Detekcja ogórków grunotwych w środoiwsku naturalnym przy użyciu algorytmu YOLO

Jan Tyc

April 2023

1 Wstep

Pomimo ogromnego udziału rolnictwa w gospodarce światowej oraz ciagłego zastepowania pracy ludzkiej przez maszyny, wciaż wiele prac zwiazanych z uprawami jest wykonywanych recznie. Wiele owoców i warzyw jest zbieranych przez maszyny rolniczne, które zapewniaja duża wydajność, szybkość i oszczedności zarówno w małych jak i dużych gospodarstwach. Ze wzgledu na specyfike upraw wiele gatunków wciaż zbieranych jest recznie z uwagi na brak odpowiedniej metody, która zebrałaby plony nie uszkadzajac ich oraz nie pozostawiłaby zniszczeń na polu. Dzieki gwałtownemu rozwojowi sztucznej inteligencji coraz cześciej podejmowane sa próby stworzenia maszyny do automatycznego i precyzyjnego zbioru, a w ciagu ostatnich kilku lat wydajność tych systemów uległa znacznej poprawie [3]. Oczywistymi wyzwaniami przy opracowywaniu takiego systemu wykrywania upraw sa niespójne oświetlenie, nierozróżnialne tło, zachodzace na siebie lub zasłoniete plony, czy niskie rozdzielczości zdjeć na których przeprowadzana jest detekcja obiektów [4, 6, 7]. W poniższym raporcia postaramy sie dokonać szerokiego przegladu literatury w temacie optymalizacji algorytmu YOLO V7 oraz ogólnego problemu detekcji owoców i warzyw w środowisku naturalnym.
Dalsza cześć raportu jest ułożona w nastepujacy sposób: rozdział 2 dotyczy objaśnienia działania rodziny algorytmów YOLO (na przykładzie YOLO V7),w rozdziale 3 zawarte sa informacje na temat danych użytych do trenowania algorytmu, rozdział 4 dotyczy przegladu literatury i potencjalnych rozwiazań wprowadzonych w algorytmach detekcji obiektów w celu ich optymalizacji, rozdział 5 to podsumowanie zgromadzonej wiedzy i prezentacja planu dalszej cześci badań.

2 Wstep do YOLO

YOLO (You Only Look Once) to rodzina algorytmów detekcji obiektów, która chrakteryzuje sie obecnie najlepszymi metrykami dokładności wykrywania obiektów oraz bardzo duża szybkościa przetwarzania obrazu [5]. Najnowsze odsłony sa

w stanie zwracać położenia obiektów na obrazie w tempie przekraczajacym 100 FPS co sprawia, że algorytm może być wykorzystywany do przetwarzania danych w czasie rzeczysiwtym [1].
Algorytm składa sie z trzech cześci: "Ogona" (eng. Backbone), "Szyi" (eng. Neck) oraz "Głowy" (eng. Head)

2.1 "Ogon"

Pierwsza cześcia każdego algorytmu YOLO jest "Ogon". Jest to pierwsza warstwa do której wprowadzany jest obraz, który nastepnie przechodzi przez wiele warstw konwolucji, normalizacji batchy, funkcji aktywacji (Relu, Leaky Relu, Mish, Silu), Maxpoolingu czy konkatenacji wcześniejszych warstw. W ten sposób z obrazu ekstrachowane sa cechy, które nastepnie beda dalej przewarzane i posłuża do detekcji obiektu. Na przestrzeni rozwoju algorytmu stosowano "ogony" o różnej architekturze (Darknet24, CSPDarknet53 ,Modified CSP v7 [5]) natomiast w wersji YOLO V7 zaproponowano sieć E-ELAN [1], która rozwiazuje problem zanikajacej ścieżki gradientu, zachowujac przy tym odpowiednia złożoność obliczeniowa.

2.2 "Szyja"

Po wstepnym przetworzeniu dane przechodza do "Szyi" gdzie beda dalej przetwarzane. Architektura "Szyi" różni sie znaczaco w zależności od wersji YOLO. W YOLO V7 dane przechodza przez moduł SPPCSP oraz przez warstwy konwolucji i konkatenacji wcześniejszych warstw.

2.3 "Głowa"

Algorytm działa w oparciu o nastepujace cztery podejścia:

1. Podział obrazu na sekcje
2. Rysowanie 'bounding boxes'
3. Wyliczenie IOU
4. Non-maximum-suppresion

2.3.1 Podzial na sekcje

Rozpoczynamy od podzielenia oryginalnego obrazu na komórki siatki NxN o jednakowym kształcie. Każda komórka siatki jest odpowiedzialna za zlokalizowanie i przewidzenie klasy obiektu, który obejmuje, wraz z wartościa prawdopodobieństwa/pewności.

