Unmasked - Image Inpainting

Martin Eršek [xersek00], Svätopluk Hanzel [xhanze10]

30. mája 2021

1 Úvod

Image inpainting je metóda, ktorá syntentizuje alternatívny realisticky vyzerajúci vizuálny obraz do vyznačených miest v obrázku tak, aby doplnený obsah vyzeral čo najrealistickejšie a zároveň bol aj sémanticky korektný. Medzi možné využitia tejto metódy patrí napríklad odstránenie nežiadúcich objektov (ľudí alebo predmetov) z obrázku, prípadne takzvaným „retouching“ nežiadúcich oblastí v obrázku.

1.1 Existujúce prístupy

K riešeniu problému image inpaintingu existuje mnoho navrhovaných prístupov. Tradičné „staré“ prístupy za pomoci „patch-based“ algoritmov ako [1, 2], ktoré progresívne rozširujú pixely blízko okrajov dopĺňanej lokality a na základy nízko úrovňových „features“, akými sú napríklad MSD[1] v RGB priestore vyhľadávajú a vkladajú najpodobnejšie „patches“. Tento starší prístup funguje veľmi dobre na „statických“ obrázkoch akými sú napriklad textúry, no má veľké problémy pri „nestatických“ obrázkoch. Novšie prístupy k image inpaintingu sú založené na hlbokom učení vďaka ktorému je možné priamo predikovať hodnoty dopĺňaných pixelov v inpainting maske. Veľkou výhodou týchto modelov je ich schopnosť naučiť sa adaptívne „features“ obrázkov pre rôzne sémantické situácie vyskytujúce sa v obrázkoch. Vďaka tejto vlastnosti sú schopné syntetizovať viac vizuálne uveriteľnejší obsah obzvlášť pre obrázky akými sú napríklad ľudské tváre [3, 4, 5, 6, 7], objekty [8, 7] alebo prírodné scenérie [5, 6, 7].

2 Definícia úlohy

Rozhodli sme sa implementovať neurónovú sieť, ktorej úlohou bude odstrániť „face-mask“[2] z tváre človeka. Úloha pozostáva z detekcie umiestnenia tvárovej masky a následnej syntézy pixelov tváre osoby, ktoré sú skryté pod danou maskou tak, aby čo najvierohodnejšie reprezentovali tvár človeka. Keďže je vďaka maske zakrytá tvár skúmanej osoby a o tejto osobe nepoznáme žiadne aditívne informácie, znamená to, že aj keď bude síce doplnená časť tváre vyzerať realisticky, nebude (nemôže) reprezentovať reálnu tvár skúmanej osoby. Túto skutočnosť by bolo možné riešiť doplnením informácií o tvári skúmanej osoby. Týmto spôsobom by bolo možné dopĺňať nie len realisticky vyzerajúce časti tváre pod maskou ale zároveň by toto doplnenie skutočne reprezentovalo tvár skúmanej osoby. Takúto rozšírenú verziu našej úlohy by bolo možné použiť napríklad na „re-touching“ rodinných fotiek zo spoločenských oslav na ktorých bolo nosenie tvárových másk povinné. Toto rožšírené zadanie úlohy však ponechávame na prípadné ďalšie práce, ktorých autori by sa rozhodli na našu prácu nadviazať.

[1] Mean Square Difference

[2] Rúško alebo respirátor

3 Riešenie a implementácia

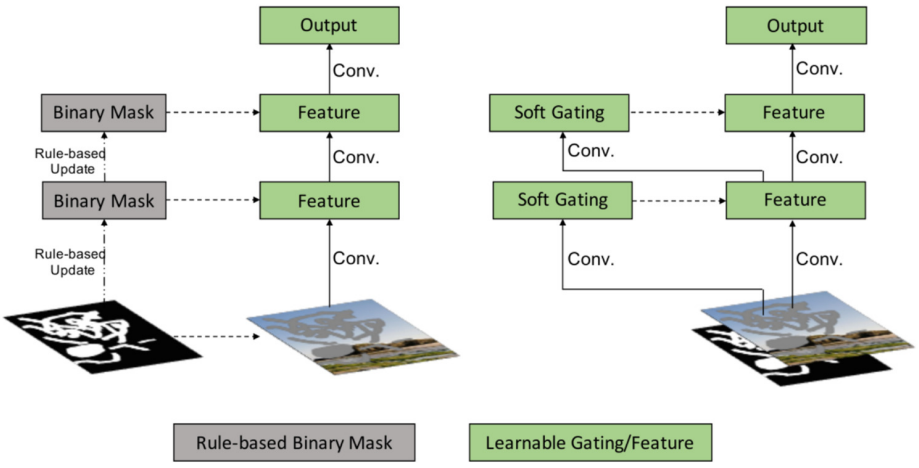
Naše riešenie tohto zadania spočíva primárne v implementácií ML inpainting modelu a jeho natrénovaní na datasete 3.3.

