JOURNAL of COMPUTER SCIENCES INSTITUTE

JCSI 17 (2020) 401–404
Received: 22 September 2020
Accepted: 5 October 2020

Faster R-CNN model learning on synthetic images

Model Faster R-CNN uczony na syntetycznych obrazach

Błażej Łach*, Edyta Łukasik

Department of Computer Science, Lublin University of Technology, ul. Nadbystrzycka 38, 20-618 Lublin, Poland

Abstract

Machine learning requires a human description of the data. The manual dataset description is very time consuming. In this article was examined how the model learns from artificially created images, with the least human participation in describing the data. It was checked the effect of augmentation and progressive image size when training model on a synthetic dataset. The model has achieve up to 3.35% higher mean average precision on syntetic dataset in the training with increasing images resolution. Augmentations improved the quality of detection on real photos. The production of artificially generated training data has a great impact on the acceleration of prepare training, because it does not require as much human resources as normal learning process.

Keywords: computer vision; synthetic images; Faster R-CNN; deep learning

Streszczenie

Uczenie maszynowe wymaga opisu danych przez człowieka. Opisywanie zbioru danych ręcznie jest bardzo czasochłonne. W artykule zbadano jak model uczył się na zdjęciach sztucznie wytworzonych, z jak najmniejszym udziałem człowieka przy opisywaniu danych. Sprawdzono jaki wpływ miało zastosowanie augmentacji i progresywnego rozmiaru zdjęcia przy treningu modelu na syntetycznym zbiorze. Model osiągnął nawet o 3,35% wyższą średnią precyzję na syntetycznym zbiorze danych przy zastosowaniu treningów z rosnącą rozdzielczością. Augmentacje poprawiły jakość detekcji na rzeczywistych zdjęciach. Wytwarzanie sztucznie danych treningowych ma duży wpływ na przyśpieszenie przygotowania treningów, ponieważ nie wymaga tak dużych nakładów ludzkich, jak klasyczne uczenie modeli z danymi opisanymi przez człowieka.

Słowa kluczowe: computer vision; sztuczne obrazy; Faster R-CNN; głębokie uczenie

1. Wprowadzneie

Trenując modele sieci neuronowych do nauki potrzeba danych z opisem. Zbiór danych oznaczją zwykle ludzie, wykorzystując swoje zmysły poznawcze, zaznaczają obiekty widoczne na zdjęciach. Deskrypcji danych zwykle trzeba poświęcić wiele godzin, aby dokładnie scharakteryzować większą liczbę obrazów. Wykorzystując komputery w prosty sposób można wygenerować zbiór danych syntetycznych, poprzez nanoszenie wzorców na zdjęcia tła i zapisywanie cech obiektów.

Oznaczając jakość modelu przy detekcji obiektów stosuje się metrykę mean average precision (mAP). Metryka ta ocenia predykowanie położenia i klasyfikacji obiektów na zadanym zbiorze. Podaczas wyliczania mAP stosowana jest skala od 0 do 1. Jednak warto zaznaczyć, że w pracach z dziedziny rozpoznawania obiektów, przedmiotowa metryka jest najczęściej podawana w procentach, co zostało zastosowane również w ninejszej pracy.

*Corresponding author
Email address: blazej.lach@pollub.edu.pl (B. Łach)

Powszechnie wiadomo, że przy danych opisywanych przez człowieka, modele są w stanie nauczyć się rozpoznawać obiekty na zdjęciach. Twórcy Faster R-CNN nauczyli wykrywać obiekty na podstawie zdjęć ze zbioru COCO i Pascal VOC, poprawili wynik osiągnięty przez Fast R-CNN o 2,2% mAP na zbiorze COCO [1,2]. Wykorzystując Faster R-CNN nauczono model wykrywać twarze ludzi z dość dużą dokładnością [3].

