



**Bewegungserkennung auf mobilen Geräten mit Verwendung
von GANs für eine automatische Datensatzgenerierung**

Master-Thesis

Florian Hansen
Hochschule Flensburg

30. Mai 2021

Inhaltsverzeichnis

1	Einleitung	3
2	Generative Adversarial Networks	4
2.1	Das Mode-Collapse-Problem	5
2.2	Deep Convolution GAN	6
2.3	Wasserstein GAN	6
2.4	Wasserstein GAN mit Gradient Penalty	6
2.5	Unrolled GAN	6
2.6	Least Squares GAN	6
3	Erstellen eines Datensatzes	7
3.1	Rahmenbedingungen	7
3.2	Verwendung von GANs	7
3.3	Durchführung von Experimenten mit unterschiedlichen GANs	7
3.4	Analyse der Ergebnisse aus den Experimenten	7
4	Bewegungserkennung	8
4.1	Ground-Truth	8
4.2	Background-Subtraction	8
4.3	Erkennung von Geschwindigkeiten	8
4.4	Erkennung von Anomalien	8
4.5	Erkennung von Bewegungsarten	8
4.6	Vorhersage von Bewegungen	8
4.7	Architektur einer mobilen Anwendung	8
5	Fazit und Ausblick	9

1 Einleitung

2 Generative Adversarial Networks

In Machine-Learning existieren viele verschiedene Modelle, die vorhandene Datensätze analysieren und anhand der Daten lernen, Strukturen in den Datensätzen zu erkennen. Besitzt man beispielsweise einen Datensatz bestehend aus Fotoaufnahmen von Tieren, so kann ein Klassifizierer trainiert werden, um einem Bild eine Tierklasse zuzuweisen. Aus diesem Grund fasst man diese Modelle unter dem Begriff *Bildklassifizierung* zusammen.

Wesentlich interessanter ist das Erkennen von vielen Objekten innerhalb eines Bildes, anstatt das gesamte Bild nur einer einzigen Klasse zuzuweisen. In der *Objekterkennung* entwickelt man Modelle, welche mehr als nur eine Klasse erkennen können. Sie liefern zusätzlich zu den erkannten Klassen ihre Position und Größe innerhalb des Bildes. Diese Modelle treffen also keine Aussage über das Gesamtbild, sondern treffen Aussagen über einzelne Objekte innerhalb des Bildes.

Neben Modellen, die zu einem bestimmten Sachverhalt eine Aussage treffen können, existieren auch Modelle, welche in der Lage sind, neue Sachverhalte zu erzeugen. Diese fallen unter dem Begriff *Generative Adversarial Networks* (GANs) und bilden das Hauptthema dieses Abschnitts. Das interessante an diesen generativen Modellen ist, dass sie nicht nur die Strukturen eines Datensatzes lernen, sondern darüber hinaus neue Elemente der Ausgangsdistribution erzeugen können. Trainiert man also ein generatives Modell auf einen Datensatz, welcher Bilder von verschiedenen Tieren enthält, können neue Bilder der gleichen Art erzeugt werden.

Aber nicht nur zum Erzeugen von Bildern kann diese Art von Modellen verwendet werden. Auch bei Aufgaben, bei denen eine Voraussagung getroffen werden soll, werden generative Modelle eingesetzt. Beispielsweise wurde in [1] gezeigt, wie zu bereits getätigten menschlichen Bewegungen unterschiedliche, darauf folgende Bewegungssequenzen aussehen können. Hier hat man also versucht, eine Vorhersage zur Entwicklung von menschlichen Bewegung zu tätigen.

Die Funktionsweise von GANs ist im Prinzip ziemlich simpel. Während beim klassischen supervised-learning meistens nur ein Modell beim Training involviert ist, verhält

sich das bei generativen Modellen etwas anders. Zum Einen wird ein Generator definiert, welcher, wie sein Name andeutet, Ausgaben selbst erzeugt. Zum Anderen wird ein Diskriminator in das Training eingebaut, welcher zwischen künstlich erzeugten und reellen Daten unterscheidet. Diese beiden Modelle werden dann gleichermaßen Trainiert. Während der Generator versucht, Fälschungen immer genauer zu erzeugen, versucht der Diskriminator immer besser zwischen Fälschung und Realität unterscheiden zu können. Die Ausgabe ist daher entweder 0 für Fälschung und 1 für Realität. Die beiden Komponenten spielen mit anderen Worten ein Spiel, in welchem die eine Partei versucht, die andere zu täuschen [2].

$$\min_G \max_D V(G, D) = \mathbb{E}_{x \sim p_{data}(x)} [\log D(x)] + \mathbb{E}_{z \sim p_z(z)} [\log(1 - D(G(z)))]$$

Im Verlauf des Trainings entwickelt sich damit ein Generator, welcher im Idealfall so gute Fälschungen erzeugt, sodass sich diese nicht mehr von Daten der Ausgangsdistribution unterscheiden lassen. Der Diskriminator kann bestenfalls nur raten, sollte also eine Genauigkeit von 50% erreichen.

2.1 Das Mode-Collapse-Problem

Ein großes Problem beim Trainieren von generativen neuronalen Netzen ist, dass sich der Generator sehr häufig auf bestimmte Merkmale der Ausgangsdistribution des Datensatzes fixiert. Das Ergebnis sind signifikant erhöht wiederkehrende Ergebnisse, die sich kaum bis gar nicht von anderen Ausgaben unterscheiden. Man erwartet jedoch, dass das jeweilige GAN eine vielseitige Variation aus allen Elementen des Datensatzes erzeugt. Mit anderen Worten, bei einer zufälligen Eingabe in das Netz, soll immer eine unterschiedliche Ausgabe erzeugt werden. Bei einem Mode-Collapse ist dies nicht der Fall. Es kann beispielsweise passieren, dass wenn das Netz auf das Erzeugen von neuen Gesichtern trainiert wird, dass dieses ausschließlich weibliche Gesichter erzeugt, weil das Netz herausgefunden hat, dass es einfacher ist, weibliche Gesichtszüge zu generieren, als männliche [3]. Dies lässt sich damit erklären, dass der Generator beim Trainingsvorgang mehr Erfolg beim Generieren von weiblichen Gesichtern hatte und der Diskriminator es schwerer hatte, Fälschung von Realität zu unterscheiden.

2.2 Deep Convolution GAN

2.3 Wasserstein GAN

2.4 Wasserstein GAN mit Gradient Penalty

2.5 Unrolled GAN

2.6 Least Squares GAN

3 Erstellen eines Datensatzes

3.1 Rahmenbedingungen

3.2 Verwendung von GANs

3.3 Durchführung von Experimenten mit unterschiedlichen GANs

3.4 Analyse der Ergebnisse aus den Experimenten

4 Bewegungserkennung

4.1 Ground-Truth

4.2 Background-Subtraction

4.3 Erkennung von Geschwindigkeiten

4.4 Erkennung von Anomalien

4.5 Erkennung von Bewegungsarten

4.6 Vorhersage von Bewegungen

4.7 Architektur einer mobilen Anwendung

5 Fazit und Ausblick

Literatur

- [1] Emad Barsoum, John Kender und Zicheng Liu. *HP-GAN: Probabilistic 3D human motion prediction via GAN*. 2017. arXiv: 1711.09561 [cs.CV].
- [2] Ian J. Goodfellow u. a. *Generative Adversarial Networks*. 2014. arXiv: 1406.2661 [stat.ML].
- [3] Eitan Richardson und Yair Weiss. *On GANs and GMMs*. 2018. arXiv: 1805.12462 [cs.CV].