3.1 Model

Zvolený ML model je GAN sieť - konkrétne SN-PatchGAN [7]. Od implementácie v pôvodnom článku sme sa odklonili hlavne v tom, že náš model neberie ako vstup 5.-ty, tzv *sketch*, kanál a preto nepodporuje vedené (*guided*) dokreslovanie.

3.1.1 Gated konvolúcie

SN-PatchGAN sa vyznačuje hlavne použitím tzv. *gated* konvolúcií, ktoré nadväzujú na *parciálne konvolúcie*[9]. Gated konvolúcie na rozdiel od parciálnych konvolúcií umožňujú sieti naučiť sa *soft* masku konvolúcie.



Obr. 1: Parciálna vs gated konvolúcia

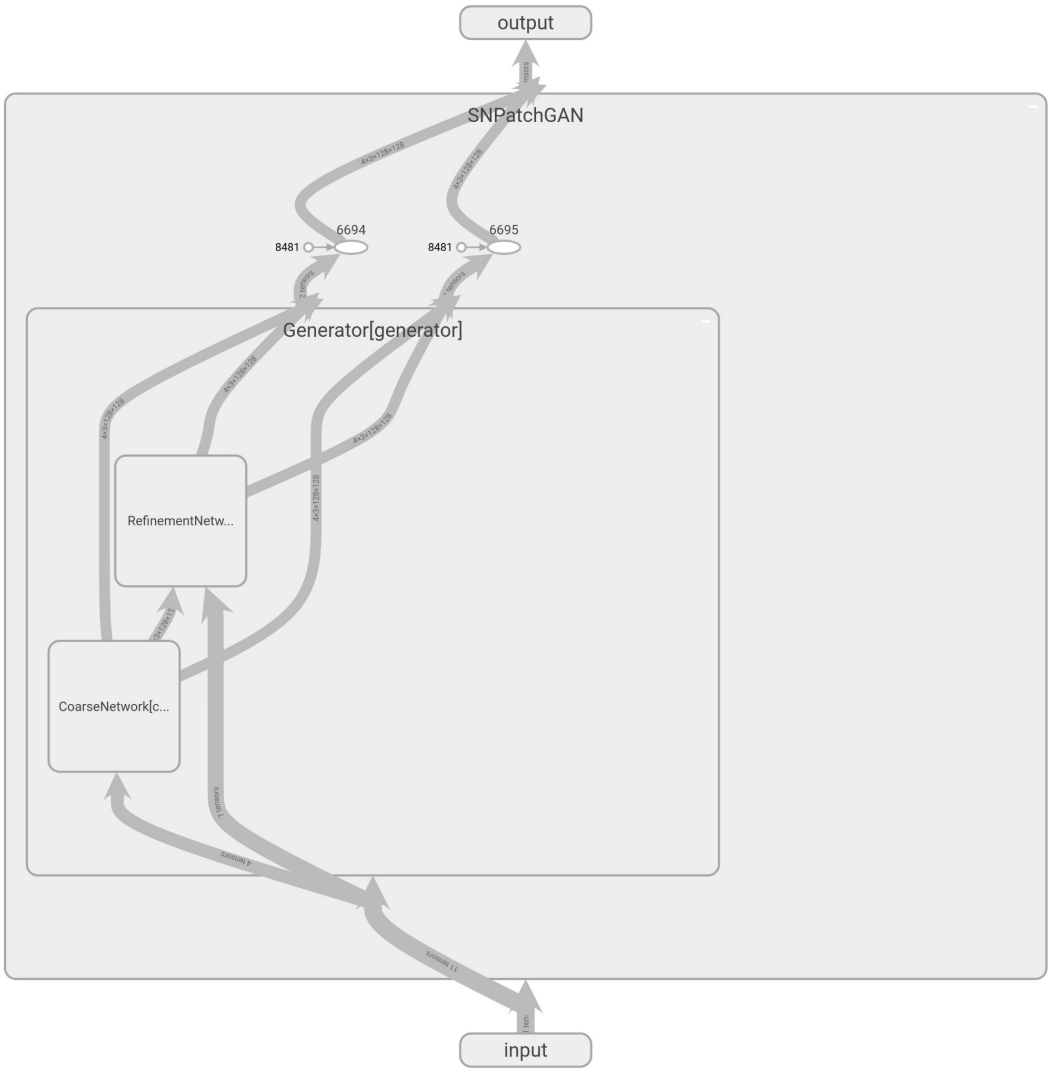
3.1.2 Generátor

Generátor tohto modelu sa skladá z 2 podsietí - *coarse* a *refinement*, pričom prvá z nich má za úlohu vytvoriť priebežný medzivýsledok, zatiaľ čo druhá ho zdokonalí a vykreslí detaily.

Vstupom je 3-kanálový vstupný obrázok a 1-kanálový obrázok - maska, ktorá určuje polohu tvárovej masky vo vstupnom obrázku. Výstupom sú 2 3-kanálové obrázky – coarse a refined obrázok.

Coarse sieť sa skladá z niekoľkých konvolučných a dekonvolučných gated3.1.1 vrstiev, pričom prostredné z nich sú dilatačné.

Refinement sieť je mierne komplikovanejšia s dvoma vetvami. Prvá z nich obsahuje *contextual attention* [6] vstvu, ktorá sieti umožňuje naučiť sa kedy si má „požičať“ informácie z pozadia obrázku. Druhá vetva obsahuje len gated konvolučné vrstvy, pričom niektoré z nich sú dilatačné.



