Objekterkennung mit YOLO (You Only Look Once)

YOLO ist ein populärer und leistungsstarker Ansatz für die Objekterkennung in Echtzeit. Es wurde von Joseph Redmon und seinen Kollegen entwickelt und revolutionierte die Art und Weise, wie Objekterkennung durchgeführt wird. YOLO ist ein Deep-Learning-Modell, das darauf abzielt, Objekte in Bildern und Videos in Echtzeit zu erkennen und zu lokalisieren. Im Gegensatz zu traditionellen Methoden, die auf region-basierten Ansätzen wie R-CNN basieren, betrachtet YOLO das gesamte Bild auf einmal. Das ermöglicht eine schnelle und genaue Objekterkennung. YOLO gibt es mittlerweise schon in vielen Versionen. Wir verwenden die aktuellste Version 8.

Grundprinzipien von YOLO

Im Gegensatz zu anderen Methoden, die das Bild in Regionen unterteilen und jede Region separat analysieren, betrachtet YOLO das gesamte Bild auf einmal. Dies führt zu einer einheitlichen und konsistenten Objekterkennung. YOLO verwendet ein einzelnes neuronales Netzwerk, das das Bild in ein festgelegtes Raster unterteilt. Jeder Rasterzelle werden dann die Wahrscheinlichkeiten und die Positionen der Objekte zugewiesen, die sich in dieser Zelle befinden könnten. Aufgrund seiner Architektur und Effizienz ist YOLO in der Lage, Objekte in Echtzeit zu erkennen, was es besonders nützlich für Anwendungen wie autonome Fahrzeuge, Videoüberwachung und Augmented Reality macht.

Die YOLO-Architektur besteht aus einer Abfolge von Convolutional Layers, die die Eingabebilder verarbeiten. Das ursprüngliche YOLO-Modell hat folgende Eigenschaften: - Eingabegröße: Das Bild wird auf eine feste Größe (z.B. 448x448 Pixel) skaliert. - Grid: Das Bild wird in ein S x S Raster (z.B. 7x7) unterteilt. - Bounding Boxes: Jede Zelle im Raster prognostiziert eine feste Anzahl von Bounding Boxes und die zugehörigen

Konfidenzwahrscheinlichkeiten. - Klassifikationen: Jede Zelle prognostiziert auch die Wahrscheinlichkeit, dass jedes Objekt in einer der vordefinierten Klassen liegt.

1. Eingabe und Vorverarbeitung

Eingabebild: Ein Bild in beliebiger Größe wird als Eingabe verwendet. Größenanpassung: Das Eingabebild wird auf eine feste Größe (z.B. 416x416 Pixel) skaliert, damit es durch das neuronale Netzwerk verarbeitet werden kann. Normalisierung: Die Pixelwerte des Bildes werden normalisiert, um die Berechnungen im Netzwerk zu erleichtern.

1. Netzarchitektur

YOLO verwendet eine Convolutional Neural Netzwerk (CNN), das aus mehreren Schichten von Convolutional Layers, Batch Normalization, ReLU-Aktivierungsfunktionen und Pooling-Schichten besteht. Die ersten Schichten des Netzwerks extrahieren Feature Maps aus dem Eingabebild. Diese Schichten bestehen aus mehreren Convolutional Layers, die dafür sorgen, dass wichtige Merkmale (z.B. Kanten, Texturen) erkannt werden. Die letzten Schichten des Netzwerks sind dafür verantwortlich, die Objekte zu detektieren und die Bounding Boxes sowie die Klassenvorhersagen zu machen.

1. Rasteraufteilung, Bounding Boxes und Klassenvorhersage Das Bild, das dem YOLO-Modell zugeführt wird, hat eine feste Größe (z.B. 416x416 Pixel). Dieses wird in ein Gitter mit S×S Zellen aufgeteilt. Ein typischer Wert für S ist 7, 13, oder 19, abhängig von der verwendeten YOLO-Version und den gewünschten Kompromissen zwischen Genauigkeit und Geschwindigkeit. Jede Rasterzelle ist verantwortlich für die Erkennung von Objekten, deren Mittelpunkt (Zentrum) innerhalb dieser Zelle liegt. Wenn ein Objekt sich so positioniert, dass sein Mittelpunkt in die Zelle (3, 4) des Rasters fällt, dann ist diese spezielle Zelle für die Erkennung und Vorhersage der Bounding Box und der Klasse dieses Objekts verantwortlich. Jede Rasterzelle sagt mehrere (typischerweise 2 bis 5) Bounding Boxes voraus. Für jede dieser Boxen werden folgende Parameter vorhergesagt:
2. (x, y): Relative Koordinaten des Mittelpunkts der Bounding Box innerhalb der Rasterzelle. Diese Werte sind normalisiert, sodass sie zwischen 0 und 1 liegen.
3. w, h: Relative Breite und Höhe der Bounding Box, ebenfalls normalisiert zur Bildgröße.
4. Confidence Score: Ein Wert, der die Wahrscheinlichkeit angibt, dass die Bounding Box tatsächlich ein Objekt enthält.

Zusätzlich zu den Bounding Boxes sagt jede Rasterzelle eine Wahrscheinlichkeitsverteilung über alle möglichen Klassen voraus. Diese Werte geben an, wie wahrscheinlich es ist, dass sich das Objekt, das von der Zelle erkannt wird, in eine der Klassen einordnet. Wenn Objekte über mehrere Zellen überlappen, dann wird nur die Zelle, in der der Mittelpunkt des Objekts liegt, für die Vorhersage dieses speziellen Objekts verantwortlich gemacht. Wenn mehrere Objekte ihren Mittelpunkt in

derselben Zelle haben, kann das Modell Schwierigkeiten haben, sie korrekt zu unterscheiden. In der Praxis ist dies jedoch selten, da das Raster in der Regel fein genug ist, um die meisten Szenarien zu handhaben.

1. Kombinieren der Ergebnisse Der endgültige Score für jede Box wird durch Multiplikation des Konfidenz-Scores mit der jeweiligen Klassenwahrscheinlichkeit berechnet. Der Algorithmus Non-Maximum Suppression (NMS) wird verwendet, um redundante Boxen zu eliminieren. Nur die Boxen mit den höchsten Scores und ohne signifikante Überlappung werden behalten.

2. Ausgabe Endgültige Vorhersagen: Die verbleibenden Boxen, nach Anwendung von NMS, sind die endgültigen Vorhersagen des Modells. Jede Box enthält:

3. Die Koordinaten (x,y,w,h)
4. Eine Klassenkennung
5. Einen Konfidenz-Score

--> Lesen Sie sich das original Yolo Paper durch und notieren Sie sich wichtige Merkmale. Dabei müssen sie sich nicht die tiefgreifenderen Mathemathischen Hintergründe anschauen, sondern eher auf die grundlegenden Konzepte achten.