Augmentacje są to przekształcenia obrazu takie jak: skalowanie, obracanie, nadawanie perspektywy, rozmazanie, zmiana kolorów, wykorzystywane do modyfikacji danych treningowych. Dzięki przekształceniom obrazów treningowych można powiększać zbiór danych lub zwiększać różnorodność cech obiektów występujących w zbiorze. Bezsprzecznie można wykorzystać augmentacje do poprawienia wyników detekcji na zdjęciach rzeczywistych. Stosując grupy przekształceń zdjęć do prawdziwych obrazów, można zwiększyć wykrywalność obiektów na zbiorze COCO o 2,3% mAP i o 2,7% mAP na zbiorze Pascal VOC [4]. Ponadto wykorzystując grupy augmentacji można uzyskać większy przyrost mAP na mniejszych zbiorach danych [4]. Augmentacje

wspomagają również klasyfikatory obrazów. Dzięki stosowaniu przekształceń zdjęć poprawiono wyniki precyzji na zbiorach ImageNet o 0,4% i CIFAR-10 o 0,6% przy klasyfikacji obiektów [5]. Mając na uwadze powyższe, można zauważyć jak bardzo istotne są augmentacje w procesie uczenia modeli.

Przeglądając prace innych autorów dotyczące tego tematu można zauważyć, że większości sytuacji syntetyki wykonywane są pod ściśle określony przypadek, niewykorzystywany do detekcji obiektów. Udało się nauczyć model określać pozycje ciała człowieka na podstawie nacisku na materac. Model ten trenowany był na sztucznie wytwarzanych obrazach nacisku generowanych z obiektów 3D, oddziaływujących ze sobą [6]. Wytrenowano również model rozpoznający głębię przedmiotów z wykorzystaniem masek obiektów, stosując Mask R-CNN. Do treningu wykorzystano sztucznie wygenerowany zbiór danych, stworzony z obiektów 3D umieszczonych w pudełku [7]. Nauczono również model sterować ramieniem robota na podstawie sztucznego środowiska 3D. Wykorzystując augmentacje obrazów syntetycznego środowiska zmniejszono błąd położenia kostki, którą robot miał przenosić. Model sprawdził się przy sterowaniu rzeczywistym ramieniem dzięki nauce na sztucznym środowisku [8].

Na podstawie przeprowadzonego przeglądu dokonań innych autorów można, zauważyć, że nikt nie przeprowadzał eksperymentów uczenia modelu do rozpoznawania syntetyków bez wykorzystywania obiektów 3D. Według tych informacji wyznaczono następujące cele badawcze. Sprawdzono, czy model jest w stanie nauczyć się rozpoznawać obiekty na zdjęciu przy treningu na obrazach sztucznie generowanych. Dodatkowo zweryfikowano, jakie znaczenie ma uczenie z rosnącym rozmiarem zdjęcia treningowego na detekcję obiektów. W badaniu sprawdzano również, jaki wpływ mają augmentacje syntetycznego zbioru treningowego na predykcje modelu.

Kolejna sekcja skupia się na wykorzystanych technologiach do przeprowadzenia badania. Zamieszczona będzie również charakterystyka maszyny wykorzystywanej do treningu i sposób prowadzenia treningów. W tym dziale opisana będzie również metoda badawcza. Następna sekcja przedstawiać będzie wyniki badania i ich interpretację. Ostatni dział będzie podsumowaniem przeprowadzonego badania.

2. Materiały i metody

Python był głównym narzędziem stosowanym przy przeprowadzaniu eksperymentu. Język ten jest szeroko wykorzystywany przy uczeniu maszynowym. Wybrano go głównie ze względu na dużą dostępność bibliotek wspomagających programowanie. Do badań korzystano z Python'a w wersji 3.7.9. Jako bibliotekę do prowadzenia treningów wykorzystano TensorFlow w wersji 1.15.2. Biblioteka ta współpracuje z wyżej wymienionym językiem. TenosrFlow zapewnia kod umożliwiający treningi, ewaluację i serwowanie predykcji modelu, przy użyciu do obliczeń CPU lub GPU [9]. Faster R-CNN jest szkieletem do detekcji obiektów. Został stworzony poprzez dodanie sieci proponującej regiony (RPN) do Fast R-CNN [1]. Szkielet ten jest zaimplementowany w TensorFlow. W badaniu korzystywano z Faster R-CNN, we współpracy z ResNet50, jako siecią ekstraktującą cechy. Albumentations jest szybką biblioteką stosowaną do nakładania augmentacji na zbiór danych [10]. Bibliotekę tą wykorzystano w wersji 0.4.6. Jest ona dedykowana do języka Python. Dzięki Albumentations możliwe jest nakładanie wybranych przekształceń na obrazy. Przy modyfikacji obrazów metamorfozie ulegają również opisy zdjęć. Tworząc zdjęcia syntetyczne wykorzystywano bibliotekę OpenCV. Przy jej pomocy wczytywano zdjęcia, nanoszono wzorce na zdjęcia tła i zapisywano zamiany do pliku. Biblioteka ta zaimplementowana w języku C++ współpracuje dzięki nakładce z Python'em i Albumentations. Wykorzystywano ją w wersji 4.4.0.42.

