

Politechnika Warszawska

W Y D Z I A Ł E L E K T R O N I K I
I T E C H N I K I N F O R M A C Y J N Y C H



Instytut Radioelektroniki i Technik Multimedialnych

Praca dyplomowa

na kierunku Studia Podyplomowe
w specjalności Głębokie Sieci Neuronowe - Zastosowania w Mediach Cyfrowych

Inżynieria wsteczna modelu Stable Diffusion 2.0 - generowanie prompta
na podstawie obrazu

Paweł Kalisz

Numer albumu 1184926

promotor
dr Jacek Komorowski

WARSZAWA 2024

Inżynieria wsteczna modelu Stable Diffusion 2.0 - generowanie prompta na podstawie obrazu

Streszczenie.

Główym celem pracy dyplomowej jest przygotowanie modelu głębokiej sieci neuronowej przewidującej opis tekstowy jaki został użyty do wygenerowania podanego na wejściu obrazu wygenerowanego za pomocą modelu dyfuzyjnego. W ramach pracy zaimplementowano, wytrenowano i przeprowadzono ewaluację dwóch modeli sieci neuronowych generujących opisy tekstowe na podstawie zadanego obrazu. Modele wytrenowano na zbiorze danych zawierającym 1,084,245 par złożonych z obrazu wygenerowanego modelem Stable Diffusion 2.0 i tekstuowego opisu (promptu) wykorzystanego do wygenerowania obrazu. W pierwszej części pracy przedstawiono proces przygotowania danych do postaci gotowych wsadów oraz opisano architektury obu modeli i proces trenowania. Następnie opisano metryki użyte do ewaluacji obu modeli oraz wyniki. Ewaluacja eksperymentalna pokazuje znaczącą przewagę modelu opartego na architekturze Transformer nad modelem opartym o koder CNN-dekoder LSTM. W zakończeniu pracy przedstawiono podsumowanie wykonanych działań.

Słowa kluczowe: StableDiffusion2.0, Transformer, LSTM, CNN, RNN



.....
miejscowość i data

.....
imię i nazwisko studenta

.....
numer albumu

.....
kierunek studiów

OŚWIADCZENIE

Świadomy/-a odpowiedzialności karnej za składanie fałszywych zeznań oświadczam, że niniejsza praca dyplomowa została napisana przeze mnie samodzielnie, pod opieką kierującego pracą dyplomową.

Jednocześnie oświadczam, że:

- niniejsza praca dyplomowa nie narusza praw autorskich w rozumieniu ustawy z dnia 4 lutego 1994 roku o prawie autorskim i prawach pokrewnych (Dz.U. z 2006 r. Nr 90, poz. 631 z późn. zm.) oraz dóbr osobistych chronionych prawem cywilnym,
- niniejsza praca dyplomowa nie zawiera danych i informacji, które uzyskałem/-am w sposób niedozwolony,
- niniejsza praca dyplomowa nie była wcześniej podstawą żadnej innej urzędowej procedury związanego z nadawaniem dyplomów lub tytułów zawodowych,
- wszystkie informacje umieszczone w niniejszej pracy, uzyskane ze źródeł pisanych i elektronicznych, zostały udokumentowane w wykazie literatury odpowiednimi odnośnikami,
- znam regulacje prawne Politechniki Warszawskiej w sprawie zarządzania prawami autorskimi i prawami pokrewnymi, prawami własności przemysłowej oraz zasadami komercjalizacji.

Oświadczam, że treść pracy dyplomowej w wersji drukowanej, treść pracy dyplomowej zawartej na nośniku elektronicznym (płycie kompaktowej) oraz treść pracy dyplomowej w module APD systemu USOS są identyczne.

.....
czytelny podpis studenta

Spis treści

1. Wstęp	9
2. Cel pracy	11
3. Przegląd literatury	13
4. Opis rozwiązania	17
4.1. Przygotowanie danych	17
4.2. Model oparty o koder CNN-dekoder LSTM	18
4.3. Model oparty o architekturę Transformer	19
5. Wyniki ewaluacji eksperymentalnej	21
5.1. Metryki użyte do ewaluacji modeli	21
5.2. Ewaluacja modeli	22
5.3. Demonstracja działania modeli na obrazach ze zbioru testowego	23
5.4. Demonstracja działania modeli na obrazach wygenerowanych przez autora	25
6. Podsumowanie	29
7. Bibliografia	30
Spis rysunków	31
Spis tabel	31

1. Wstęp

Modele dyfuzyjne zyskały ogromną popularność, umożliwiając tworzenie obrazów kontrolowalnych na podstawie poleceń tekstowych zapisanych w języku naturalnym, tzw. promptów. Od momentu publikacji takich modeli jak DALL-E, Midjourney czy Stable Diffusion, różni specjaliści zaczęli je stosować w swoich dziedzinach. Na uwagę zasługuje tu między innymi użycie modelu Midjourney do wygenerowania zwycięskiego obrazu "Théâtre D'opéra Spatial" (Rys. [1.1]) w konkursie Colorado State Fair [1], generowanie syntetycznych obrazów radiologicznych za pomocą dostrojonego modelu Stable Diffusion [2] czy generowanie realistycznych filmów za pomocą modelu dyfuzyjnego Imagen Video [3].



Rysunek 1.1. Théâtre D'opéra Spatial

Generowanie obrazów z pożdanymi cechami nie jest jednak łatwe, ponieważ wymaga od użytkowników napisania odpowiednich promptów, określających dokładnie oczekiwane rezultaty. Opracowywanie takich promptów zazwyczaj odbywa się metodą prób i błędów, a efekty mogą wydawać się czysto przypadkowe. Na przykład, aby wygenerować szczegółowe obrazy, powszechnie stało się dodawanie do polecenia specjalnych słów kluczowych, takich jak "highly detailed", czy "unreal engine".

