

KNN — Identifikace osob podle obličeje

xnekut00, xletan00, xplach08

květen 2021

1 Úvod

Tato práce se zabývá verifikací osob podle obličeje s nasazenou rouškou pomocí hluboké konvoluční neuronové sítě. Přesněji řečeno jde o určení, zda se jedná o tutéž osobu na základě dvojice fotek obličeje, kdy na každé z těchto fotek může mít osoba nasazenou roušku. Z důvodu časové náročnosti trénování rozpoznávání obličejů jsme použili již předtrénovanou síť *FaceNet*[4], kterou jsme upravili a dotrénovali pro tento specifický problém.

Abychom mohli síť dotrénovat a vyhodnotit, upravili jsme datasety *Casia Web Face* (pro trénování) a *Labeled Faces in the Wild* (pro vyhodnocení) tak, že jsme na obličeje uměle umístili roušky. Na upraveném datasetu *Casia Web Face* pak proběhl fine-tuning sítě *FaceNet*, přičemž jsme experimentovali s různými chybovými funkcemi.

2 Cíl projektu

Cílem projektu je dotrénovat síť *FaceNet* takovým způsobem, aby dosahovala lepších výsledků při verifikaci obličejů s nasazenými rouškami. Dále chceme ověřit, zda model dotrénovaný pomocí chybové funkce *ArcFace* dosáhne lepších výsledků, než model dotrénovaný pomocí *Cross-entropy*.

3 Relevantní materiály

Z množství odborných publikací, které se zabývají problematikou rozpoznávání obličejů je pro tuto práci nejrelevantnější článek pojednávající o síti *FaceNet*[4], jelikož na ní naše řešení zakládáme, a článek o chybové funkci *ArcFace*[1], jelikož jsme ji použili při fine-tuningu sítě *FaceNet*.

3.1 FaceNet

FaceNet představuje postup trénování hluboké konvoluční neuronové sítě, aby její výstup (*embedding*) pro zadaný obrázek obličeje představoval nízko dimenzionální informaci o dané identitě, která je následně možná použít v různých

aplikacích jako je verifikace, rozpoznávání a shlukování. Trénování *embeddingů* probíhá přímo pomocí *Triplet* chybové funkce. Euklidovská vzdálenosti dvou *embeddingů* na výstupu sítě přímo odpovídá podobnosti odpovídajících obličejů. Konkrétní implementace předtrénované sítě *FaceNet*, kterou jsme použili, je založena na architektuře *InceptionResNet-v1* a byla trénovaná na datasetu *Casia Web Face*[2].

3.2 ArcFace

ArcFace je state-of-the-art chybová funkce používaná pro trénování neuronových sítí pro rozpoznávání obličejů. Pokud si představíme *embedding* jednoho obrázku jako bod v n -dimenzionálním prostoru, pak *ArcFace* slouží k minimalizaci vzdálenosti mezi příklady jedné identity a zvyšováním vzdálenosti mezi příklady různých identit. Tohoto efektu je dosaženo přičtením penalizačního marginu v úhlové doméně, díky čemuž je síť nucena umísťovat *embeddingy* v tomto prostoru blíže resp. dál od sebe, aby i po přičtení penalizačního marginu byl vzorek přiřazen ke správné identitě.

4 Implementace modelu

K implementaci jsme zvolili jazyk *Python* s využitím nástroje *Pytorch*. Použili jsme síť *FaceNet* předtrénovanou na datasetu *Casia Web Face*. Délka *embeddingu* byla 512 příznaků. Pro standardní trénování s chybovou funkcí *Cross-entropy* stačilo předtrénovaný model použít přímo. V případě trénování s *Arcface* jsme museli doplnit L2 normalizaci vah poslední plně propojené vrstvy a odebrat předpětí taktéž poslední vrstvy.

4.1 Chybová funkce

Jak již bylo nastíněno, experimentovali jsme s dvěma chybovými funkcemi. Použili jsme jednak standardní funkci *Cross-entropy* a následně i funkci *Arcface*, kterou jsme implementovali dle popisu z článku, ve kterém byla představena[1].

5 Příprava datasetů a trénování

Pro trénování modelu jsme použili dataset *Casia Web Face*, vzhledem k tomu, že konkrétní implementace *FaceNet*, kterou jsme použili, byla předtrénovaná právě na tomto datasetu. Dalším důvodem je velikost – obsáhlejší dataset jsme si nemohli vzhledem k omezeným zdrojům pro trénování dovolit. Dataset obsahuje 10 575 identit a celkem 494 414 vzorků. V rámci přípravy dat jsme na obličej každé identity automatizovaně umístili roušky a rozšířili tak počet vzorků každé identity. Finální počet vzorků vzrostl na 836 032. Vzorků není dvojnásobek, jelikož doplnění roušky na některých obličejích, především na těch z polovičního nebo plného profilu, selhalo.

Rouška byla na obličej mapována na základě obličejových bodů získaných pomocí modelu *Dlib*[3].

Obličej byly následně oříznuty pomocí modelu *Mtcnn*[2] stejným způsobem, který byl použit při trénování námi zvolené implementace *FaceNet*.

Dataset jsme nakonec rozdělili v poměru 9:1 na trénovací a testovací část.



Obrázek 1: Ukázka upravených vzorků datasetu *Casia Web Face*. V prvním řádku jsou neupravené obličej, na druhém obličej s přidánými rouškami a na posledním finální vzorek po oříznutí.

