Wykrywanie obiektów w strumieniu danych video - teoria

Jerzy Szyjut 193064
Artur Binczyk 193138
Patryk Welenc 193241
Mateusz Fydrych 193410

Wprowadzenie

Rozpoznanie obrazu

Image recognition

Lokalizacja obrazu

Image localization

+

=

Rozpoznanie obrazu

+

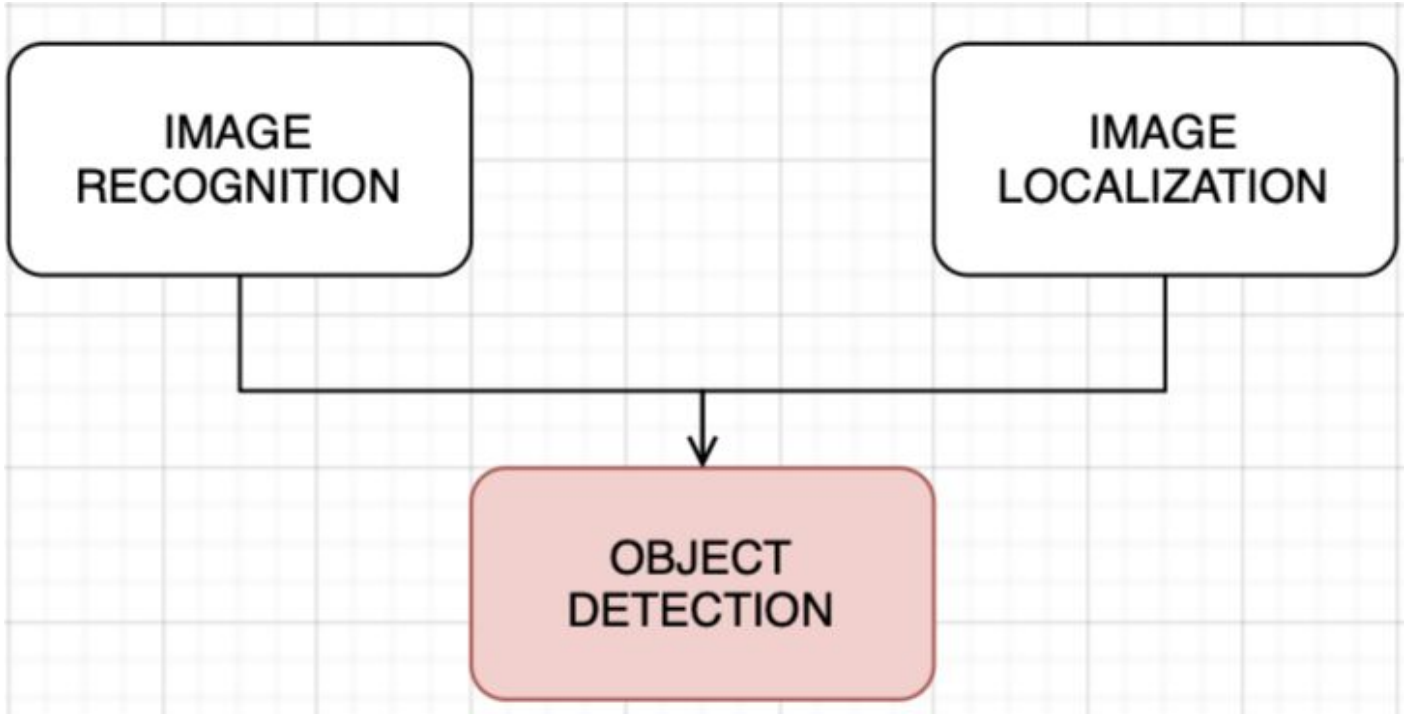
Lokalizacja obrazu

=

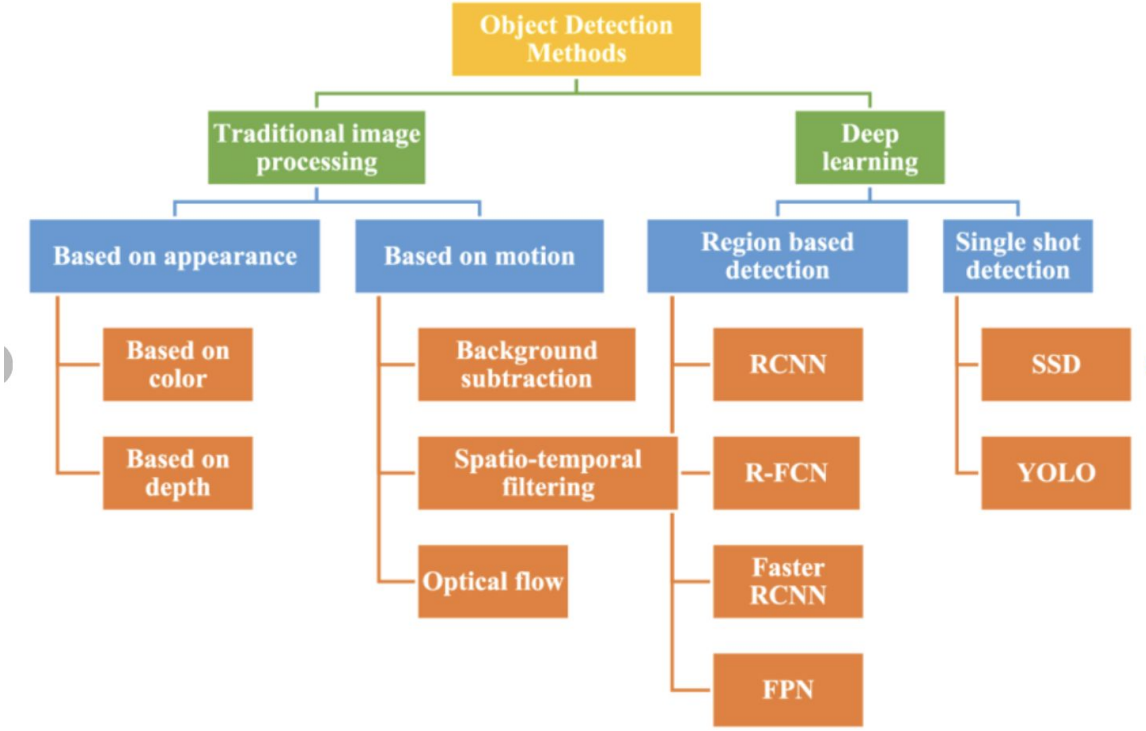
...

Detekacja obiektów
Object Detecion

Diagram zależności



Metody detekcji obiektów



TEORIA

Porównanie metod

	Faster R-CNN	YOLO (You Only Look Once)	DETR (DEtection TRansformer)
Paradygmat	Detekcja dwustopniowa	Detekcja jednostopniowa	Oparty na Transformerze
Sposób działania	Proponowanie Regionów (lub Generowanie Propozycji Regionów) oraz Klasyfikowanie Regionów	Przewidywanie Prostokątów Ograniczających i Klas w jednym przebiegu (lub w jednym kroku)	Przewidywanie zestawu obiektów za pomocą zapytań
Główny komponent	Sieć propozycji regionów (Region Proposal Network)	Siatka	Transformer typu Koder-Dekoder