Obr. 2: Graf siete nášho modelu

3.1.3 Diskriminátor

Diskriminátor je 6-vrstvová plne konvolučná sieť so spektrálnou normalizáciou na všetkých vrstvách.

3.2 Implementácia

Model je implementovaný v Pythone [3] verzie 3.9 s použitím knižnice PyTorch[4] a PyTorch Lightning[5].

PyTorch Lightning je pomocná knižnica PyTorch, ktorá zjednodušuje prácu s PyTorch a umožnuje spúšťanie kódu na rôznych platformách bez potreby zmeny kódu. Vďaka tomu môžeme náš kód spustiť na CPU, CUDA GPU aj TPU bez zmeny kódu. Zároveň podporuje aj distribuované učenie na viacerých grafických kartách, čo nám veľmi zrýchlilo trénovanie a pomohlo s experimentami.

[3] `https://www.python.org/`
[4] `https://pytorch.org/`
[5] `https://www.pytorchlightning.ai/`

Samotná implementácia siete je realizovaná v moduli `network`, pričom PyTorch modul obaľujúci celú sieť je implementovaný cez triedu `SNPatchGAN` v súbore `gan.py`. Všetky ostatné podsiete, rovnako ako vrstvy sú uložené v samostatných súboroch pre lepšiu orientáciu a udržateľnosť kódu.

3.3 Dataset

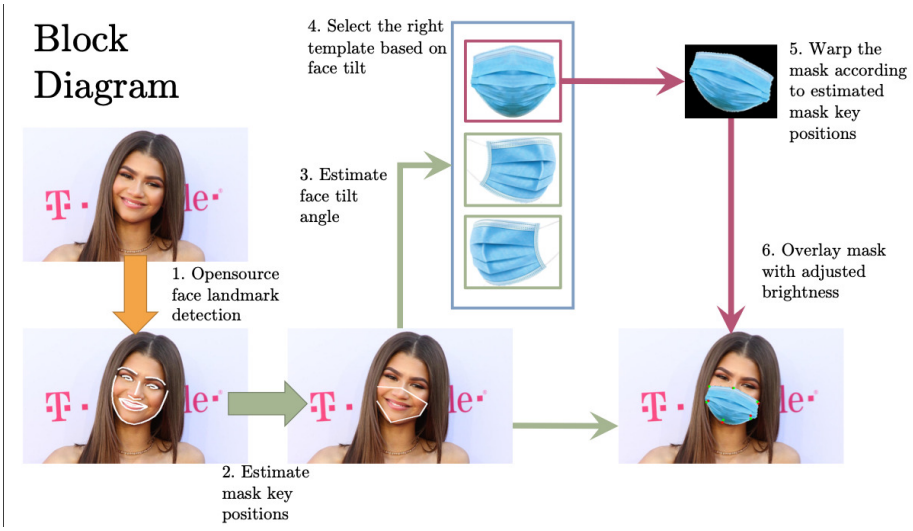
Pre trénovanie tohto modelu sme použili dataset CelebA-HQ [10], ktorý obsahuje celkovo 30 000 obrázkov tvárí celebrít v rozlíšení 1024x1024px. Tieto fotky sú však bez prekrytia tváre, ktoré potrebujeme pre tento projekt.