Ucząc sieci neuronowe ważna jest fizyczna maszyna wykorzystywana do treningów. Aby zapewnić szybkie obliczenia, wskazane jest wykorzystanie wydajnych podzespołów. Badania przeprowadzano na komputerze wyposażonym w 8 rdzeniowy procesor Intel i7 7700K z taktowaniem maksymalnym 4,5 GHz. Podczas badania modele miały do dyspozycji 16 GB pamięci RAM o częstotliwości 3600 MHz w dwóch modułach po 8 GB, pracujących w trybie dual channel. Wszystkie obliczenia były wykonywane na karcie graficznej Nvidia GTX1080Ti, posiadającej 11 GB pamięci VRAM. Taktowanie pamięci GPU wynosiło 11124 MHz, natomiast taktowanie rdzenia wynosiło 1683 MHz.

Trenując modele potrzeba dwóch zbiorów danych: treningowego i ewaluacyjnego. Sieć neuronowa wykorzystując syntetyki uczy się informacji o wykrywanych obiektach. Zbiór danych testowych wykorzystywany jest do sprawdzania, czy model nauczył się generalizować informacje o obiektach. Zbiory te przy rzeczywistych danych powinny pochodzić z różnych dystrybucji, aby w inny sposób ukazywać obiekty.

Obiektami do rozpoznania przez model były wybrane polskie znaki drogowe. Generując syntetyki nanoszono zdjęcia wzorców na obrazy tła. Obrazy tła zostały selekcjonowane w taki sposób, aby przedstawiały naturalny kontekst występowania obiektów. Na zdjęciach występowały głównie ulice, zadbano o to, by nie było na nich obiektów, które model miał rozpoznawać. Do przygotowania zbiorów wyselekcjonowano 40 zdjęć dla zbioru danych treningowych i 10 innych zdjęć dla zbioru danych testowych. Wszystkie obrazy tła przycięto lub przeskalowano do rozmiarów 1536 pikseli szerokości i 1536 pikseli wysokości. Łącząc wzorce i tła, nanoszono w losowej ilości i wielkości zdjęcie obiektu na obraz tła, zapisując położenie i nazwę klasy. Według powyższego sposobu wygenerowano 1000 obrazów dla zbioru treningowego i 100 obrazów dla zbioru testowego. W danych treningowych i ewaluacyjnych występują wszystkie obiekty które model miał rozpoznawać. Liczba obiektów dla klas

w zbiorze testowym wynosi od 20 do 34 wystąpień. Natomiast w danych treningowych jest to od 238 do 318 obiektów dla każdej klasy. Zbiór ewaluacyjny przygotowano stosując inne tła i skalowanie wzorców niż występujące w zbiorze treningowym. Wygenerowany zbiór treningowy będzie określany, jako zbiór pierwszy.

Stosując Albumentations stworzono przekształcone zbiory danych treningowych. Drugi zbiór wytworzono przez nałożenie przekształcenia IAAPerspective, uzyskując obiekty posiadające perspektywę. Trzeci zbiór stworzono wykorzystując przekształcenie ShiftScaleRotate, zmniejszając i obracając całe zdjęcia. Czwarty zbiór augmentowany był przy pomocy obu przekształceń, otrzymując wszystkie powyższe efekty zdjęć. Każde z przekształceń nakładano na zdjęcia treningowe z rozkładem równomiernym.