2.3.2 Rysowanie 'bounding boxes'

Kolejnym krokiem jest wyznaczenie bounding boxów, które odpowiadaja prostokatom zaznaczajacym wszystkie obiekty na obrazie. Możemy mieć tyle bounding boxów, ile jest obiektów w obrebie danego obrazu. YOLO określa atrybuty tych pól ograniczajacych za pomoca pojedynczego modułu regresji w nastepujacym formacie, gdzie Y jest ostateczna reprezentacja wektorowa dla każdego pola ograniczajacego. Y = [bx, by, bh, bw, c1, c2, ..., cn] gdzie:

- bx, by - koordynaty środka prostokata,
- bh, bw - wysokość i szerokość prostokata,
- c1, ..., cn - prawdopodobieństo że w prostokacie znajduje sie obiekt danej klasy

2.3.3 IOU

W wiekszości przypadków, pojedynczy obiekt na obrazie może mieć wiele kandydatów do przewidywania, nawet jeśli nie wszystkie z nich sa istotne. Celem IOU (wartość miedzy 0 a 1) jest odrzucenie takich prostokatów, aby zachować tylko te, które sa istotne.

2.3.4 Non-max-suppression

Ustawienie progu dla IOU nie zawsze jest wystarczajace, ponieważ obiekt może mieć wiele pól z IOU poza progiem, a pozostawienie wszystkich tych pól może zawierać szum. Tutaj możemy użyć NMS, aby zachować tylko pudełka z najwyższym wynikiem prawdopodobieństwa wykrycia.

3 Dane użyte do treningu

Zbiór danych tworza 594 zdjecia ogórków gruntowych w środowisku naturalnym. Zdjecia zostały zrobione w wielkopolsce w Lipcu 2022 r. na dwóch różnych polach. Do ich zrobienia użyto aparatu z telefonu Xiaomi Redmi Note 8 Pro, a rozdzielczość każdego zdjecia to 4624x3472. Dane zostały oetykietowane tzn. na każdym zdjecia zostały zaznaczone ogórki, które nastepnie posłuża jako 'ground truth' w uczeniu algorytmu.

4 Przeglad literatury

Odpowiednio szybka i dokładna detekcja owoców i warzyw w środowisku naturalnym jest kluczowym elementem w stworzeniu automatycznej i precyzyjnej maszyny do zbioru plonów. Problemy zwiazane z detekcja obiektów możemy podzielić na dwie klasy:

- Zwiazane ze zdjeciami na których bedzie przeprowadzana detekcja (dane)

- Zwiazane z algorytmem oraz jego treningiem

W dalszej cześci przyjrzymy sie jakie najwieksze wyzwania czekaja nas w każdej z klas oraz jakie ich rozwiazania sa obecnie dostepne.

4.1 Problemy z danymi potrzebnymi do detekcji

Algorytmy detekcji ucza sie na zdjeciach na których zaznaczone sa położenia obiektów, które wytrenowany algorytm powienien zlokalizować. W przypadku błednego lub niedokładnego zaznaczenia danych treningowych algorytm bedzie sie źle uczył i nigdy nie osiagnie zadawalajacej skuteczności detekcji, dlatego szczególnie ważnym jest dbanie o wysoka jakość zbioru uczacego. W przypadku małej ilości danych, moga one zostać poddane augmentacji, czyli zmienie zdjecia np. poprzez obrót, zmiane wielkości, odbicie lustrzane, zmiane saturacji i jasności zdjecia, w celu zwiekszenia ilości próbek uczenia [2].

4.2 Problemy zwiazane z algorytmami detekcji obiektów

Powstanie maszyny do automatycznego zbioru jest dyktowane szybkościa z jaka algorytm jest w stanie znajdywać warzywa i owoce na zdjeciach z kamery. Jeśli jedno zdjecie jest przetwarzane zbyt długo to maszyna nie jest wydajna, dlatego detekcja obiektów musi odbywać sie w tempie przynajmniej 30-40 FPS, żeby móc konkurować z człowiekiem. Algorytm podczas uczenia może wpadać w minima lokalne, przez co może nie osiagać zadawalajacych rezultatów i tracić możliwość generalizacji, dlatego bardzo ważny jest odpowiedni dobór architektury i hiperparametrów uczenia. Sposób podawania zdjeć podczas treningu też ma duży wpływ na metryki uczenia. "Curriculum learning" to technika polegajaca na stopniowym podawaniu do uczenia coraz to trudniejszych przypadków w celu powolnej i głebszej generalizacji algorytmu.

4.3 Inne podejścia problemu detekcji

Stosuje sie również inne techniki detekcji owoców i warzyw, nieoparte na uczeniu głebokim [3]. Detekcja za pomoca koloru to podejście, które znalazło zastosowanie w detekcji warzyw i owoców o kontrastowym kolorze (w odniesieniu do otoczenia) i powstały już pierwsze maszyny do zbioru, które obecnie sa w fazie testów [8]. Stosuje sie również detekcje za pomoca tekstury oraz detekcje kształtu na obrazie [3].

