Ako uvidieť čokoľvek

Andrej Lúčny

Katedra aplikovanej informatiky, Fakulta matematiky, fyziky a informatiky, Univerzita Komenského
KAI FMFI UK, Mlynská dolina, 842 48 Bratislava
lucny@fmph.uniba.sk

Abstrakt

Umožniť robotovi uvidieť na obraze objekty patriace k určitej zvolenej sade kategórii je dnes už celkom možné. Výzvou sa stáva schopnosť uvidieť akýkoľvek objekt. Veľký pokrok v tejto oblasti priniesli vizuálne transformery, prvé stroje, ktoré vedia lokalizovať objekt na základe určenia jeho kategórie. Kľúčovú rolu však v poslednom čase zohrávajú aj inovatívne metódy ich trénovania z veľkého množstva neanotovaných dát, ako je metóda CUTLER. Štartovacím prostriedkom takéhoto trénovania sú klasické metódy na segmentáciu obrazu. Tieto sa opierajú o optimalizáciu rozdelenia obrazu na popredie a pozadie, čo je spravidla NP-ťažká úloha. Návrh algoritmov na efektívne hľadanie približného riešenia týchto úloh sa tým dostáva do hlavného prúdu výskumu v hlbokom učení a prináša zaujímavé výsledky.

1 Úvod

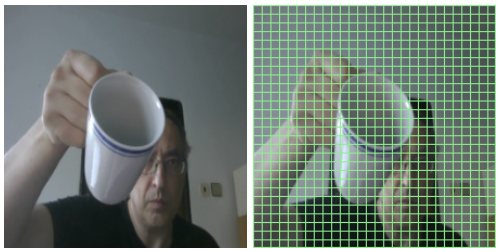
V tomto príspevku čitateľovi približujeme problematiku tvorby univerzálnych vizuálnych detektorov objektov, ktorou sa zaoberáme, ako aj náš skromný príspevok k nej. Detaily matematického a informatického charakteru pritom zväčša neuvádzame a to nielen pre nedostatok priestoru, ale za účelom priblíženia tejto problematiky celému spektru výskumníkov z oblasti kognitívnej vedy.

2 Vývoj vizuálnych detektorov

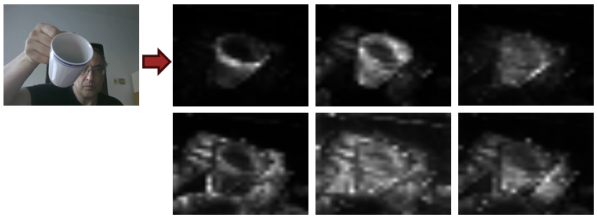
Prvé detektory objektov, ktorých ich kvalita robila reálne použiteľnými, boli vyvinuté v rokoch 2015 – 2016. Medzi priekopníkov patria You Only Look Once – YOLO vo verzii 1 (Redmon a spol., 2016), YOLO vo verzii 3 (Redmon a Farhadi, 2018), až verzia 12 (Tian a spol., 2025) a Faster Region-based Convolution Neural Network – FRCNN (Ren a spol., 2015), vo verzii MaskRCNN (He a spol., 2017). Oba spomenuté prístupy používajú v princípe hrubú silu grafických kariet s architektúrou CUDA. YOLO sa naraz pozerá na rôzne miesta na obrázku a robí klasifikáciu kategórie objektu, ktorý by mohol byť v skúmanej oblasti a regresiu veľkosti tohto objektu; výsledok vznikne potom pozliepaním najsľubnejších návrhov. RCNN pokrýva obraz veľkým množstvom možných obdĺžnikových regiónov, klasifikuje nakoľko by mohli obsahovať celý objekt a pre najsľubnejšie testuje, či vie klasifikovať objekt v danom regióne. Obe tieto siete sa trénujú na pevný zoznam kategórii možných objektov. Objekty iného druhu majú tendenciu vnímať ako objekt patriaci k niektorej podporovanej kategórii, avšak s veľmi nízkou dôveryhodnosťou.