Prompt engineering stał się dziedziną badań w kontekście generowania text-to-text, w której bada się jak konstruować polecenia, aby skutecznie rozwiązywać różne zadania. Ponieważ duże modele text-to-image są stosunkowo nowe, istnieje potrzeba zrozumienia, jak reagują one na polecenia jak pisać skuteczne prompts oraz jak projektować narzędzia, które pomogą użytkownikom generować obrazy [4].

1. Wstęp

W zrozumieniu powyższego zagadnienia pomocna może się okazać inżynieria wsteczna modeli text-to-image, czyli odtwarzanie prompta z wygenerowanego obrazu. W niniejszej pracy zaprezentowałem dwa modele, które mierzą się z tym zadaniem:

1. Koder-dekoder z pretrenowanym modelem ResNet50 do ekstrakcji cech z obrazów oraz dekoderem opartym na komórkach LSTM z mechanizmem uwagi Bahdanau
2. Koder-dekoder oparty na architekturze Transformer przyjmujący na wejściu kodera mapę cech głębokich wyekstrahowanych z obrazu za pomocą pretrenowanego modelu ResNet50

Modele zostały wytrenowane na parach złożonych z obrazu wygenerowanego modelem Stable Diffusion 2.0 i promptu wykorzystanego do wygenerowania obrazu. Tak wytrenowane modele powinny generować opisy tekstowe zbliżone do promptów użytych do wygenerowania obrazów podanych na wejściu.

2. Cel pracy

Głównym celem pracy jest opracowanie modeli głębokich sieci neuronowych przewidujących podpis tekstowy jaki został użyty do wygenerowania podanego na wejściu obrazu wygenerowanego za pomocą modelu Stable Diffusion 2.0. W ramach pracy dyplomowej zaimplementowano, wytrenowano i przeprowadzono eksperymentalną ewaluację dwóch modeli sieci neuronowych generujących opisy tekstowe na podstawie zadanego obrazu. Z jednej strony jest to klasyczne zadanie generowania podpisu dla zadanego obrazu, z drugiej natomiast dochodzi tu dodatkowa trudność polegająca na tym, że nie każde słowo wpisane w promptcie musi być odzwierciedlone w wygenerowanym obrazie. Może się również zdarzyć sytuacja, w której model generuje obiekty nie będące składową prompta.

Do ewaluacji modeli użyłem metryk stosowanych w tłumaczeniu maszynowym takich jak BLEU, METEOR oraz BERT Score.

Poniżej przedstawiono kilka przykładów promptów oraz obrazów wygenerowanych na ich podstawie ze zbioru testowego, pokazujących jednocześnie przykłady wsadów modelu (obraz podany na wejściu) oraz oczekiwanych rezultatów (prompt użyty do wygenerowania obrazu).

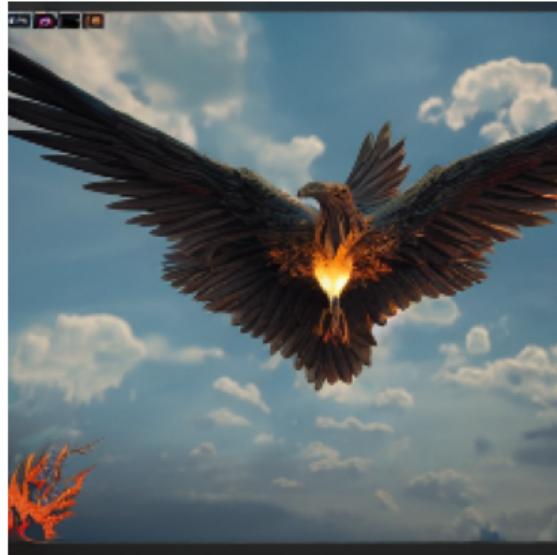
Prompt: a baroque close up portrait of a fantasy alien cyborg shaman god wearing facepaint and a colorful futuristic aztec visor with metallic technology holding a bird black background studio lighting highly detailed science fiction fantasy painting by norman rockwell moebius frank frazetta syd mead and sandro botticelli high contrast renaissance masterpiece artstation



Rysunek 2.1. Obraz ze zbioru testowego nr 1

2. Cel pracy

Prompt: phoenix flying in the sky super detailed detail unreal engine



Rysunek 2.2. Obraz ze zbioru testowego nr 2

Prompt: facing the dark star with a sword in hand galactic nebular astral realm sacred journey in oil painting trending on artstation award winning emotional highly detailed surrealist art



Rysunek 2.3. Obraz ze zbioru testowego nr 3

3. Przegląd literatury

Przy tworzeniu modeli oraz ewaluacji wyników wzorowałem się na rozwiązańach przedstawionych w literaturze naukowej opisanej poniżej.

Model przedstawiony w artykule **Show, Attend and Tell: Neural Image Caption Generation with Visual Attention** [5] łączy w sobie konwolucyjną sieć neuronową z dekoderem opartym na komórkach rekurencyjnych. Model składa się z dwóch głównych komponentów:

- Sieć konwolucyjna (CNN): Ta część modelu jest odpowiedzialna za przetwarzanie obrazów i ekstrakcję cech wizualnych. Sieć konwolucyjna analizuje obraz i tworzy mapę cech, która reprezentuje różne aspekty obrazu, takie jak kształty, kolory i tekstury.
- Sieć rekurencyjna (RNN): Generuje opis na podstawie mapy cech utworzonej przez sieć konwolucyjną. RNN działa sekwencyjnie, generując słowa jedno po drugim. W tym procesie model wykorzystuje mechanizm uwagi Bahdanau, aby zdecydować, na których częściach obrazu powinien się skoncentrować przy generowaniu każdego słowa. Dzięki temu model może tworzyć bardziej trafne i kontekstowo związane z obrazem opisy.

Pierwszy z modeli zaprezentowany w niniejszej pracy został zainspirowany powyżej opisanym artykułem.