5 Podsumowanie i plan dalszych prac i eksperymetów

Algorytm YOLO V7, zaprezentowany we wcześniejszej cześci raportu, jest obecnie najlepszym dostepnym algorytmem do detekcji obiektów, który oferuje najlepsze rezutaty dla szybkości od 5 do 160 FPS. Zmiana struktury warstw algorytmu, może okazać sie nieskuteczna z uwagi na to, że architektura algorytmu została starannie dobrana przez twórców i wszelkie zmiany moga nieznacznie

pogarszać metryki. Wobec tego najlepszym podejściem w przypadku optymalizacji detekcji może okazać sie optymalizacja samego etapu treningu sieci. Dobranie odpowiednich hiperparametrów uczenia takich jak liczba epok, learning rate, decay itp. może znaczaco przyczynić sie do zwiekszenia poprawności detekcji. W przypadku problemu detekcji małych obiektów, przetestowana zostanie metoda rozpoznawania małych ogórków na podstawie detekcji koloru żółtych kwiatów. Nienznacznie zwiekszajac długość działania algorytmu, możemy znacznie poprawić metryki detekcji dla małych obiektów dla tego konkretnego przypadku. Z uwagi na mała liczbe danych zostanie przeprowadzona augemtacja danych w celu zwiekszenia ich ilości i poprawienia generalizacji modelu. Żeby zapobiec problemowi cieżko-rozpoznawalnych przypadków zostanie przetestowana metoda oparta na Curriculum Learning tzn. na poczatku podawania do sieci łatwych przypadków detekcji (ogórek w całości widoczny, dobrze oświetlony), a nastepnie po wstepnym uczeniu, douczanie sieci trudniejszymi przypadkami (ogórki nachodzace na siebie, cześciowo zasłoniete, z kiepskim oświetleniem) zapewnia wieksza generalizacje uczenia.
Pomimo wielu trudności zwiazanych ze stworzeniem maszyny do automatycznego zbioru owoców i warzyw, postep technologiczny i osiagniecia w dziedzinach detekcji obiektów i przetwarzania obrazów pozwalaja zachować optymizm w dalszych pracach nad rozwojem tego rozwiazania. Stworzenie takiej maszyny może znaczaco zmienić rynek rolniczy na świecie, zmniejszyć koszta ponoszone przez producentów żywności co może bezpośrednio przełożyć sie na spadek cen żywności na świecie.

6 Bibliografia

References

[1] YOLOv7: Trainable bag-of-freebies sets new state-of-the-art for real-time object detectors, Chien-Yao Wang and Alexey Bochkovskiy and Hong-Yuan Mark Liao, 2022, 2207.02696, arXiv, cs.CV

[2] R. Fernández, H. Montes, J. Surdilovic, D. Surdilovic, P. Gonzalez-De-Santos and M. Armada, "Automatic Detection of Field-Grown Cucumbers for Robotic Harvesting," in IEEE Access, vol. 6, pp. 35512-35527, 2018, doi: 10.1109/ACCESS.2018.2851376.

[3] Xiao, F.; Wang, H.; Li, Y.; Cao, Y.; Lv, X.; Xu, G. Object Detection and Recognition Techniques Based on Digital Image Processing and Traditional Machine Learning for Fruit and Vegetable Harvesting Robots: An Overview and Review, Agronomy 2023, 13,639, https://doi.org/10.3390/agronomy13030639

[4] Gai Rongli, Chen Na, Yuan Hai, A detection algorithm for cherry fruits based on the improved YOLO-v4 model, 2021/05/26, Neural Computing and Applications, https://doi.org/10.1007/s00521-021-06029-z

[5] Diwan, T., Anirudh, G. Tembhurne, J.V. Object detection using YOLO: challenges, architectural successors, datasets and applications. Multimed Tools Appl 82, 9243–9275 (2023). https://doi.org/10.1007/s11042-022-13644-y

[6] Lawal, Mubashiru Olarewaju. "Tomato detection based on modified YOLOv3 framework." Scientific Reports 11.1 (2021): 1-11.

[7] Liu, Guoxu, et al. "YOLO-tomato: A robust algorithm for tomato detection based on YOLOv3." Sensors 20.7, 2020

[8] Muhammad Hammad Malik, Ting Zhang, Han Li, Man Zhang, Sana Shabbir, Ahmed Saeed, Mature Tomato Fruit Detection Algorithm Based on improved HSV and Watershed Algorithm, IFAC-PapersOnLine, Volume 51, Issue 17, 2018, Pages 431-436, ISSN 2405-8963, https://doi.org/10.1016/j.ifacol.2018.08.183.