedenen anderen Klimazonen auch noch berechnen kann und somit eine viel grössere Ausgabe der KI habe. Diese Ausgabe kann ich dann wiederum verwenden, um meine Haupt-KI, welche die Koordinaten vorhersagt, zu trainieren.