Artykuł **Attention Is All You Need** [6] przedstawia przełomową architekturę w dziedzinie przetwarzania języka naturalnego (NLP) tzw. Transformer. Model ten, zaprezentowany w 2017 roku przez badaczy z Google Brain, wprowadził zupełnie nowe podejście do modelowania sekwencji, eliminując potrzebę stosowania tradycyjnych sieci rekurencyjnych (RNN) w pełni opierając się na mechanizmie atencji przy przetwarzaniu sekwencji.

Główne komponenty modelu Transformer:

- Scaled Dot-Product Attention: Jest kluczowym komponentem mechanizmu uwagi stosowanym w architekturze Transformer, działa na zasadzie obliczania wag dla różnych części danych wejściowych, aby określić, na które elementy model powinien zwracać większą uwagę podczas przetwarzania. Sekwencje wejściowe są przeliczane, przy użyciu macierzy wag, na trzy główne składniki: Queries (Q), Keys (K) i Values (V). Każde z tych składników jest reprezentacją wektorową. Następnie na podstawie poniższego wzoru obliczane są wektory uwagi.

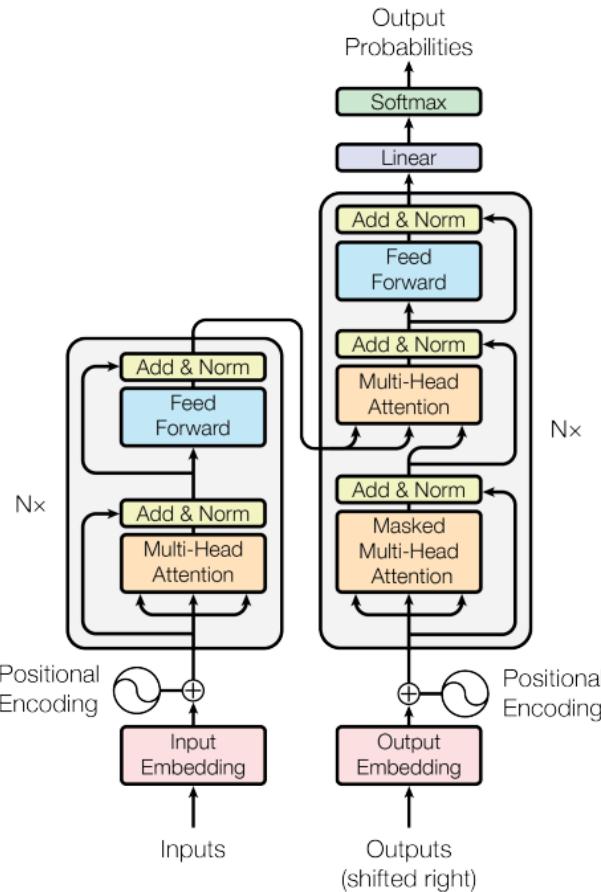
$$\text{Attention}(Q, K, V) = \text{softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Ten mechanizm pozwala modelowi na dynamiczne skupianie się na najważniejszych częściach danych wejściowych, co jest szczególnie przydatne w zadaniach

3. Przegląd literatury

związań z przetwarzaniem sekwencji, takich jak tłumaczenie maszynowe czy generowanie tekstu.

- Multi-Head Attention: Rozwija ideę Scaled Dot-Product Attention, pozwalając modelowi równocześnie skupiać się na różnych aspektach informacji z różnych perspektyw. W Multi-Head Attention, proces uwagi jest wykonywany kilkukrotnie w równoległych "głowach". Każda głowa wykonuje Scaled Dot-Product Attention, ale z różnymi, niezależnymi zestawami wag (czyli z innymi zestawami zapytań, klu-czy i wartości). To pozwala każdej głowie na koncentrację na różnych aspektach informacji wejściowej. Na przykład jedna głowa może skupiać się na kontekście gramatycznym zdania, podczas gdy inna może analizować aspekty semantyczne. Po przeprowadzeniu uwagi w każdej głowie, wyjściowe wektory są łączone i przekazywane przez dodatkową warstwę gęstą (fully-connected layer), aby zintegrować informacje z wszystkich głów. To połączenie pozwala modelowi na bardziej kompleksowe i wszechstronne rozumienie danych wejściowych, co jest szczególnie użyteczne w przetwarzaniu i generowaniu języka naturalnego, gdzie różne aspekty informacji, takie jak znaczenie, kontekst czy styl, są równie ważne.
- Encoder-Decoder Attention: Moduł ten jest identyczny jak Multi-Head Attention, z tym że na wejściu przyjmuje parametry zarówno z kodera jak i dekodera. Queries brane są z dekodera, natomiast Keys i Values z kodera.
- Positional Encoding: Ponieważ Transformer nie używa rekurencji ani konwolucji, wprowadza się kodowanie pozycyjne do sekwencji wejściowej, aby uwzględnić informacje o kolejności słów lub innych elementów w sekwencji.



Rysunek 3.1. Architektura modelu Transformer

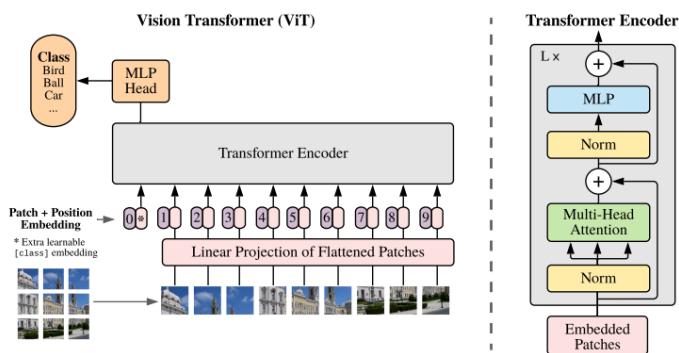
Artykuł **An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale** [7] przedstawia adaptację wyżej opisanego modelu Transformer do przetwarzania obrazów - Vision Transformer (ViT).