Pro výsledné vyhodnocení modelu jsme použili dataset *LFW*, jelikož je všeobecně používán k vyhodnocení kvality natrénovaných modelů.

Dataset jsme ovšem opět upravili přidáním roušek na některé obličej tak, aby mezi celkovým počtem 6 000 párů, která *LFW* obsahuje, byly zastoupeny přibližně ve stejném počtu páry bez roušek, páry, ve kterých má roušku jeden obličej, a páry, kde mají roušku oba.

6 Metriky pro vyhodnocení úspěšnosti

Pro vyhodnocení jsme poslední plně propojenou vrstvu odstranili a použili jen část tvořící *embedding*, jelikož se jedná o *open set* problém. Kvalitu výsledných modelů jsme vyhodnotili, jak již bylo zmíněno, na datasetu *LFW* s přidánými rouškami, a to pomocí 2 různých metrik.

- a) úspěšnost určení, zda se jedná o pozitivní nebo negativní pár
- b) verifikační přesnost – hodnota *true accept rate* pro zafixovanou maximální hodnotu *false accept rate* 10^{-3} .

7 Experimenty

V rámci experimentů jsme nejprve vyhodnotili kvalitu původního předtrénovaného modelu na originálním datasetu *LFW*, poté na upraveném datasetu s

přidanými rouškami, kde hodnoty metrik dle očekávání poklesly, a nakonec kvalitu dotrénovaného modelu na datasetu *LFW* s rouškami. V ideálním případě by se měly hodnoty metrik po dotrénování zpět přiblížit k hodnotám vyhodnocení na *LFW* bez roušek pro původní předtrénovaný model.

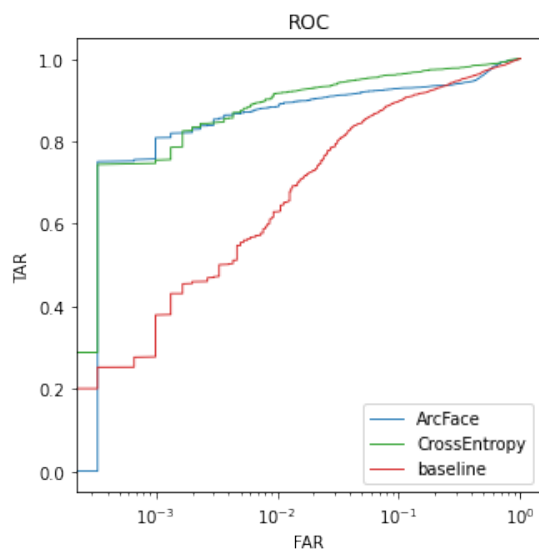
Pro měření výsledné úspěšnosti jsme vybrali takové modely, které v průběhu trénování dosáhly nejlepší úspěšnosti na testovacích datech.

7.1 Úspěšnost

Pro metriku a) se oběma dotrénovaným modelům podařilo velmi přiblížit původní úspěšnosti, která byla naměřena předtrénovanou sítí na datasetu bez roušek. Podle metriku b) modely vyhodnocovaly o něco hůře a k původní *true accept rate* se podařilo přiblížit pouze na rozdíl necelých čtyř procent. Přesto ale pokládáme zlepšení ze 33.85 % (kterých dosáhl nedotrénovaný model) na 75.79 % a na 78.66 % za úspěch. U obou modelů každopádně došlo ke zlepšení oproti úspěšnosti sítě *FaceNet* předtrénované na datasetu bez roušek.

porovnání výsledných hodnot metrik a) a b)			
model	dataset	a)	b)
předtrénovaný model	lfw	95.58 %	82.40 %
předtrénovaný model	lfw s rouškami	90.27 %	33.85 %
dotrénováno s <i>Cross-entropy</i>	lfw s rouškami	95.13 %	75.79 %
dotrénováno s <i>ArcFace</i>	lfw s rouškami	94.10 %	78.66 %

7.2 ROC křivky



Obrázek 2: ROC křivky výsledných modelů získané evaluací na datasetu *LFW* s rouškami (baseline značí předtrénovanou síť *FaceNet* bez úprav).

8 Závěr

Dosáhnout lepší úspěšnosti verifikace osob s rouškami, než jaké dosahuje předtrénovaná síť *FaceNet* bez fine-tuningu, se nám podařilo s oběma modely dotrénovanými na datasetu s obličejí v rouškách. Porovnání modelů dotrénovaných s různými chybovými funkcemi však nevedlo na jednoznačné výsledky. Model dotrénovaný s chybovou funkcí *ArcFace* dosáhl lepších výsledků podle metriky b), ale model dotrénovaný s chybovou funkcí *Cross-entropy* dosáhl lepších výsledků podle metriky a).

Reference

- [1] Jiankang Deng, Jia Guo, Niannan Xue, and Stefanos Zafeiriou. Arcface: Additive angular margin loss for deep face recognition, 2019.
- [2] Tim Esler. Facenet-pytorch. <https://github.com/timesler/facenet-pytorch>.
- [3] Davis E. King. Dlib. <https://github.com/davisking/dlib>.
- [4] Florian Schroff, Dmitry Kalenichenko, and James Philbin. Facenet: A unified embedding for face recognition and clustering. *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2015.