Projektplan

Wie funktioniert Object Detection, wie funktioniert YOLO?

## Projektbeschreibung

Für viele Anwendungen ist es heutzutage wichtig, in Bildern oder Videos Objekte zu entdecken und klassifizieren zu können. Im Bereich des Autonomous driving oder in der Verwendung von Drohnen ist das großer Bedeutung. Wir wollen Object Detection anwenden, um Autos und Motorräder in Fotos zu erkennen und zu zählen. Dafür benutzen wir das YOLO Modell für die Erkennung. Das Ergebnis werden klassifizierte Testbilder sein, in denen die erkannten Objekte mit farbigen Bounding Boxes markiert sind. Zusätzlich werden die erkannten Fahrzeuge gezählt.

## Vorgehen

Das ist zwar der Kern meines TO DO gewesen, aber ich hab das Gefühl viel besser als dieser [Artikel](https://www.analyticsvidhya.com/blog/2018/12/practical-guide-object-detection-yolo-framewor-python/) und dieses [Notebook](https://github.com/enggen/Deep-Learning-Coursera/blob/master/Convolutional%20Neural%20Networks/Week3/Car%20detection%20for%20Autonomous%20Driving/Autonomous%20driving%20application%20-%20Car%20detection%20-%20v1.ipynb) (das tatsächlich eine Angabe mit Erklärung und TO DOs für Vehicle detection ist!) kann ich es gar nicht erklären.

### Erkenntnisse

Wir könnten ein trainiertes Modell verwenden (siehe Notebook), dann brauchen wir gar keine Labels und müssen nur die bereits gefundenen Weights verwenden. Das würde viel Arbeit sparen und wahrscheinlich bessere Ergebnisse liefern. Autos, Motorräder und LKWs werden außerdem bei dem vortrainierten Modell erkannt. Wenn wir es selbst machen möchten müssen wir zusätzlich zu dem Vorgehen oben die Daten bearbeiten damit sie diese Form haben:

* train\_image\_folder <= the folder that contains the train images.
* train\_annot\_folder <= the folder that contains the train annotations in VOC format.
* valid\_image\_folder <= the folder that contains the validation images.
* valid\_annot\_folder <= the folder that contains the validation annotations in VOC format.

Und das Modell trainieren (z.B. mit [diesem](https://github.com/experiencor/keras-yolo3) Code für YOLO3 in Keras).

Laut dieser Erklärung auf Github könnte man das Training auch mit dem COCO Datenset machen, weil die Bilder mit anderen Labels als die die man angibt eh ignoriert werden. Mit COCO trainiert ist allerdings das bereits trainierte Modell aus dem Notebook auch.

Mein Vorschlag ist, dass wir das trainierte Modell nutzen um mal ein funktionierendes Programm zu schreiben. Falls wir das schnell hinkriegen dann können wir probieren ein eigenes Modell zu bauen. Das Gute daran ist auch, dass wir anscheinend einfach selbst Bilder verwenden können um das Modell zu testen, was heißt wir könnten interessante Ergebnisse erzielen wenn wir sehr unterschiedliche Arten von Fotos (Winkel, Lichtverhältnisse, etc.) ansehen.