Do weryfikacji celów zaplanowano 8 treningów. Wykorzystując 4 zbiory przeprowadzono po jednym treningu na zbiór, w standardowy sposób. Po jednym zbiorze na trening zastosowano również do kolejnych 4 treningów, które wykorzystywały zwiększający się w trakcie rozmiar zdjęcia. Wszystkie treningi rozpoczynały naukę z tego samego punktu kontrolnego. Treningowy punkt kontrolny przechowuje dokładną wartością wszystkich patametrów używanych przez model. Rosnąca rozdzielczość oznacza uruchomienie treningu przy przeskalowanym zdjęciu wejściowym do 384x384 pikseli. Z takiego treningu wybierano najlepszy punkt kontrolny i kontynuowano go ze zwiększoną rozdzielczością do 768x786 pikseli. Ponownie wybierano najlepszy punkt kontrolny i kontynuowano go z rozdzielczością docelową 1536x1536 pikseli. Przy analizie wyników dla każdego z modeli będzie podane mAP na sztucznym zbiorze testowym. W celu sprawdzenia, czy modele potrafią rozpoznawać rzeczywiste obiekty, każdy model zostanie odpytany o predykcję na tym samym zdjęciu rzeczywistym.

3. Rezultaty i dyskusja

Wyniki mAP wszystkich modeli sprawdzane na sztucznym zbiorze testowym zestawiono w tabeli 1. Można zauważyć, że stosowanie wybranych w tym badaniu augmentacji pogarsza mAP na syntetycznym zbiorze testowym. Prawdopodobnie jest to spowodowanie zbyt małą różnorodnością cech obiektów w zestawie danych ewaluacyjnych. Zastosowanie treningu z rosnącą rozdzielczością przeważnie zwiększało mAP na sztucznym zbiorze testowym. Stosowanie rosnącej rozdzielczości poprawiło wynik mAP o 1,08% dla pierwszego zbioru danych. Rosnąca rozdzielczość wykorzystana podczas treningu poprawiła wynik o 1,98% mAP dla zbioru z augmentacjami ShiftScaleRotate. Jedynie użycie zestawu danych przekształcanych za pomocą IAAPerspective pogorszyło o 0,62% mAP przy zastosowaniu treningu z rosnącą rozdzielczością. Największy przyrost mAP odnotowano dla rosnącej rozdzielczości trenowanej na zbiorze łączonych augmentacji IAAPerspective, ShiftScaleRotate i wynosił 3,35% mAP. Zastosowanie zwiększającej się rozdzielczości prawdopodobnie nauczyło model lepiej generalizować informacje o obiektach ze zbioru syntetycznego.

Tabela 1: Wyniki modeli testowanych na zbiorze syntetycznym

Zastosowany trening	Wykorzystane augmentacje	mAP (%)
Standardowy	Brak	72,50
	IAAPerspective	70,04
	ShiftScaleRotate	69,21
	Obie	67,36
Rosnąca rozdzielczość	Brak	73,58
	IAAPerspective	69,42
	ShiftScaleRotate	71,19
	Obie	70,71

Rysunek 1: Predykcje na zdjęciu rzeczywistym modelu trenowanego standardowo na zbiorze bez augmentacji

Przeglądając predykcje modeli na prawdziwych zdjęciach można zauważyć, że treningi bez augmentacji wskazywały koło samochodu, jako znak ruch okrężny (Rysunek 1, 2). Warto zauważyć, że model trenowany ze zwiększającą rozdzielczością, z mniejszą pewnością wskazał koło samochodu, jako znak (Rysunek 2). Oba modele oznaczyły poprawnie znaki przejście dla pieszych i zakaz zatrzymywania się. Model uczony z zastosowaniem narastającego rozmiaru zdjęcia wskazał prawidłowo znak zakaz postoju, którego model trenowany w standardowy sposób nie oznaczył. Sieć trenowana z rosnącą rozdzielczością oznaczyła dodatkowo tabliczkę z nazwą ulicy (zasłoniętą przez opis), jako znak teren zabudowany z dużą pewnością.