Podstawowe komponenty architektury to:

- Tokenizacja obrazu: W Vision Transformer, obraz wejściowy jest najpierw dzielony na małe, kwadratowe fragmenty (patche), zwykle o wymiarach 16x16 pikseli. Każdy z tych fragmentów jest traktowany jak słowo (token) w przetwarzaniu języka naturalnego. Następnie, każdy fragment obrazu jest przekształcany do jednowymiarowej postaci (flattening) i przekazywany przez warstwę liniową do zmapowania na wymaganą wielkość wektora cech, która odpowiada wymiarowi modelu Transformer.
- Token klasyfikacyjny: Do sekwencji tokenów obrazu dodawany jest specjalny token klasyfikacyjny (zazwyczaj oznaczany jako [CLS]), którego celem jest akumulacja informacji globalnej z całego obrazu. Stan tego tokena na wyjściu z transformera jest używany do klasyfikacji obrazu.
- Kodowanie pozycyjne: Ponieważ architektura transformera nie jest wrażliwa na kolejność danych wejściowych, do sekwencji tokenów dodaje się kodowania pozycyjne, aby zachować informację o lokalizacji fragmentów obrazu w przestrzeni.

3. Przegląd literatury

- Koder Transformer: Serce modelu stanowi stos warstw koderów Transformer, które przetwarzają sekwencję tokenów (wraz z tokenem klasyfikacyjnym i kodowaniami pozycyjnymi) za pomocą mechanizmu samouwagi (self-attention) i sieci feed-forward. Mechanizm uwagi pozwala modelowi na zrozumienie kontekstu i relacji między różnymi częściami obrazu.
- Głowa klasyfikacyjna: Na wyjściu z transformera, stan tokena klasyfikacyjnego jest przekazywany do "głowy klasyfikacyjnej", która dokonuje ostatecznej klasyfikacji obrazu.



Rysunek 3.2. Architektura modelu Vision Transformer

Drugi z modeli zaprezentowanych w pracy jest oparty właśnie na architekturze Transformer, który podobnie jak w jak w ViT na wejściu kodera przyjmuje sekwencję "tokenów" z obrazu. Jednak w odróżnieniu od ViT, w zaprezentowanym w niniejszej pracy modelu "tokenami" są wektory cech głębokich podane na wyjściu sieci splotowej.

4. Opis rozwiązania

4.1. Przygotowanie danych

Do trenowania oraz walidacji modeli użyłem zbioru danych DiffusionDB 2M, zawierającego 2 miliony par złożonych z obrazu i podpisu użytego do jego wygenerowania, jednak z powodu ograniczeń pamięci platformy Google Colab, zbiór został zredukowany do 1,084,245 obrazów oraz 884,825 unikalnych promptów.

Dane składają się z zestawu obrazków oraz pliku zawierającego metadane do każdego z nich. Rozdzielczością, która przeważa w zbiorze jest 512x512. Pierwszym krokiem obróbki danych było przeskalowanie obrazków do mniejszej rozdzielczości 224x224. Zabieg ten przyspieszył proces uczenia oraz pozwolił na wczytanie większej ilości danych do środowiska Google Colab. Następnie wyselekcjonowane zostały jedynie obrazki o stosunku wysokości do szerokości równym 1. Kolejnym krokiem było oczyszczenie zbioru z przykładów potencjalnie szkodliwych dla modelu. Zidentyfikowałem tu trzy kategorie w oparciu o metadane obrazu:

- NSTW = 2: Obrazki, w których parametr NSTW (Not Safe For Work) jest równy 2 zostały wyblurowane przez model Stable Diffusion 2.0
- CFG < 0: CFG jest hiperparametrem modelu i definiuje jak bardzo wygenerowany obraz jest podobny do prompta. Wartości poniżej 0 oznaczają przeciwnieństwo prompta. Przykładowo przy CFG = -1 przy użyciu prompta "superman, sharp focus, highly detailed..." model wygenerował talerz zupy
- Step < 10: Niska wartość hiperparametru Step może prowadzić do generowania rozmazanych obrazów [4]

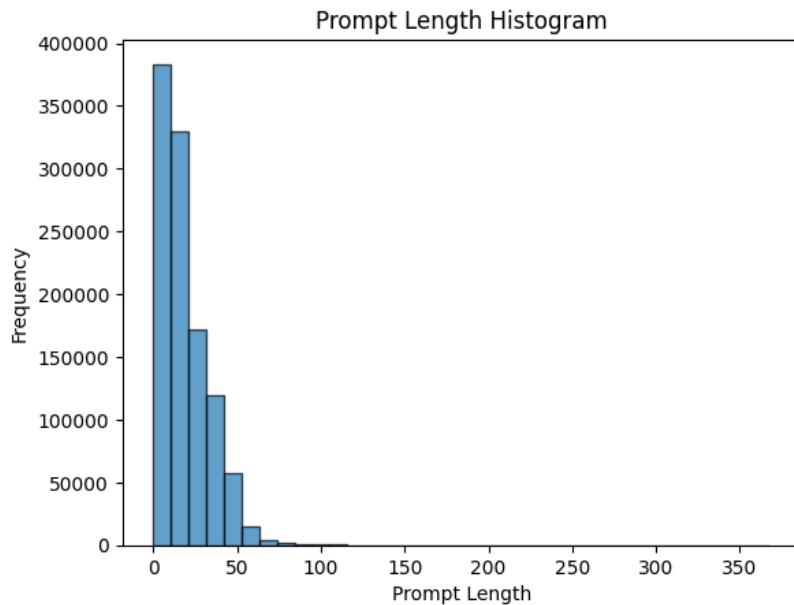
Następnie dane tekstowe zostały ztokenizowane za pomocą tokenizatora "punkt" dostępnego w bibliotece NLTK, usuwającego z fraz znaki interpunkcyjne. Usunięte zostały również tokeny zawierające cyfry. Z tak przygotowanych danych stworzony został słownik, przydatny później do tworzenia embeddingów. Poniżej przedstawiono przykładowy prompt przed i po tokenizacji.