Prostokąty kotwiczące	Tak używane przez RPN	Tak w większości wersji	Nie, zamiast tego używa zapytań o obiekty stanowiące bezpośrednie żądanie do dekodera
Usuwanie niemaksymalnych pikseli Non-Maximum Suppression"NMS"?	Tak wymagane w postprocessingu	Tak wymagane w postprocessingu	Nie wymagane
Szybkość	Najwolniejszy	Najszybszy najlepszy dla real-time	Zależy od wariantu
Dokładność	Wysoka	Bardzo wysoko (w szczególności w ostatnich wersjach)	Wysoka

CZY ZDARZYŁO WAM SIĘ KIEDYŚ BYĆ NAJLEPSZYM W JAKĄŚ GRĘ?!

A CZY ZDARZYŁO WAM SIĘ KIEDYŚ BYĆ NAJLEPSZYMI W COUNTER-STRIKE?!

A CZY...
CHCIELIBYŚCIE BYĆ NAJLEPSI W COUNTER-STRIKE?!

NIE?
NIC NIE SZKODZI!

(BO NASZA PREZENTACJA ZAKŁADA, ŻE CHCECIE)

Co planujemy?

Być najlepsi w CS2 😎

Czego potrzebujemy?

Umiejętności strzelania 🔫

Co nam w tym pomoże?

Wiedza przedstawiona w teorii 📖

W jaki sposób?

Pomożemy sobie w strzelaniu 🙏

Okej... To co robimy?