Predykcja na zdjęciu rzeczywistym modelu wytrenowanego bez rosnącego rozmiaru zdjęcia na zbiorze z augmentacjami ShiftScaleRotate, jest jedną z najlepszych predykcji spośród wszystkich modeli (Rysunek 3). Model poprawnie i z wysoką pewnością oznaczył dobrze

Rysunek 2: Predykcje na zdjęciu rzeczywistym modelu trenowanego z rosnącą rozdzielczością na zbiorze bez augmentacji

widoczne znaki, dokładnie zaznaczając obwiednie obiektów. Sieć nie wskazała jedynie małych znaków przejście dla pieszych w centralnej części zdjęcia.

Rysunek 3: Predykcje na zdjęciu rzeczywistym modelu trenowanego standardowo na zbiorze z augmentacją ShiftScaleRotate

Poprawnie oznaczone obiekty na zdjęciach rzeczywistych, sugerują że można wytrenować model na zdjęciach syntetycznych, do rozpoznawania obiektów wystepujących na prawdziwych zdjęć. Przy zastosowaniu augmentacji można poprawić jakość detekcji na rzeczywistych zdjęciach. Błędnie wskazane obiekty na zdjęciach mogą oznaczać, że model nie wyuczył się generalizować wystarczająco dobrze informacji. Rozwiązaniem tego problemu mogło by być dodanie do zbioru syntetycznego obrazów podobnych do obiektów, które model ma rozpoznawać, przykładowo koła samochodów lub tablice z nazwami miast i ulic. Brak rozpoznawania małych obiektów na zdjęciach rzeczywistych sugeruje, brak występowania tak małych wielkości obiektów w zbiorze treningowym. Aby zwiększyć wykrywalność małych obiektów, należało by wygenerować zbiór syntetyczny zawierający mniejsze zdjęcia wzorców nanoszonych na obrazy treningowe.

4. Podsumowanie

Dzięki komputerom nie ma przszkód, aby wygenerować sztuczne zbiory danych wraz z opisem. Na takim zbiorze można wytrenować model do rozpoznawania obiektów na zdjęciach rzeczywistych. Wykorzystując augmentacje na sztucznym zbiorze jest możliwe poprawienie jakości predykcji. Do przebadania pozostaje wpływ na mAP zastosowania treningu z rosnącą rozdzielczością przy wykorzystaniu zbioru ze zdjęciami rzeczywistymi.

Literatura

[1] S. Ren et al., Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks, IEEE Transactions on Pattern Analysis and Machine Intelligence, t. 39, nr 6, s. 1137–1149, cze. 2017, https://doi.org/10.1109/TPAMI.2016.2577031.

[2] R. Girshick, Fast R-CNN, [W:] 2015 IEEE International Conference on Computer Vision (ICCV), 2015, https://doi.org/10.1109/ICCV.2015.169.

[3] H. Jiang, E. Learned-Miller, Face Detection with the Faster R-CNN, [W:] 2017 12th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2017), 2017, https://doi.org/10.1109/FG.2017.82.

[4] B. Zoph et al., Learning Data Augmentation Strategies for Object Detection, 2019, https://arxiv.org/pdf/1906.11172.pdf.

[5] E. Cubuk et al., AutoAugment: Learning Augmentation Strategies from Data, [W:] 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, https://doi.org/10.1109/CVPR.2019.00020.

[6] H. Clever et al., Bodies at Rest: 3D Human Pose and Shape Estimation from a Pressure Image using Synthetic Data, [W:] 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, https://doi.org/10.1109/CVPR42600.2020.00625.

[7] M. Danielczuk et al., Segmenting Unknown 3D Objects from Real Depth Images using Mask R-CNN Trained on Synthetic Data, [W:] 2019 International Conference on Robotics and Automation (ICRA), 2019, https://doi.org/10.1109/ICRA.2019.8793744.

[8] A. Pashevich et al., Learning to Augment Synthetic Images for Sim2Real Policy Transfer, [W:] 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2019, https://doi.org/10.1109/IROS40897.2019.8967622.

[9] M. Abadi et al., TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems, 2016, https://arxiv.org/pdf/1603.04467.pdf.

[10] A. Buslaev et al., Albumentations: fast and flexible image augmentations, Information, t. 11, nr 2, s. 125, luty 2020, https://doi.org/10.3390/info11020125.