BSPro-Klassifizierung von Blumen mit Faltungs-Neuronalen Netzen Zusammenfassung

Sunday 21st June, 2020 - 14:41

Licina Esada
Universität von Luxemburg
Email: esada.licina.001@student.uni.lu

Kieffer Emmanuel
Universität von Luxemburg
Email: emmanuel.kieffer@uni.lu

***Abstract*—Diese Zusammenfassung über den finalen Bericht, wurde von der Studentin Esada Licina und mit der Unterstützung ihres Projekt Tutors Emmanuel Kieffer gemacht. In diesem Projekt wird der wissenschaftliche und technische Teil beschrieben.Für mehr Information, bitte den finalen Bericht durchlesen.**

1. Einführung und Projektbeschreibung

Klassifikation ist ein Begriff, das ein strukturiertes Universum von gegebenen Daten darstellt. Hier wird eine Datei aus Kaggle benutzt und es besitzt 5 Kategorien von Blumen. Dies wird mehr im technischen Bereich erläutert. Dann gibt es noch die Faltungs-Neuronalen Netze (FNN), welches ein Modell ist um Daten zu erkennen und zu kategorisieren. Dieser Begriff wird deutlicher im wissenschaftlichen Teil erklärt.

Dieses Projekt ist ein Teil von der Künstlichen Intelligenz (KI) und zu dieser gehören die zwei Untergruppen maschinelles und tiefes Lernen. In diesen Untergruppen wird sich das Projekt mehr fokussieren, es bezieht sich auf mathematischen Theorien und Algorithmen um durch Mustern von Daten zu lernen und sie wiederzuerkennen. Hier wird eine 'supervised' Methode verwendet wo das Modell Daten bekommt und dadurch lernt.

Um das alles zum Laufen zu bringen, wird die Programmiersprache Python benutzt mit Hilfe von der FastAi Bibliothek. Es werden zwei Module benutzt, einer um ein Modell zu bauen und der andere ist für die Echtzeitklassifizierung.

2. Voraussetzungen

Man fängt zuerst mit einem Plan an, damit man sich organisiert. Dann verbringt man viel zeit damit Informationen über das Thema zu finden, um das ganze mal gut zu verstehen. Nach der Theorie, beginnt man mit dem Programmierern.

3. Das Gehirn von Faltungs-Neuronalen Netzen

3.1. Anforderungen

Funktionale Anforderungen:

1) Datenverarbeitung von Blumen.

Nicht-funktionale Anforderungen:

1) Die Effizienz des FNNs.

3.2. Design und Produktion

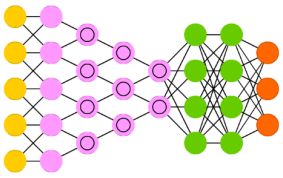

Figure 1. FNN Struktur

Im oberen Bild sieht man Wie ein FNN aufgebaut ist. Seine Architektur besteht aus drei Hauptschichten:

1 **Die Faltungsschicht (Merkmal-Lernen Teil):** Hier wird das Skalarproduckt verwendet bei der Eingabe Matrix vom Bild mit einer Filter Matrix. Nach jeder Faltungsschicht wird ein Aktivierung Funktion eingesetzt, die ReLU genannt wird. Zur Information, ein Bild besteht aus Pixel Werten die eine Matrix bilden.

2 **Die Pooling-Schicht (Merkmal-Lernen Teil):** Hier wird gesorgt das die Matrix verfeinert wird.

3 **Die vollständig verbundene Schicht (Klassifikation Teil:** Nach diesem Durchlauf, wird dann das Ergebnis vorhergesagt.

Für den FNN wird eine spezielle Architektur benutzt, das ResNet34 Modell, welche aber die 3 Hauptschichten behält.

3.3. Bewertung

Die Anforderungen wurden erfolgreich vom Projekt erfüllt.

4. Klassifizierungsprozess in PyCharm

4.1. Anforderungen

Funktionale Anforderungen:

1) Aufbau eines Modells.
2) Echtzeitklassifizierung.

Nicht-funktionale Anforderungen:

1) Benutzerfreundlichkeit von PyCharm.
2) Leistung.

4.2. Design und Produktion

In dem unterem Bild erkennt man die verschieden Schritte die man tun muss um die Echtzeitklassifikation hinzubekommen.

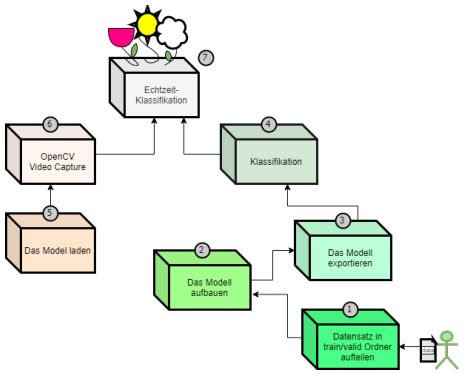

Figure 2. Klassifikation Prozess

1 Zuerst wird der Datensatz aufgeteilt in zwei verschieden Ordnern. Eine für das Training (80% vom Datensatz) und das andere für das Testen (20% vom Datensatz). Hier wird die Funktion np.split() benutzt von der Bibliothek numpy.
2 Dann wird das Modell gebaut von dem Datensatz, mit der ResNet34 Architektur. Die Funktion cnn_learner() von FastAi ist hilfreich in dem Bereich.
3 Das Modell wird dann exportiert, damit man es verwenden kann. Hier kann man learn.export() benutzt von FastAi.
4 Das Modell wird dann getestet mit unbekannten Blumen Bildern, ob es fehlerfrei läuft. Die Funktion learn.predict() ermöglicht das.
5 Dann geht es weiter beim Zweiten Teil vom Code. Das Modell wird geladen was exportiert wurde, damit man es benutzen kann.
6 Jetzt wird mit Hilfe von der Bibliothek OpenCV eine Videoaufnahme erstellt. Das Video wird dann in Bilder umgewandelt und durch das Modell eingeordnet.
7 Im Bild unten sieht man das Resultat, wo über dem rosa Fenster die Prognose vom Modell erscheint.

Figure 3. Ergebnis von Videoaufnahme

4.3. Bewertung

An den Bildern erkennt man,dass die Anforderungen erfolgreich erfüllt wurden.

5. Abschluss und Anerkennung

Schlussendlich wurde das Ziel erreicht und ein Modell für Blumen Klassifikation erstellt.

Ich, Esada Licina danke meinem Tutor Emmanuel Kieffer, der BICS Gruppe, meiner Familie und Freunden für die Unterstützung.