Pre potreby trénovania siete sme vytvorili triedu `MaskedCelebADataset`[6], ktorá dokáže načitavať obrázky z určeného miesta, aplikovať na ne on-the-fly masku a vrátiť tieto dáta vo formáte vhodnom pre ďalšie spracovanie sieťou. Táto trieda očakáva cestu k zložke s datasetom, pričom táto zložka ďalej musí obsahovať zložku `images`, v ktorej sú priamo umiestnené všetky obrázky.

Rozdelenie dát Dataset obsahuje 30 000 obrázkov, ktoré sme rozdelili v pomere 70:30 na trénovacie a validačné vzorky, takže validácia prebiehala vždy len na obrázkoch, ktoré táto sieť ešte nevidela.

3.3.1 Maskovanie tváre

Na prekrytie tváre sme použili projekt MaskTheFace[7] [11], ktorý dokáže pomocou predučeného modelu rozoznať rysy tváre a podľa nich vložiť jeden z preddefinovaných obrázkov masiek na správne miesto spolu s natočením a zmenou veľkosti masky, čím sa zvyšuje dôveryhodnosť výsledku.



Obr. 3: Diagram fungovania maskovania tváre [11]

V rámci riešenia projektu UnMasked sme zároveň refaktorovali MaskTheFace, prerobili sme ho na importovateľnú python knižnicu a zjednodušili niektoré volania v ňom. Tieto zmeny nám umožnili vytvorenie on-the-fly maskovania počas trénovania. Zároveň sa chystáme doplniť funkcionalitu, ktorá pre nás nebola potrebná a vytvoriť merge request z nášho[8] do pôvodného repozitára.

[6] súbor `dataset.py`
[7] `https://github.com/aqeelanwar/MaskTheFace`
[8] `https://github.com/sveatlo/MaskTheFace/`

3.4 Trénovanie

Trénovanie siete prebiehalo na 1 počítači s dvomi grafikami - NVIDIA GTX1080 8GB a NVIDIA GTX1070 8GB.

Trénovací skript je implementovaný v `train.py` a umožňuje predávanie hyperparametrov siete cez jednoduché používateľské rozhranie. Najdôležitejšími parametrami sú:

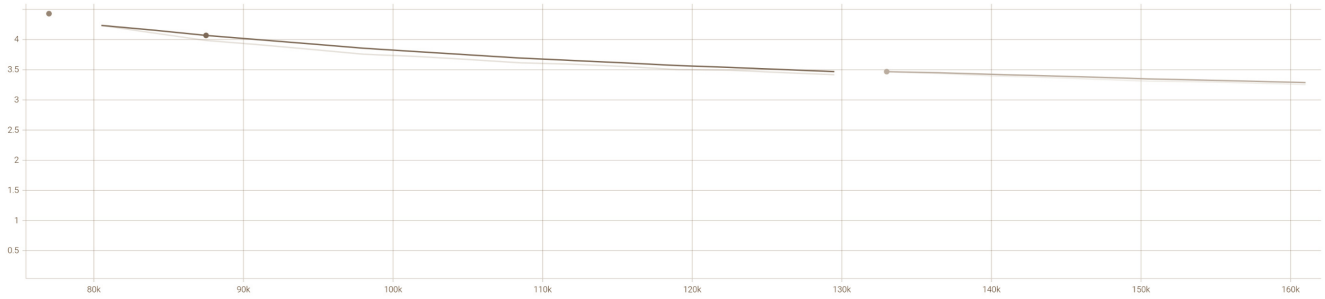
- `--gpus`, ktorý umožňuje zvoliť počet GPU, ktoré sa majú použiť
- `--resume_checkpoint`, ktorý umožňuje obnoviť prerušené trénovanie

Batch size je voliteľná pomocou prepínača `--batch_size`, pričom nám sa počas testovania s obrázkami veľkosti 256x256px najviac osvedčila veľkosť 16.