Oryginalny prompt: "george washington by kehinde wiley and rembrandt : 5. photorealistic facial features, portrait, detailed face, ultra high details, large, close - up : 1." .

Prompt po tokenizacji: "george", "washington", "by", "kehinde", "wiley", "and", "rembrandt", "photorealistic", "facial", "features", "portrait", "detailed", "face", "ultra", "high", "details", "large", "close", "up".

4. Opis rozwiązania

Po analizie histogramu długości podpisów tekstowych po tokenizacji (Rys. [4.1]) zdecydowano na ograniczenie maksymalnej długości podpisu do 30 tokenów.



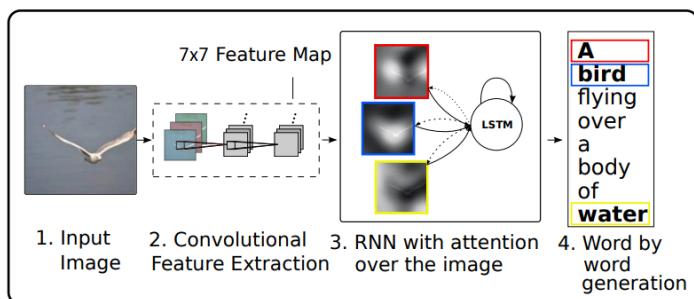
Rysunek 4.1. Histogram długości promptów

Tak przygotowane dane zostały podzielone na część treningową (90%), walidacyjną (5%) oraz testową (5%). Część walidacyjna posłużyła do oceny modelu podczas trenowania, część testowa natomiast (out-of-sample) do policzenia metryk przedstawionych w kolejnym rozdziale.

4.2. Model oparty o koder CNN-dekoder LSTM

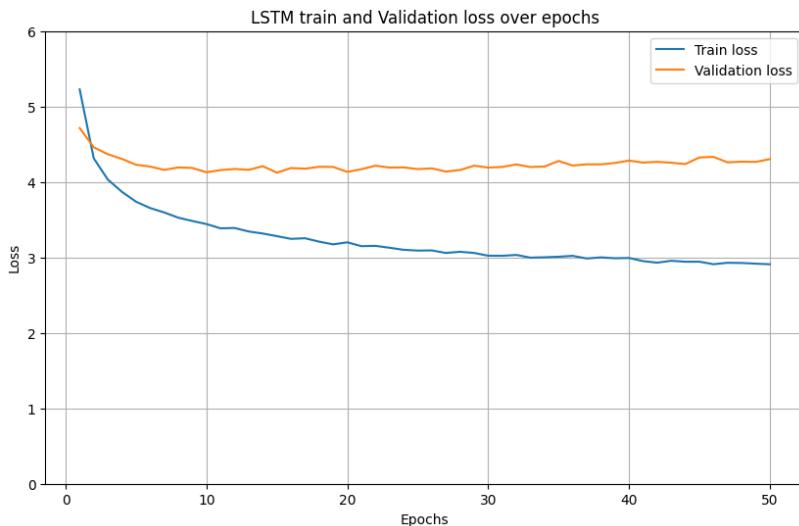
Pierwszy z modeli, których użyłem w swojej pracy oparty jest w pełni na rozwiązaniu zaproponowanym w artykule **Show, Attend and Tell: Neural Image Caption Generation with Visual Attention** [5] opisanym w poprzednim rozdziale. Część kodera stanowi tu pretrenowana sieć splotowa ResNet50 z odciętymi dwiema ostatnimi warstwami i zamrożonymi wagami. Zwraca ona mapę cech o wymiarach $7 \times 7 \times 2048$ spłaszczoną następnie do wymiarów 49×2048 . Dekoderem jest sieć rekurencyjna LSTM (Long Short-Term Memory), która sekwencyjnie generuje kolejne słowa. Ważny element stanowi mechanizm uwagi Bahdanau, który pozwala modelowi skupić się na różnych częściach obrazu podczas generowania każdego słowa opisu.

Schemat modelu przedstawiono poniżej.



Rysunek 4.2. Architektura kodera CNN-dekoder LSTM

Model był trenowany przez 50 epok, ale do ewaluacji wybrano checkpoint z epoki 27, ponieważ w następnych epokach funkcja straty na zbiorze walidacyjnym zaczęła wyraźnie rosnąć.



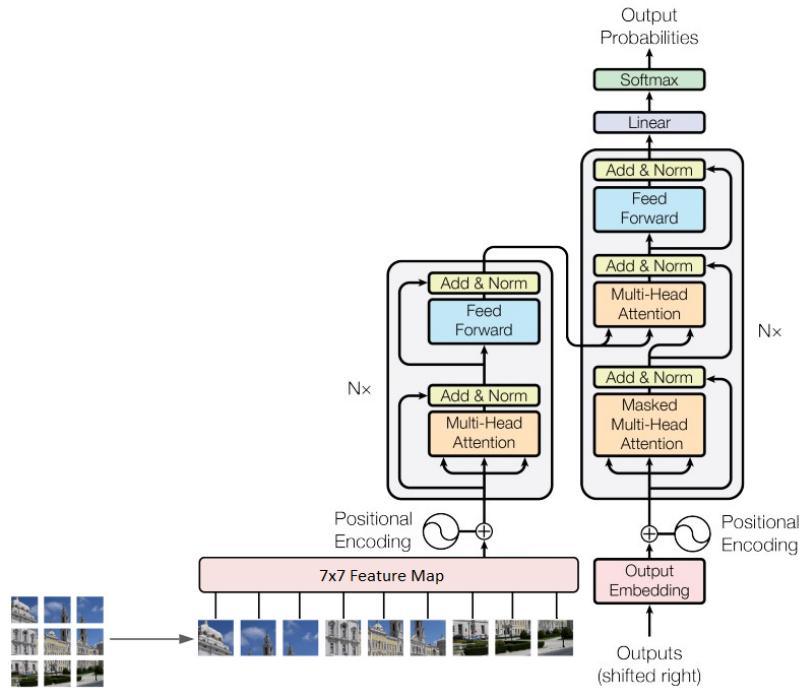
Rysunek 4.3. Krzywe uczenia modelu CNN-LSTM.

