



Detekce prvků webového uživatelského rozhraní s Faster R-CNN

Jiří Vyskočil¹

1 Úvod

Při navrhování nových uživatelských rozhraní (UI) se mohou naskytnout určité výzvy, např. během komunikace mezi designery a vývojáři, čemuž detekce prvků UI může pomoci. ImageCLEF DrawnUI challenge přichází s 2 soutěžními úlohami založené na detekci takových prvků: *Screenshot task* obsahující snímky obrazovky z webových stránek společně s chybnými anotacemi a *Wireframe task* pro detekci náčrtů uživatelských rozhraní.

Pro zlepšení detekce ve zmíněných úlohách byl použit jednoduchý algoritmus založený na hranovém detektoru pro filtraci nevhodných dat pro hluboké učení. Detekční neuronová síť založená na architektuře Faster R-CNN představené v práci Ren et al. (2015) byla použita společně s Feature Pyramid Network (FPN) od Lin et al. (2017) umožňující zvýšit detekční výkon díky zavedeným spojení na úrovni map příznaků. Modely byly předtrénované na COCO datasetu a jako páteří architektura byla zvolena ResNet-50 popsaná v článku He et al. (2016). Navržené řešení získalo **první místo** v obou úlohách soutěže. Zdrojové kódy jsou dostupné na adrese: <https://github.com/vyskocj/ImageCLEFdrawnUI2021>

2 Filtrování nevhodných dat

Jelikož součástí úlohy se snímky obrazovky mohou být nevhodná data, byl zaveden jednoduchý filtrační algoritmus. Záměrem je odstranění homogenních obrázků bez závislosti na intenzitě pixelů, tj. pixely obsahující podobné barvy. Z toho důvodu byl využit hranový detektor, jehož využití je popsáno podrobně v následujícím algoritmu 1, který lze jednoduše rozšířit také o odstranění nevhodných boxů z trénovací sady:

Algoritmus 1 Filtrování homogenních obrázků z trénovací sady.

Definuj prahové hodnoty T_{img}

pro každý obrázek :

Použij hranový detektor na celý obrázek a vypočti střední hodnotu μ_{img}

když $\mu_{img} \leq T_{img}$:

Vyřaď obrázek z trénovací sady

skonči **když**

skonči **pro**

Představený filtrační algoritmus s manuálně nastavenými prahovými hodnotami přináší zlepšení na validační sadě Screenshot tasku oproti modelu používající datové sady bez provedené filtrace. Zmíněné zlepšení je o **0,008** až **0,009** pro mAP (mean average precision) a mAR (mean average recall), kdy výsledná hodnota činí **0,412** pro mAP a **0,612** pro mAR.

¹ student doktorského studijního programu Aplikované vědy a informatika, obor Kybernetika, e-mail: vyskocj@kky.zcu.cz

3 Přizpůsobení detektoru

Silným nástrojem pro nevýhody spojené s hlubokým učení jsou augmentace. Kromě jasových byly použity i Cutout augmentace vyřezávající z obrázku náhodně až 4 boxy o velikosti do 5 % obrázku, a augmentace pro náhodnou změnu velikosti vstupu dle zadaného intervalu. Zadaný interval reprezentuje relativní hodnoty, o kolik se má obrázek zvětšit nebo zmenšit. V provedených experimentech vyšlo nejlépe použití intervalu od 0,6 do 1 pro obě řešené úlohy.

Poslední částí výzkumu bylo použití takových poměrů stran pro generování návrhů boxů (anchor box proposals), které nejlépe popisují distribuci anotovaných boxů v datasetu. Na základě analýzy byly vybrány následující poměry stran: 0,1, 0,5, 1 a 1,5. Oproti výchozím, tj. 0,5, 1 a 2, pomohly nové poměry stran zvýšit detekční výkon na validační sadě na hodnoty **0,426** mAP a **0,627** pro Screenshot task, a **0,704** mAP a **0,764** mAR pro Wireframe task

4 Vyhodnocení na testovací sadě

Nakonec byly porovnány páteřní architektury ResNet-50, ResNet-101 a ResNeXt-101. I přes očekávání, že složitější architektura bude dosahovat lepších výsledků se ukázalo, že pro Screenshot task tomu tak nebylo. ResNeXt-101 si vedla nejhůře pro tuhle úlohu, zatímco v druhé úloze, tj. Wireframe task, dosahovala prvnotřídních výsledků **0,706** mAP a **0,766** mAR. Modely byly vyhodnoceny podrobeny testovací sadě, výsledky nejlepších modelů oproti baseline stanovené organizátory soutěže jsou uvedeny v tabulce 1.

Tabulka 1: Porovnání metody na testovací sadě s baselinou stanovenou organizátory

Screenshot task			Wireframe task		
Páteřní architektura	mAP@0,5 IoU	mAR@0,5 IoU	Páteřní architektura	mAP@0,5 IoU	mAR@0,5 IoU
ResNet-50	0.628	0.830	ResNeXt-101	0.900	0.934
baseline	0.329	0.408	baseline	0.747	0.763

Poděkování

The work has been supported by the grant of the University of West Bohemia, project No. SGS-2019-027. Computational resources were supplied by the project "e-Infrastruktura CZ"(e-INFRA LM2018140) provided within the program Projects of Large Research, Development and Innovations Infrastructures.

Literatura

He, K., Zhang, X., Ren, S. and Sun, J. (2016), Deep residual learning for image recognition, in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 770-778.

Dollár, P., Girshick, R., He, K., Hariharan, B. and Belongie, S. (2017), Feature pyramid networks for object detection, in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 936–944.

Ren, S., He, K., Girshick, R. and Sun, J. (2015), Faster R-CNN: Towards real-time object detection with region proposal networks, in C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama and R. Garnett, eds, *Advances in Neural Information Processing Systems 28*, Curran Associates, Inc., pp. 91–99.