Optimalizátor Pre učenie generátora aj diskriminátora je použitý stochastický optimalizátor Adam [12]. Rozdiel medzi optimalizáciou generátora a diskriminátora je, že optimilizátor pre diskriminátor sa spúšťa len raz za 5 batch-ov, čo zabraňuje jeho rýchlemu preučeniu.

Loss funkcia Celková loss funkcia pre generátor je výsledkom spojenia L1 loss funkcie na coarse a refined obrázkoch, lokálnej loss funkcie (výstup diskriminátora 3.1.3) a globálnej loss funkcie (výstup perceptual network - predučená sieť VGG16).

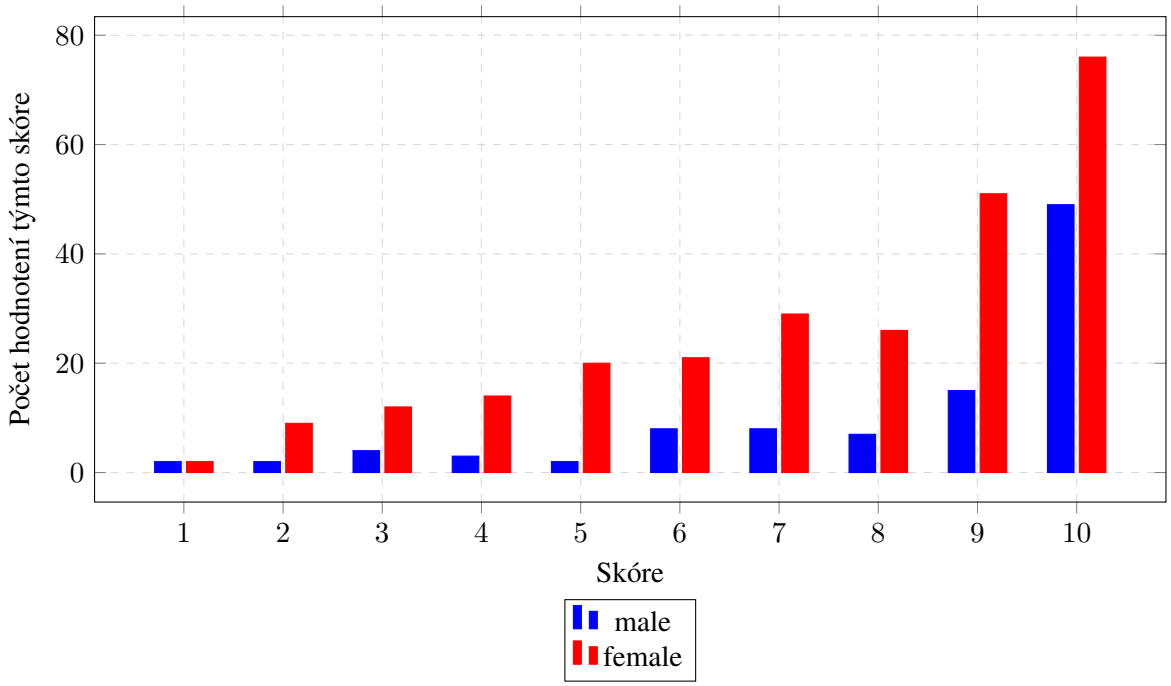
Learning rate je nastaviteľná samostatne pre generátor a diskriminátor. Nám sa overila hodnota 2e-4, vďaka ktorej sme dosiahli stabilné učenie.