Zásadný pokrok do metód detekcie priniesli vizuálne transformery (ViT). Hoci potrebujú ešte väčšiu výpočtovú kapacitu než prechádzajúce riešenia, nedívajú sa rôzne miesta na obraze, ale narežú obraz na malé políčka (Obr. 1) a sledujú asociácie medzi týmito políčkami plus ešte jedným pridaným, tzv. klasifikačným, ktorého úlohou je vychytať globálny význam obrazu). Význam každého políčka je reprezentovaný vektorom určitej zvolenej dimenzie, ktorá udáva koľkými príznakmi sa môžu políčka navzájom líšiť. Na začiatku je políčko premenené na takýto vektor (projekcia, ktorá to robí, sa nazýva vnorením – angl. embedding), potom sa niekoľko krát tieto reprezentácie spôsobom typickým pre transformer transformujú a na koniec sa na vektor klasifikačného políčka uplatňuje projekcia, ktorá vyhodnotí nakoľko sa podobá reprezentatom jednotlivých kategórii (táto projekcia sa nazýva vynorením, angl. wipe out), čím sa vyrátajú pravdepodobnosti príslušnosti obrazu ku každej z kategórii. Silnou vlastnosťou ViT-u je, že v priebehu klasifikácie toho, čo na obraze vidíme, sa ako vedľajší produkt vypočíta tzv. pozornostná mapa (Obr. 2), ktorá udáva, kde to vidíme – vieme ju odčítať z asociácii medzi políčkami a klasifikačným políčkom. ViT má šancu uvidieť objekt na obraze dokonca aj vtedy, keď je typ objektu určený nesprávne a do istej miery aj keď typ tohto objektu nie je medzi kategóriami, na ktoré sme ViT natrénovali.

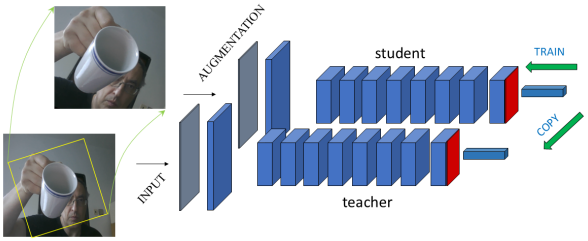
Každá neurónová sieť je však len tak kvalitná, ako sú kvalitné dáta, z ktorých bola natrénovaná. Preto je prirodzenou snahou trénovať univerzálny detektor z obrovského množstva dát. Súčasné technické prostriedky umožňujú zozbierať, uskladniť a používať pri trénovaní obrovské množstvo dát, napríklad všetky obrázky z Internetu. Problémom je získanie ich anotácií. Tie robia ľudia a tí sú príliš pomalí a drahí. Preto ďalším krokom k univerzálnemu detektoru je snaha potrebu anotácie



Obr. 1: Pokrytie obrázka políčkami typickými pre spracovanie obrazu vizuálnym transformerom.



Obr. 2: Pozornostná mapa vizuálneho transformeru.



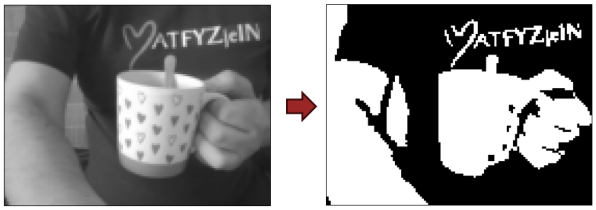
Obr. 3: Trénovanie modelu DINO.