Cel: Utworzenie aimbota, który będzie „delikatnie wspomagał” nasze celowanie*

Tzn. nakierowywał za nas kursor prosto na przeciwnika, nam jedynie pozostanie pociągnąć za spust :)

*oczywiście wyłącznie w celach edukacyjnych, pamiętajcie że cheatowanie jest złe, a Święty Mikołaj patrzy...

Plan działania aimbota

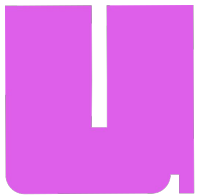
1. Zrobienie zrzutu ekranu, na którym będzie widoczna gra
2. Wrzucenie wykonanego zrzutu ekranu do wytrenowanego modelu, który zajmie się oznaczaniem przeciwników z określonym prawdopodobieństwem
3. Odczytanie wyniku z modelu i ustalenie koordynatów przeciwników
4. Przesunięcie kursora myszki w odpowiednie miejsce

Oczekiwany efekt końcowy

Mniej więcej coś takiego...

prezentacja tematu projektu, narzędzi, środowiska pracy i scenariusza demo,

Środowisko



pydirectinput
python mss

Uczenie maszynowe jako problem gier

arXiv:2205.07060v1 [cs.AI] 14 May 2022

GAN-Aimbots: Using Machine Learning for Cheating in First Person Shooters

Anssi Kanervisto, Tomi Kinnunen, and Ville Hautamäki

***Abstract*—Playing games with cheaters is not fun, and in a multi-billion-dollar video game industry with hundreds of millions of players, game developers aim to improve the security and, consequently, the user experience of their games by preventing cheating. Both traditional software-based methods and statistical systems have been successful in protecting against cheating, but recent advances in the automatic generation of content, such as images or speech, threaten the video game industry; they could be used to generate artificial gameplay indistinguishable from that of legitimate human players. To better understand this threat, we begin by reviewing the current state of multiplayer video game cheating, and then proceed to build a proof-of-concept method, *GAN-Aimbot*. By gathering data from various players in a first-person shooter game we show that the method improves players' performance while remaining hidden from automatic and manual protection mechanisms. By sharing this work we hope to raise awareness on this issue and encourage further research into protecting the gaming communities.**

I. Introduction

VIDEO games attract millions of players, and the industry reports their revenue in billions of dollars. For instance, one of the biggest video game publishers, Activision Blizzard, reported more than 75 million players of their game *Call of Duty: Modern Warfare (2019)* and net revenue of over 3.7 billion US dollars in the first half of 2020 [1]. The gaming communities also contain e-sport tournaments with prize pools in millions, e.g. World Electronic Sports Games 2016 contained a prize pool of 1.5 million USD. With such popularity, cheating practices similar to doping in physical sports is commonplace in video games. For example, two public cheating communities have existed since 2000 and 2001, and have more than seven million members in total [2], [3], and this is only a fraction of such communities. In these communities, users can share and develop different ways to cheat in multiplayer games. Although cheating is more difficult in in-person tournaments, cheating in online games is still prevalent. For example, UnknownCheats [3] has 213 threads and more than a million views for the game *Call of Duty: Modern Warfare (2019)*. The presence of cheaters degrades the user experience of other players, as playing with cheaters is not fun. Game developers are therefore encouraged to prevent such cheating.

A common way to cheat in online games is by using tools and software ("hacks") used by "hackers" [4], dating back to 1990 with the first forms of hacking with Game Genie [5]. Hacking is prohibited by game publishers, and they monitor players with anti-cheat software to detect hacking. A standard approach is to check running processes on the player's computer for known cheating software—similar to how antivirus scanners look for specific strings of binary on a machine. The publisher can also analyse players' behaviour and performance to detect suspicious activity, such as a sudden increase in a player's performance. While hacks can be hidden from the former detection approach, the latter cannot be bypassed, as the system runs on game servers. Analysis of player behaviour is thus an attractive option for publishers.

These data-based anti-cheats have also attracted the attention of the academic community, as well as from the game industry. Galli et al. (2011) [6] developed hacks that provide additional information to the player and move the mouse in the game *Unreal Tournament III*. The authors then used machine learning to distinguish these hackers from legitimate players. Hashen et al. (2013) [7] extended this work by evaluating the accuracy of different classifiers versus different hacks. Yeung and Lui (2008) [8] analysed players' performance and then used Bayesian inference to analyse players' accuracy and classify if a player was hacking.

