

# PAIN: Painting Agressive Images Network

Matej Kunda, Marek Mudroň, Samuel Repka

Máj 2023

## 1 Image inpainting

Úlohou je doplnenie zamaskovaných oblastí obrázku takým spôsobom, aby obrázok doplnený o zrekonštruovanú časť nejakým spôsobom dával konceptuálny význam. Táto rekonštrukcia sa okrem iného môže vykonávať za účelom opravy poškodených oblastí obrázku či vnesením kreatívnych prvkov do fotografií.

### 1.1 Súčasné riešenia

Súčasné riešenia implementujú viaceré prístupy využívané v iných oblastiach spracovania obrazu a riešia pomocou nich problém image inpaintingu.

#### 1.1.1 GAN

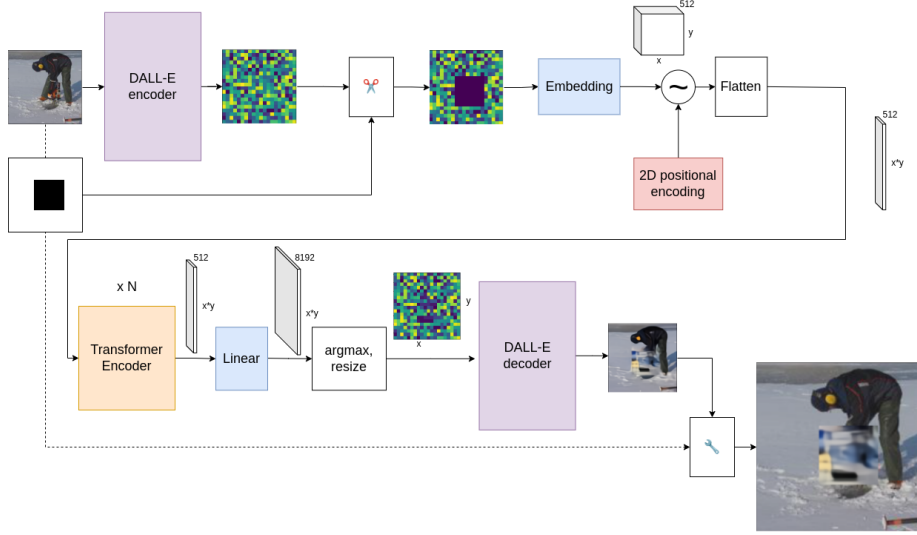
Modely tohto typu sa trénujú takým spôsobom, že generátor vytvára potenciálne riešenia, ktorých vhodnosť je ohodnotená diskriminátorom. Obe siete sa trénujú súčasne.

#### 1.1.2 Difúzne modely

V poslednej dobe veľmi populárne difúzne modely generujú výstup iteratívnym procesom, kde sa z počiatočnej konfigurácie náhodného vstupu (často riadiacej sa normálnym rozdelením), generuje výstup. V každej iterácii sa generovaný výstup má približovať zmysluplnej reprezentácii obrázku. Nevýhodou difúzných modelov je, že spomenutý proces generovania je často časovo náročný. A to nie len pri tréňovaní ale aj pri inferencii.

## 2 PAIN

### 2.1 Architektúra modelu - variant 1



Nakoľko sa transformery využívajú najmä pre modelovanie jazyka, tak obrázok predtým, ako prejde modelom, musí zodpovedať vhodnej reprezentácii. V jazykových modeloch sa slová *tokenizujú*, teda každému z nich sa priradí jedinečný identifikátor, čím je zaistený bijektívny vzťah medzi slovom a tokenom.

Pre tokenizáciu využívame DALL-E encoder, ktorý z obrázku na vstupe vytvorí 2D maticu, kde sa v každom poli nachádza jedna z 8192 možných hodnôt. Každá z týchto hodnôt zmysluplným spôsobom reprezentuje časť vstupného obrázku o veľkosti  $8 \times 8$ px.

Z tejto matice vynulujeme niektoré z tokenov podľa masky na vstupe. Dôvod, prečo sa maska aplikuje na maticu tokenov namiesto surového obrázku je ten, že z obrázka s aplikovanou maskou je neskôr nesprávne vytvorená matica tokenov v dôsledku toho, že DALL-E encoder považuje vynulované hodnoty v obrázku za významné v rámci kontextu.