obísť. Takýto prístup viedol k modelu DINO (DIstilation with NO labels), viď Caron a spol. (2021). Ten spracúva obraz a vracia príznakový vektor (vektor reprezentujúci klasifikačné políčko; na rozdiel od bežného ViT-u, DINO nemá projekciu realizujúcu vynorenie). V najmenšej verzii má tento vektor púhych 384 príznakov. Tento vektor nehovorí o akú kategóriu objektu ide, avšak vektory obrázkov objektov rovnakého typu sú podobné, zatiaľ čo rôzneho typu odlišné. Tento model je trénovaný (Obr. 3) tak, že udržiavame dve kópie modelu: učiteľa a študenta. Parametre učiteľa sa nemenia, upravujeme len parametre študenta. Do oboch privádzame vstup pochádzajúci z rovnakého obrázka, avšak vstup do študenta je náhodne pootočený, orezaný a zväčšený. Študenta upravujeme tak, aby dával rovnaký výstup ako učiteľ. Takto predložíme študentovi obrovské množstvo obrázkov mnoho krát a potom ho urobíme učiteľom a podľa neho trénujeme nového študenta. Výsledkom opakovania takéhoto procesu s obrovskou dátovou sadou obrázkov je, že sa model naučí rozlišovať medzi všetkými možnými objektami. Hoci DINO nepovie jasne o aký objekt ide (vráti len príznakový vektor objektu), je z neho možné vytiahnuť pozornostné mapy, ktoré hovoria, kde sa objekt nachádza.

Ako vidno na Obr. 2 pozornostná mapa je skôr machuľou, než že by udávala presné hranice objektu. Pre určité úlohy, napríklad aby sa robot otáčal za ukazovaným objektom, stačí vypočítať ťažisko mapy, takže je priamo použiteľná. Nestačí to však na určenie obdĺžnika v ktorom sa objekt nachádza. Na to je potrebné pozornostnú mapu binarizovať. Presne to využíva model CUTLER (ClUsTer with LEaRning alebo aj CUT and LEaRn), viď Wang a spol. (2023), poskytujúci masky ktorý opakovane trénuje sieť podobnú MaskRCNN veľkého množstva automaticky anotovaných obrázkov a poskytuje lepšiu kvalitu i rozlíšenie masiek (Obr. 4)

Obr. 4: Vývoj kvality masiek univerzálnych detektorov objektov.

CUTLER Vezme model DINO a z výstupu poslednej transformačnej vrstvy ViTu v ňom (pred koncovou hlavou) vytiahne príznaky pre každé políčko obrázka. Na ne aplikuje binarizačnú metódu NCut (Normalized Cut) – pozri Shi a Malik (2000), ktorá rozdeľuje políčka do dvoch skupín – „popredia“ a „pozadia“ (Obr. 5).



Obr. 5: Normalized Cut (NCut).

Takýmto spôsobom získame približné anotácie celého obrovského množstva obrázkov. Potom z nich trénujeme detektor klasickým spôsobom – na základe požadovaných výstupov (ktoré sme však nemuseli vyrobiť ručne). S tou výnimkou, že pri chybovej funkcii nesmie vadiť, že nejaký objekt vidí a v požadovanom výstupe pritom nie je. Trénovanie teda počíta s tým, že anotácie niektoré objekty nezachycujú, avšak model sa zdokonaľuje práve v tom, že uvidí aj ďalšie objekty. Po natrénovaní detektora (ktorý v tomto prípade dáva nielen obdĺžnik aj masky), prehodnotíme anotácie tak, že miesto pôvodných anotácií použijeme jeho výstupy. A natrénujeme nový detektor. Toto opakujeme, pričom postupne pribúda stále viac detekovaných objektov, až kým sme spokojní s výsledkom. A vskutku takýto detektor vidí „čokoľvek“. Kvalita je tu však limitovaná presnosťou rozdelenia obrazu na políčka. Delenie základného modelu DINO na 28×28 je pomerne hrubé.

Samozrejme neurónová sieť vie ísť na lepšiu presnosť tým, že z týchto hrubých dát trénuje združovaciu hlavu (pooler) ktorá má o niečo lepšie rozlíšenie, avšak v súčasných modeloch ani takto nedosahujeme pixelovú presnosť.

3 Súvis a NP-ťažkými problémami

Rozlíšenie masiek v anotáciách či výstupoch neurónových sietí má jednu zaujímavú súvislosť s teóriu vypočítateľnosti. Ako sme už spomenuli, anotovanie veľkého počtu obrázkov musí byť založené na nejakej automatickej metóde. Tvorcovia modelu CUTLER použili ako metódu na štartovaciu anotáciu NCut. NCut hľadá bipartíciu políčok (prípadne až pixelov, je to všeobecná metóda), pri ktorej dosahuje minimum určitá miera vyjadrujúca kvalitu rozdelenia obrazu do dvoch častí.