Obr. 4: Priebeh perceptual loss

4 Vyhodnotenie

Vyhodnotenie nášho modelu sme vykonali za pomoci užívateľského výskumu, ktoré sa zúčastnilo 10 užívateľov. Ich úlohou bolo ohodnotiť rôzne výstupné obrázky modelu a určiť na škále od 1 do 10 ako veľmi realistický je výstup nášho modelu. Skóre 10 značí, že si užívateľ myslí, že obrázok vyzerá ako reálny obrázok (fotografia) a nebol nijako upravovaný počítačom. Skóre 1 značí, že je zjavné, že obrázok nie je reálny a do obrázku niečo dokresloval počítač. Užívatelia hodnotili subjektívne a to tak, že pokiaľ tvár vyzerala uveriteľne no obsahovala nejaký miniatúrny artefakt, ktorý spozorovali až po dlhšom skúmaní, tak uviedli skóre blížiace sa viac k 10. V opačnom prípade sa blížili k 1. Skóre jedna udeľovali naozaj len v prípadoch pokiaľ doplnená časť tváre do obrázku absolutne nesedela. Ako možno vidieť z obrázku 5, veľké množstvo výstupných fotografií dostalo skóre 10 značiace, že užívatelia by nevedeli rozpoznať či sa jedná o skutočnú fotografiu alebo počítač tak dokonale dokreslil úsek tváre, že to nie je možné rozoznať. Tento poznatok značí, že náš model funguje úspešne tak ako má.



Obr. 5: Ohodnotenie vierohodnosti obrázka za pomoci užívateľského výskumu. X-ová os značí to ako veľmi realisticky vyzeral obrázok pre užívateľov. 1 - nerealistický, generovaný počítačom; 10 - originálna neupravovaná fotka.

Nižšie skóre dostali prevažne obrázky obsahujúce tvár celebrity v netypickej polohe, prípadne pokiaľ bola tvár celebrity niečím čiastočne prekrytá - napríklad rukou alebo palicou[9].

[9] V datasete existovali aj obrázky kde celebrita držala v ruke palicu, ktorá prekrývala časť jej tváre.

Zdroje

[1] Efros, A. A.; Freeman, W. T. *Image Quilting for Texture Synthesis and Transfer* in *Proceedings of the 28th Annual Conference on Computer Graphics and Interactive Techniques* SIGGRAPH '01 New York, NY, USA: Association for Computing Machinery 2001 ISBN 158113374X str. 341–346 doi:10.1145/383259.383296.

[2] Efros, A. A.; Leung, T. K. *Texture Synthesis by Non-parametric Sampling* in *IEEE International Conference on Computer Vision* Corfu, Greece September 1999 s. 1033–1038.

[3] Li, Y.; Liu, S.; Yang, J.; aj. *Generative Face Completion* 2017 1704.05838.

[4] Yeh, R. A.; Chen, C.; Lim, T. Y.; aj. *Semantic Image Inpainting with Deep Generative Models* 2017 1607.07539.

[5] Iizuka, S.; Simo-Serra, E.; Ishikawa, H. *Globally and Locally Consistent Image Completion ACM Trans. Graph.* ročník 36, č. 4 Červenec 2017 ISSN 0730-0301 doi:10.1145/3072959.3073659.

[6] Yu, J.; Lin, Z.; Yang, J.; aj. *Generative Image Inpainting with Contextual Attention* 2018 1801.07892.

[7] Yu, J.; Lin, Z.; Yang, J.; aj. *Free-Form Image Inpainting with Gated Convolution arXiv preprint arXiv:1806.03589* 2018.

[8] Pathak, D.; Krahenbuhl, P.; Donahue, J.; aj. *Context Encoders: Feature Learning by Inpainting* 2016 1604.07379.

[9] Liu, G.; Reda, F. A.; Shih, K. J.; aj. *Image inpainting for irregular holes using partial convolutions* in *Proceedings of the European Conference on Computer Vision (ECCV)* 2018 s. 85–100.

[10] Liu, Z.; Luo, P.; Wang, X.; aj. *Deep Learning Face Attributes in the Wild* in *Proceedings of International Conference on Computer Vision (ICCV)* December 2015.

[11] Anwar, A.; Raychowdhury, A. *Masked Face Recognition for Secure Authentication* 2020 2008.11104.

[12] Kingma, D. P.; Ba, J. *Adam: A Method for Stochastic Optimization* 2017 1412.6980.

Appendices

A Ukážka výstupov z modelu

todo