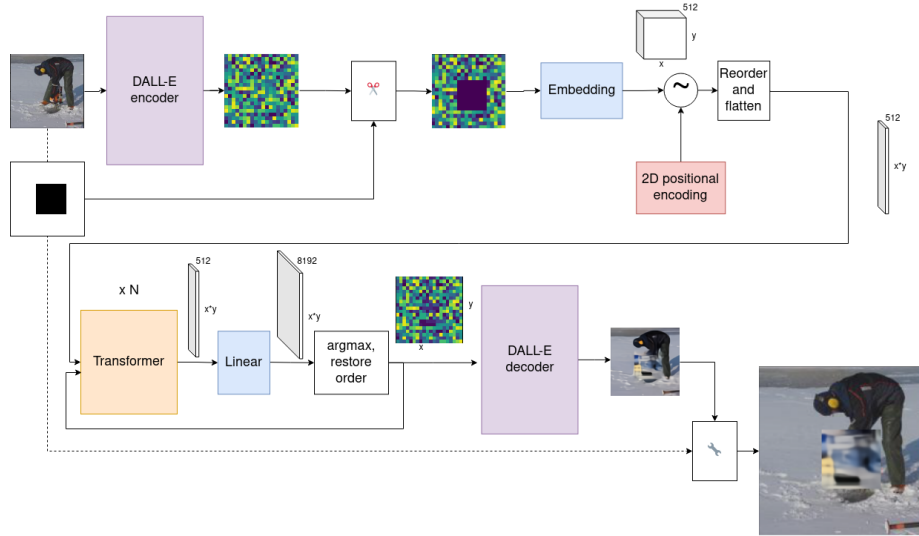
Pomocou embedding vrstvy je hodnota každého z tokenov rozšírená na vektor o veľkosti 512. Na tieto vektory aplikujeme 2D pozičný encoding<sup>1</sup> a výstup sploštíme. Takto sploštený vstup tvorený vektormi o veľkosti 512 následne vstupuje do Transformer Encoderu.

Transformer Encoder je tvorený  $N = 6$  za sebou idúcimi vrstvami. Na výstupe poslednej z nich je vektor o rovnakej veľkosti ako ten, ktorý vstupoval do prvej z nich. Takýto výstup prejde lineárnou vrstvou, ktorá rozšíri každý z prvkov o veľkosti 512 na vektor o veľkosti 8192. Aplikovaním operácie *argmax* nad každým z týchto vektorov získame hodnoty tokenov, o ktorých si model myslí, že sú na danej pozícii najvhodnejšie.

<sup>1</sup><https://github.com/tatp22/multidim-positional-encoding>

Takáto matica tokenov vstupuje do DALL-E decoderu, ktorý z nich zrekonštruuje obrázok. Z neho extrahujeme maskovanú časť a nahradíme ňou vymaskovanú časť v obrázku na vstupe.

## 2.2 Architektúra modelu - variant 2



Druhý variant modelu je veľmi podobný prvému, avšak architektúra je prispôbena autoregresívnemu tréновaniu a inferencii. Po 2D pozičnom embeddingu sa celý vstupný tenzor reštrukturalizuje spôsobom, aby všetky vymaskované tokeny boli na konci. V transformerovej časti sa pridali k enkóderom aj dekodéry. Ako vstup do transformer enkóderu idú takto poprehadzované nevymaskované tokeny a ako vstup do dekodéru idú už vygenerované vymaskované tokeny. Po doplnení všetkých vymaskovaných tokenov musíme výstupný tenzor znovu poprehadzovať, aby boli doplnené tokeny na pôvodných miestach. Na začiatku inferencie je na vstupe do dekodéru špeciálny token [SoS] (start of sequence).

## 2.3 Dataset

Bidirectional model sme trénovali na jednej triede datasetu *Imagenette2* s identifikátorom *n03000684*, kde sa zrejme nachádzajú obrázky súvisiace s motorovými pílami. Tieto obrázky majú veľkosť 160px aspoň v jednej z osí. Tie orezávame v strede na veľkosť 160x160. Ide o 858 vzoriek, čo nie je veľa, no z dôvodu nedostatku výpočtových zdrojov sme boli nútení sa uchýliť k tejto alternatíve. Autoregresívny model bol trénovaný na triede rovnakého datasetu *Imagenette2* s identifikátorom *n01440764*, kde sa nachádzajú ryby. Modely neboli trénované na rovnakých triedach kvôli tomu, že autoregresívny model nedokázal poskytnúť použiteľné výsledky pri triede motorových píl.

## 2.4 Trénovanie

Vytrénovali sme dva modely, z ktorých každý bol trénovaný odlišným spôsobom. Pre každý obrázok na vstupe bola náhodne vygenerovaná maska o veľkosti  $64 \times 64$ px, s ktorou sme ďalej pracovali. Ako loss funkciu sme použili *cross-entropy*.

Pre optimalizáciu tréningu sme použili *Adam* a learning rate 0.0003.

### 2.4.1 Trénovanie variantu 1

Pri tomto procese sme sa inšpirovali jazykovým modelom BERT, ktorý sa učí modelovať jazyk takým spôsobom, že dopĺňa zamaskované slová nahradené [MASK] tokenom vo vete v závislosti od kontextu daného ostatnými slovami. V našom prípade používame token o hodnote 0. Je zrejmé, že DALL-E Encoder považuje 0 za hodnotu, ktorá zmysluplne reprezentuje kúsok obrázka. Pri tréningu však tento token používame ako masku.