4.3. Model oparty o architekturę Transformer

Drugi z zaprezentowanych modeli oparty jest na architekturze Transformer przedstawionej w artykule **Attention Is All You Need** [6]. Na wejściu przyjmuje on sekwencję wektorów cech wyekstrahowanych z obrazu za pomocą pretrenowanego modelu ResNet50 z zamrożonymi wagami. Sieć splotowa ResNet50 zwraca mapę cech o wymiarach $7 \times 7 \times 2048$, która następnie jest spłaszczana do wymiarów 49×2048 . Tak przetworzone dane są przekazywane do kodera Transformera jako 49 tokenów, każdy o wymiarze 2048. Na wejściu dekodera natomiast podawany jest tekst, na podstawie którego obraz został wygenerowany. Można zatem powiedzieć, że tak skonstruowany model zamiast tłumaczyć tekst z jednego języka na drugi "tłumaczy" obraz na tekst.

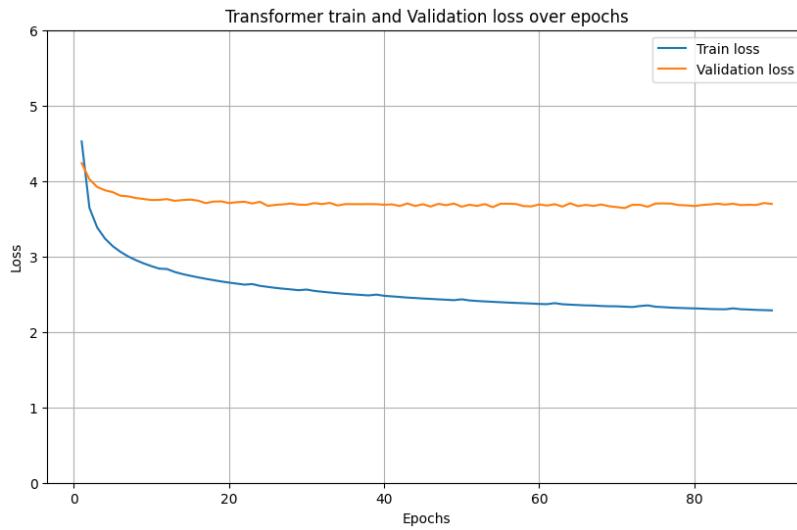
4. Opis rozwiązania

Architekturę modelu opartego o architekturę Transformer przedstawiono poniżej.



Rysunek 4.4. Architektura modelu opartego o Transformer

Model był trenowany przez 90 epok, ale do ewaluacji wybrano checkpoint z epoki 71, aby uniknąć nadmiernego dopasowania do danych treningowych. Do treningu użyto karty graficznej NVIDIA V100 dostępnej na platformie Google Colab. Cała procedura zajęła około 180 godzin.



Rysunek 4.5. Krzywe uczenia modelu opartego o architekturę Transformer

5. Wyniki ewaluacji eksperymentalnej

5.1. Metryki użyte do ewaluacji modeli

Do ewaluacji modeli w celu ich porównania użyłem metryk szeroko stosowanych w tłumaczeniu maszynowym takich jak BLEU, METEOR oraz BERT Score.

BLEU, czyli (BiLingual Evaluation Understudy) jest metryką używaną do oceny jakości tłumaczenia maszynowego. Porównuje ona maszynowo przetłumaczony tekst z jednym lub kilkoma tłumaczeniami referencyjnymi dokonanymi przez człowieka. BLEU ocenia jakość tłumaczenia na podstawie tego, jak wiele słów i fraz z tłumaczenia maszynowego pokrywa się z tłumaczeniami referencyjnymi.

Metryka BLEU koncentruje się głównie na precyzyji leksykalnej, biorąc pod uwagę zarówno pojedyncze słowa (n-gramy na poziomie 1), jak i dłuższe ciągi słów (n-gramy na wyższych poziomach). BLEU oblicza tzw. "modified n-gram precision", która uwzględniaczęstość występowania n-gramów w tłumaczeniu i w tekście referencyjnym. Dodatkowo, BLEU stosuje korektę dla krótkich tłumaczeń (tzw. "brevity penalty"), która pomaga uniknąć nadmiernej oceny tłumaczeń, które są zbyt zwięzłe, ale precyzyjne.

Wynik BLEU jest wyrażony jako wartość procentowa, gdzie wyższy wynik wskazuje na większe podobieństwo do tłumaczenia ludzkiego. BLEU jest szeroko stosowany w branży i badaniach nad tłumaczeniem maszynowym, ze względu na jego prostotę i zdolność do szybkiej oceny dużej liczby tłumaczeń. Jednakże, BLEU ma również swoje ograniczenia, np. nie jest w stanie w pełni ocenić płynności i gramatycznej poprawności tłumaczenia, ani nie uwzględnia pełnego kontekstu semantycznego. [8]

METEOR, czyli Metric for Evaluation of Translation with Explicit Ordering, jest metryką używaną do oceny jakości tłumaczenia maszynowego. Stanowi alternatywę dla bardziej znanej metryki BLEU i jest uważana za bardziej zaawansowaną w niektórych aspektach.

Głównym celem METEOR jest lepsze dopasowanie do ludzkiej oceny jakości tłumaczenia. Metryka ta nie tylko porównuje ilość wspólnych słów między tłumaczeniem a referencyjnym tekstem ludzkim, ale również uwzględnia synonimy i pararazy, co pozwala na bardziej elastyczną ocenę. METEOR korzysta także z analizy harmonogramu wyrazów (ang. word alignment) między tłumaczeniem a tekstem referencyjnym, uwzględniając kolejność słów i ich znaczenie.

