

Abstract

Deep learning wordt steeds meer gebruikt om beeldverwerkingsproblemen zoals **herkennings-systemen** en **detectiesystemen** op te lossen. Via neurale netwerken kunnen we met meer en betere features werken om de afbeeldingen te analyseren. Veel van deze modellen hebben echter behoorlijk wat rekenkracht en geheugen nodig om te werken. Om deze reden wordt bij veel huidige mobiele toepassingen het deep learning model uitgevoerd in de cloud. Hierdoor kan de mobiele toepassing echter niet werken zonder een internetconnectie. Een ander probleem is dat veel operaties die een bestaand model uitvoert niet compatibel zijn met de **mobiele omgeving** waar we het model willen gebruiken. Een veel gebruikte taal voor het ontwerpen en trainen van neurale netwerken is Python. Deze modellen worden in een Python omgeving uitgevoerd terwijl Android studio applicaties in een Java omgeving worden uitgevoerd. Het doel van deze masterproef is het onderzoeken van de **compatibiliteit** tussen de operaties van een **bestaand neurale netwerk** in zijn originele omgeving en een mobiele omgeving, zodat een bestaand en complex deep learning model geïmplementeerd kan worden op een mobiel apparaat met zo min mogelijk effect op de accuraatheid.

We gaan deze compatibiliteit onderzoeken voor de ResNet50, YOLO en Faster-RCNN architectuur. We vertrekken van bestaande architecturen die ontworpen en getraind zijn in het TensorFlow en PyTorch framework. Deze twee frameworks hebben elk een eigen methode om het model te converteren naar een taalafhankelijk model. We stellen vast dat TensorFlow de meeste operaties ondersteunt voor een Android studio implementatie. De PyTorch bibliotheek voor Android studio bevat minder operaties dan TensorFlow maar voert zijn operaties wel sneller uit. PyTorch moet zijn niet ondersteunde operaties op een alternatieve manier importeren in Android studio. Uiteindelijk zijn de drie architecturen implementeerbaar in Android studio met weinig of geen effect op de accuraatheid.