### 2.4.2 Trénovanie variantu 2

Rozdiel oproti bidirectional a autoregressive je taký, že pri autoregresívnom tréningu sa postupne odhaľujú tokeny, ktoré model vidí pri generovaní nasledujúceho slova. Je tým zaistená kauzalita (nasledujúce slovo je generované v závislosti od kontextu vytvoreného slovami nachádzajúcimi sa pred ním v rámci pozície vo vete). Takéto tréningovanie má význam najmä pri jazykových modeloch, kde model pri tvorbe komplikovaných vetných konštrukcií nevidí nasledujúce slová. Príkladom takéhoto modelu je GPT-2.

V našom prípade používame masku, reprezentovanú diagonálnou maticou, ktorá má v pravej hornej časti hodnoty  $-\infty$  a ľavej dolnej a na diagonále hodnoty 0.

$$M = \begin{bmatrix} 0 & \cdots & -\infty \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 0 \end{bmatrix} \quad (1)$$

Táto maska reprezentuje autoregresívny vzťah tak, že pre generovanie tokenu sa použijú iba vstupné a už vygenerované tokeny.

## 3 Experimenty

Experimenty sme vykonávali pomocou validačnej množiny obrázkov datasetu doplnených o masku. Vyhodnotenie výstupov modelu prebiehalo len vizuálne, vzhľadom na kvalitu výstupov sme považovali inú evaluáciu za zbytočnú.








### 3.1 Výstupy prvého variantu

Na obrázkoch je zjavné, že model sa snaží porozumieť kontextu obrázku, ale pomerne neúspešne. Farby a štruktúry pripomínajú pôvodný obrázok, ale len ťažko sa to dá považovať za úspešne doplnenú masku. Tiež má tendenciu vytvárať horizontálne/vertikálne čiary v doplnenej maske.

Vstup	Maskovaný vstup	Výstup variantu 1
		
		
		
		

### 3.2 Výstupy druhého variantu

Autoregresívny model tiež vykazuje istú mieru pochopenia scény, ale má problém doplniť miesta, ktoré by mali mať približne rovnakú farbu (napr. 2 príklad). Snaží sa vždy vložiť nejaké ozvláštnenia do zamaskovanej časti. Na druhej strane vykazuje ale menej artefaktov typu vertikálne/horizontálne čiary, ako predchádzajúci model. Existujú prípady, kedy model jednoducho nedoplní nič. Potenciálne by to mohlo byť vyriešené dlhším tréňovaním alebo väčším datasetom.

Vstup	Maskovaný vstup	Výstup variantu 2
		
		
		
		

## 4 Záver

Model 1 dosahuje slabé výsledky, aj keď je zjavné, že má potenciál rozumieť kontextu obrázku. Teoreticky to môže byť spôsobené malým počtom dát použitých pri trénovaní. Ďalší dôvod, prečo model nedosahuje uspokojivé výsledky môže

byť tým, že jedna hodnota tokenu sa môže vyskytovať v rôznych kontextoch. Rozdiel medzi prístupom, ktorý používa BERT a prístupom používaným našim modelom je ten, že BERT využíva bijektívne mapovanie tokenov na slová zo slovníka, ktorý má pevnú dĺžku. V dôsledku tohto má každý z tokenov jednodznačnú reprezentáciu v podobe slova. V našom prípade môže jedna hodnota tokenu vytvoriť rôzne reprezentácie v podobe kombinácie pixelov. Tento fenomén sa vyskytuje práve v dôsledku použitia DALL-E decoderu, ktorý pri konverzii tokenov do pixelového priestoru zvažuje hodnoty tokenov v okolí daného miesta.

Autoregresívny prístup takisto vykazuje potenciál, aj keď nami vytrénovaný model nie je spoľahlivý. Priebežne sa stane, že model odmietne čokoľvek doplniť, teda vráti len jeden token pre celú masku. Tiež treba poznamenať, že ho bolo treba trénovať veľmi dlho, kým začal generovať niečo rozumné (finálna verzia bola trénovaná 500 epoch).

V konečnom dôsledku je použitie transformerov obtiažne, ale nie nereálne. V kombinácii s jednoduchou konvolučnou neurónovou sieťou na doplnenie detailov by vygenerované časti mohli získať textúru a vyzeráť reálnejšie. Ďalšie potenciálne zlepšenia, ktoré sme neskúmali môžu byť napríklad použitie kompresie vstupných dát, ktoré nie je kontextové, prípadne dosahuje vyššieho kompresného pomeru.