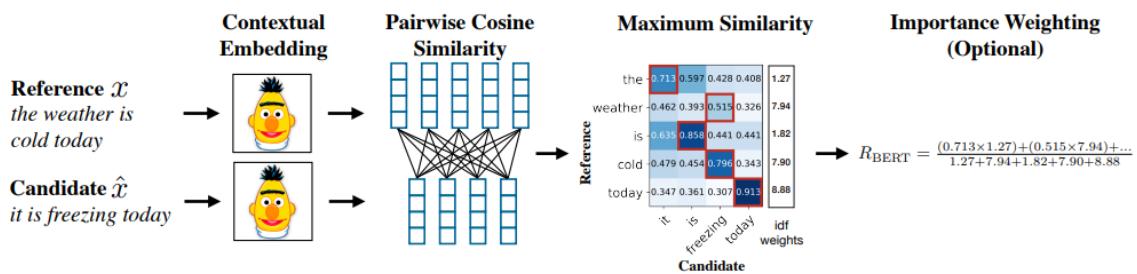
Wynik METEOR jest obliczany na podstawie trzech głównych komponentów: precyzyi (odsetek słów w tłumaczeniu, które znajdują się w tekście referencyjnym), pełności (odsetek słów z tekstu referencyjnego, które znajdują się w tłumaczeniu) i harmonogramu wyrazów. Wysoka ocena METEOR sugeruje, że tłumaczenie jest zarówno dokładne, jak i naturalne z punktu widzenia języka docelowego. Metryka ta jest szczególnie przydatna w przypadkach, gdy konieczne jest uwzględnienie różnych możliwych tłumaczeń danego fragmentu tekstu. [9]

5. Wyniki ewaluacji eksperimentalnej

BERT Score to metryka oceny jakości tłumaczenia maszynowego i innych zadań generowania tekstu, która wykorzystuje pre-trenowane modele językowe oparte na BERT (Bidirectional Encoder Representations from Transformers) do oceny podobieństwa semantycznego między tekstem referencyjnym a wygenerowanym. W przeciwieństwie do tradycyjnych metryk, takich jak BLEU, które koncentrują się na precyzji leksykalnej, BERT-Score bada podobieństwo na poziomie reprezentacji wektorowych słów lub tokenów, uwzględniając kontekst i znaczenie semantyczne.

Metryka ta działa poprzez obliczanie podobieństwa cosinusowego między wektorami reprezentującymi słowa w tekście referencyjnym i wygenerowanym. Wysoki wynik BERT-Score wskazuje, że teksty są podobne nie tylko pod względem słów, ale także ich znaczenia w kontekście zdania. BERT Score jest bardziej elastyczna i może lepiej oceniać jakość tłumaczeń w przypadkach, gdzie dosłowna zgodność słów nie jest idealnym wskaźnikiem jakości, na przykład w tłumaczeniach o większej swobodzie stylistycznej lub semantycznej.

Przykładowo "mighty kitten" i "powerful cat" osiągną niską ocenę BLEU, ponieważ obie frazy zawierają różne słowa, ale ocena BERT Score powinna być już na wysokim poziomie, bo są semantycznie podobne. [10]



Rysunek 5.1. Sposób obliczania metryki BERT Score

5.2. Ewaluacja modeli

Model	BLEU	METEOR	BERT Score F1
LSTM	1.1	8.1	44.1
Transformer	5.0	14.1	51.6

Tabela 5.1. Tabela metryk

5.3. Demonstracja działania modeli na obrazach ze zbioru testowego

True prompt: captain marvel played by scarlett johansson

Transformer: a still of scarlett johansson as captain america

CNN-LSTM: emma watson as captain america

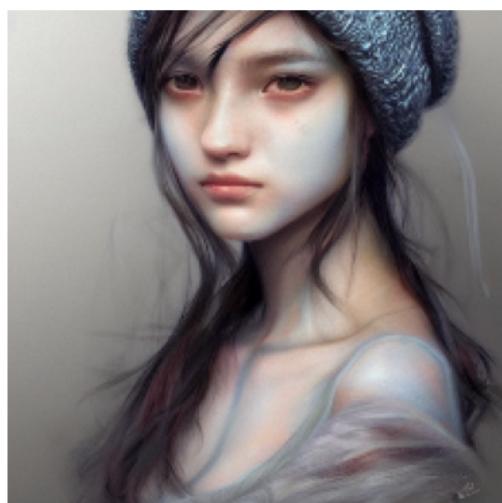


Rysunek 5.2. Obraz ze zbioru testowego nr 1

True prompt: portrait of a starving ai artist detailed clothing beanie concept art artstation detailed luminescent digital painting by alan lee and artgerm beautiful elegant exquisite masterpiece

Transformer: portrait of a beautiful elegant young adult with long white hair full round face freckles blue eyes thin nose thin lips white hair thin eyebrows thin nose white

CNN-LSTM: a portrait of a beautiful woman



Rysunek 5.3. Obraz ze zbioru testowego nr 2

5. Wyniki ewaluacji eksperimentalnej

True prompt: fallout weapons kim jung gi black and white

Transformer: cybermen

CNN-LSTM: a of of



Rysunek 5.4. Obraz ze zbioru testowego nr 3

True prompt: a picture of an elephant hanging from a tree a surrealist painting by storm thorgerson shutterstock contest winner massurrealism surrealist whimsical hyper realistic

Transformer: elephant riding a elephant

CNN-LSTM: a elephant with a



Rysunek 5.5. Obraz ze zbioru testowego nr 4

5.4. Demonstracja działania modeli na obrazach wygenerowanych przez autora

Poniższe obrazy zostały wygenerowane przy użyciu biblioteki *diffusers* opublikowanej przez Hugging Face. Umożliwia ona używanie różnego rodzaju modeli dyfuzyjnych za pomocą kilku linijek kodu.