$$NCut(A,B) = \frac{Cut(A,B)}{Cut(A,A\cup B)} + \frac{Cut(A,B)}{Cut(B,A\cup B)} \quad (1)$$

$$Cut(A,B) = \sum_{p\in A, q\in B} w(p,q) \quad (2)$$

Je založená na miere podobnosti políčok (alebo pixelov) w, ktorá zvažuje podobnosť príznakov (alebo farby či intenzity) a vzdialenosti políčok na obraze. Musíme ju zadefinovať tak, aby dávala číslo od 0 (úplne nepodobné) po 1 (zhodné). To sa dá ľahko zariadiť, pre intenzitný vstup (bez uvažovania pozície) autori navrhli:

$$w(p,q) = e^{\frac{-(I(p)-I(q))^2}{\sigma_I^2}} \text{ kde } p,q \in \langle 0,1\rangle \quad (3)$$

avšak pokiaľ ju aplikujeme na príznakové vektory na výstupe transformeru stačí uvažovať kosínusovú podobnosť:

$$w(p,q) = \frac{pq}{\|p\|\|q\|} \quad (4)$$

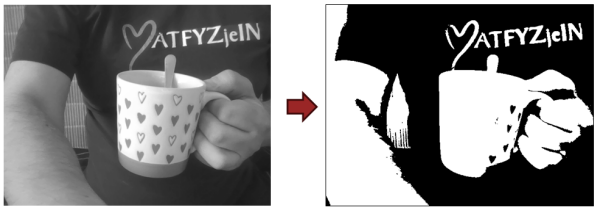
Zadefinovať problém je relatívne ľahké, avšak nájdenie NCut-u (a platí to v podstate pre akúkoľvek podobnú mieru) je NP-ťažký problém (von Luxburg, 2007). To znamená, že preň nepoznáme lepší algoritmus, než prejsť všetky možné rozdelenia pixelov, pre každé spočítať minimalizovanú mieru a zapamätať si rozdelenie, pre ktoré je minimálna. A teda pre n pixelov potrebujeme rádovo 2n operácií. Už pre to biedne rozlíšenie z DINO modelu 28×28 sa pohybujeme v rádoch 1036 a pre obrázok s bežným rozlíšením 1300×800 by sme sa výsledku nedobrali do konca sveta.

Shi a Malik (2000) preto navrhli približný algoritmus, založený na hľadaní vlastného vektora zodpovedajúceho druhej najmenšej vlastnej hodnote určitej matice rozmerov n×n, čo sa dá urobiť v čase rádovo tn2, kde n je počet pixelov (n = wh, kde w je šírka a h výška obrázka s rozlíšením w×h) a t počet iterácii (typicky t 550). To je algoritmus, ktorý pre 28×28 je normálne použiteľný, pre 128×128 už však aj na grafickej karte beží 15s a pri väčšom rozlíšení spadne na vyčerpaní pamäte GPU. Pre trénovanie modelu CUTLER bol dostatočný. Ak by sme však celý proces chceli zopakovať v lepšom rozlíšení, dostali by sme sa do vážnych problémov. Na zlepšenie tejto presnosti je preto kľúčový časovo a hlavne pamäťovo efektívnejší približný algoritmus pre výpočet NCut-u.

4 Náš príspevok k tejto problematike

Zaoberali sme sa preto možnosťou ako algoritmus na výpočet NCut-u zrýchliť. No a našli sme takú mieru podobnosti políčok či pixelov, pre ktorú nepotrebujeme vyčíslovať maticu n×n a celý algoritmus (upravenú verziu by sme mohli nazvať FastNCut) vieme spočítať v čase rádovo tn, kde n je počet pixelov a t počet iterácii (typicky t=4). Zrýchlenie je zásadné, pre obrázok 1300x800 dosahujeme čas 0.005s (Obr. 6). Detaily algoritmu presahujú rozsah tohto článku, na porovnanie uvádzame, že pre intenzitný vstup pracujeme s mierou podobnosti