All of these works have focused on **detecting** hackers with machine learning, but little attention had been given to **cheating with machine learning**. Yan and Randell [4] have mentioned the use of machine learning for cheating, but only in the context of board games like Chess and Go and not in video games. Recent advances in machine learning allow one to generate photorealistic imagery [9], speech to attack voice biometric systems [10] or computer agents that mimic the demonstrators [11], [12]. Similar methods could be applied to video games to generate human-like gameplay. We define "human-like gameplay" as artificially generated gameplay that is indistinguishable from genuine human gameplay, either by other humans or by automated detection systems. If used for cheating, these methods threaten the integrity of multiplayer games, as the previously mentioned anti-cheat systems could not distinguish these cheaters.

To develop an understanding of this threat, we design a machine learning method for controlling the computer mouse to augment the player's gameplay and study how effective it is for cheating. This work and its contributions can be summarised as follows:

- We review different ways people hack in multiplayer video games and methods for detecting such cheaters.
- We present a proof-of-concept machine learning method, GAN-Aimbot, for training a hack that mimics hu-

All authors are with School of Computing, University of Eastern Finland, Joensuu, Finland. V. Hautamäki is also with the Department of Electrical and Computer Engineering, National University of Singapore. E-mail: anssk@uef.fi, tkinnu@uef.fi, villeh@uef.fi

https://arxiv.org/abs/2205.07060

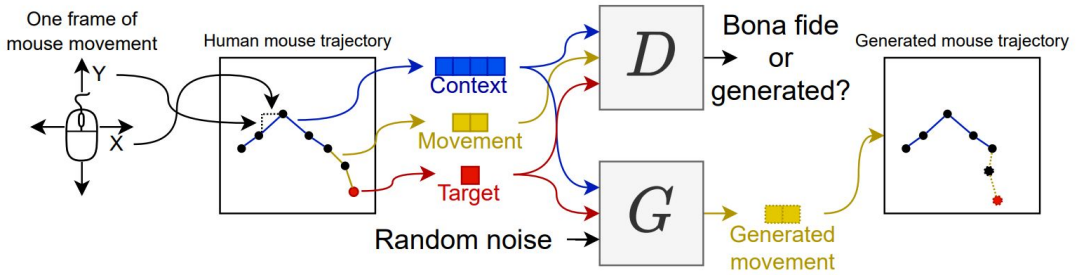


Fig. 2. An illustrative figure of the discriminator D and generator G networks and their inputs/outputs, with a context size of four and two movement steps. Note that "target" is an absolute location with respect to where the trajectory begins, while other values are one-step changes in location. This setup corresponds to training (Algorithm 1), where "target" is taken from human data.

Cheaty MLowe ciężkie do wykrycia

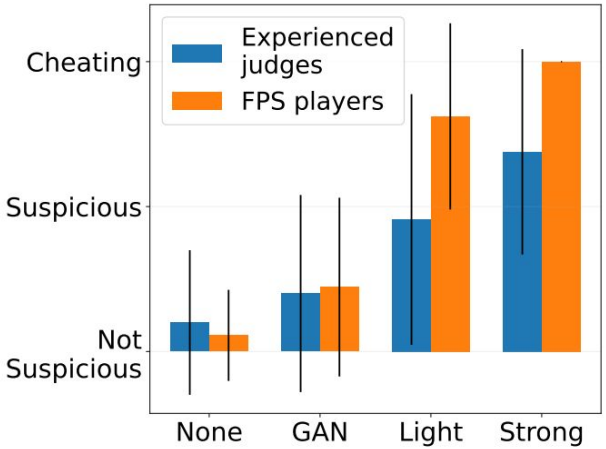


Fig. 8. Average grading (y-axis) given by human graders for different aimbots (x-axis), with a black line representing plus-minus one standard deviation, computed over 90 samples. For *strong* aimbot, all FPS player judges voted cheating, hence the standard deviation is zero.

TABLE VIII
RATIO OF GRADES PER AIMBOT PER GRADER GROUP, IN PERCENTAGES (%).

Experienced judges	**None**	**GAN**	**Light**	**Strong**
Not suspicious	84.4	71.1	42.2	13.3
Suspicious	11.1	17.8	24.4	35.6
Cheating	4.4	11.1	33.3	51.1
FPS players				
Not suspicious	88.9	62.2	8.9	0.0
Suspicious	11.1	31.1	20.0	0.0
Cheating	0.0	6.7	71.1	100.0

1) Not suspicious. I would not call this player a cheater.
2) Suspicious. I would ask for another opinion and/or monitor this player for a longer period of time to determine if they were truly cheating.
3) Definitely cheating. I would flag this player for cheating and use the given video clip as evidence.