$$w(p,q) = \cos(p-q) \text{ kde } p,q \in \langle 0,\pi/2\rangle \quad (5)$$



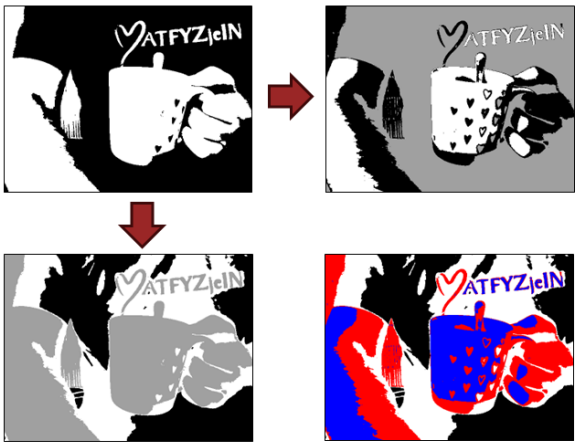
Obr. 6: Výstup algoritmu FastNCut, rozdelenie podľa intenzity (porovnaj s Obr. 4).

Podobne ako tvorcovia CUTLERa volali NCut opakovane a teda časti obrázka získané rozdelením ďalej delili na podčasti (pričom tento postup nazvali MaskCut), to isté sme po určitom elaborovaní zvládli aj s našou verziou algoritmu, viď Obr. 7.

Pomocou zrýchleného algoritmu sa v súčasnosti pokúšame o vytvorenie všeobecného detektora lepšej kvality.

5 Záver

V našom príspevku sme zachytili vývoj vizuálnych detektorov. Náš výklad sme orientovali na univerzálne detektory, ktoré nepracujú s objektmi z určitej sady kategórií, ale s akýmikoľvek objektmi. Poukázali sme na to, že pokrok v tejto oblasti ide smerom k nezávislosti od anotovaných dát. Kľúčovú úlohu v tomto procese



Obr. 7: FastNcut opakovaný na rozdelené časti, napodobňujúc metódu MaskCut.

zohrávajú vizuálne transformery, samo-učiace sa modely ako aj klasické algoritmy, menovite približné riešenia určitých NP-ťažkých problémov. To sme aj demonštrovali na konkrétnej snahe zlepšiť metódu CUTLER pomocou efektívnejšieho algoritmu na výpočet NCut-u. Tento príklad je pozoruhodný aj tým, že riešenia, na ktorých očami nevidíme chybu, dostávame – vzhľadom na to, že ide o NP-ťažký problém – neuveriteľne rýchlo (hoci o tom, že naozaj chybu nemajú by sme sa nepresvedčili do konca sveta), čo je zaujímavé aj hľadiska samotnej problematiky NP-ťažkých úloh.

Poďakovanie

Tento článok vznikol za podpory grantov VEGA 1/0373/23 a KEGA K-23-003-00.

Literatúra

Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P. a Joulin, A. (2021). Emerging properties in self-supervised vision transformers. V *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, str. 9630–9640.

He, K., Gkioxari, G., Dollár, P. a Girshick, R. (2017). Mask r-cnn. V *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, str. 2961–2969.

Redmon, J., Divvala, S., Girshick, R. a Farhadi, A. (2016). You only look once: Unified, real-time object detection. V *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, str. 779–788.

Redmon, J. a Farhadi, A. (2018). YOLOv3: An incremental improvement. *arXiv preprint arXiv:1804.02767*.

Ren, S., He, K., Girshick, R. a Sun, J. (2015). Faster r-cnn: Towards real-time object detection with region proposal networks. V *Advances in Neural Information Processing Systems*, str. 91–99.

Shi, J. a Malik, J. (2000). Normalized cuts and image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):888–905.

Tian, Y., Ye, Q. a Doermann, D. (2025). YOLOv12: Attention-centric real-time object detectors. *arXiv preprint arXiv:2502.12524*.

von Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416.

Wagner, D. a Wagner, F. (1993). Between min cut and graph bisection. V *Mathematical Foundations of Computer Science 1993*, str. 744–750.

Wang, K.-H., Zhang, R. a Isola, P. (2023). Cutler: Cut and learn for unsupervised object detection and segmentation. *arXiv preprint arXiv:2301.11320*.