True prompt: purple cat sitting in a lunar lander on the surface of the moon in the style of greg rutkowski

Transformer: a blue and white cat sitting on the moon with stars in the background

CNN-LSTM: a black hole in the shape of a a



Rysunek 5.6. Obraz wygenerowany przez autora nr 1

5. Wyniki ewaluacji eksperimentalnej

True prompt: five monkeys in dresses dancing in the jungle in the style of renoir

Transformer: a group of people dancing in a jungle party in the style of renoir

CNN-LSTM: a painting of a a



Rysunek 5.7. Obraz wygenerowany przez autora nr 2

True prompt: indiana jones riding a cow with a magic wand in his hand on the street of new york

Transformer: a cow riding a bull in new york city

CNN-LSTM: a photo of a a

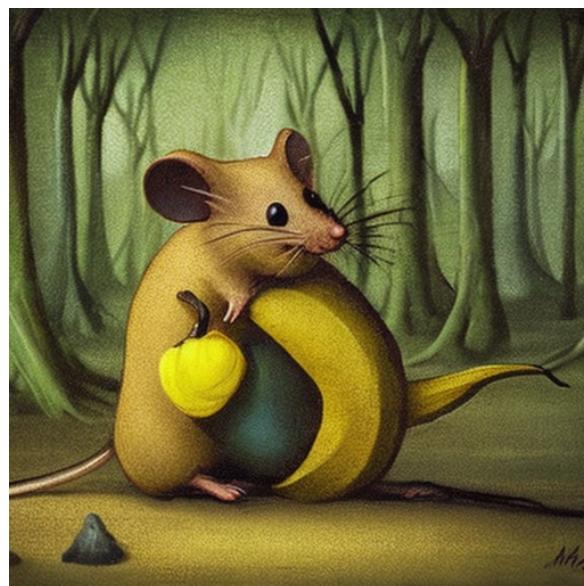


Rysunek 5.8. Obraz wygenerowany przez autora nr 3

True prompt: banana shaped mouse in the fantasy forest in the style of rembrandt

Transformer: maus in forest by rivuletpaper rivuletpaper art mouse guard by lily seika jones rivuletpaper art top cinematic lighting cinematic mood very detailed shot in canon high detail mood

CNN-LSTM: a painting of a a



Rysunek 5.9. Obraz wygenerowany przez autora nr 4

True prompt: walter white cooking dinner for his family at the swimming pool party

Transformer: walter white cooking spaghetti in a diner

CNN-LSTM: a still of a in a



Rysunek 5.10. Obraz wygenerowany przez autora nr 5

5. Wyniki ewaluacji eksperimentalnej

True prompt: darth vader ordering beer in the bar on tatooine highly detailed

Transformer: darth vader drinking beer at starbucks cinematic lighting

CNN-LSTM: a still of a in a



Rysunek 5.11. Obraz wygenerowany przez autora nr 6

True prompt: a fish running a marathon on the beach highly detailed

Transformer: a beautiful painting of a trout swimming in the ocean

CNN-LSTM: a fish swimming in a pool of water



Rysunek 5.12. Obraz wygenerowany przez autora nr 7

6. Podsumowanie

Model oparty na architekturze Transformerera osiąga znaczco lepsze rezultaty na zbiore testowym niż koder CNN-dekoder LSTM, który często ma problem z wygenerowaniem koherentnego tekstu, ale jest wyraźnie gorszy niż modele state-of-the-art trenowane i testowane na zbiorach Flickr8k lub COCO. Nie można jednak bezpośrednio porównywać modeli trenowanych na danych będącym zbiorem obrazów opisanych przez ludzi z modelami trenowanymi na przykładach wygenerowanych przez model i opisanych promptami. Z tego powodu ocena wyników modelu jest dość problematyczna. Z jednej strony model oparty na architekturze Transformerera przeważnie nie najgorzej opisuje zawartość obrazu, z drugiej jednak wygenerowane podpisy co najwyżej częściowo pokrywają się z oryginalnym promptem.

Z pewnością jednym z problemów pracy jest wybór zbioru danych oraz powiązanego z nim modelu generatywnego. Stable Diffusion 2.0 jest modelem realtywnie małym i nie dostrojonym w porównaniu do najnowszych wersji Midjourney czy DALL-E. Powoduje to, że wiele ze składowych prompta jest po prostu pomijanych w wygenerowanym obrazie, co dobrze widać w poprzednim rozdziale. Gdybym zacząał projekt od początku, z pewnością użyłbym zbioru obrazków wygenerowanych przez jeden z dobrze zestrojonych nowszych modeli. Natomiast jeśli chodzi o sam zbiór danych, to pomimo że jest on relatywnie duży, to występują w nim grupy bardzo podobnych do siebie promptów, często różniących się jedynie jednym słowem. Mogło to spowodować dopasowanie pewnych słów do tych cech obrazów, z którymi nie powinny mieć nic wspólnego, czego objaw można zaobserwować na przykładzie wygenerowanego podpisu do rysunku 5.9.

7. Bibliografia

- [1] Kevin Roose, An AI-Generated Picture Won an Art Prize. Artists Aren't Happy, <https://www.nytimes.com/2022/09/02/technology/ai-artificial-intelligence-artists.html>, 2022
- [2] Pierre Chambon, Christian Bluethgen, Curtis P. Langlotz and Akshay Chaudhari, Adapting Pretrained Vision-Language Foundational Models to Medical Imaging Domains, 2022
- [3] Jonathan Ho, William Chan, Chitwan Saharia, Jay Whang, Ruiqi Gao, Alexey Grishchenko, Diederik P. Kingma, Ben Poole, Mohammad Norouzi, David J. Fleet and Tim Salimans, Imagen Video: High Definition Video Generation with Diffusion Models, 2022
- [4] Zijie J. Wang, Evan Montoya, David Munechika, Haoyang Yang, Benjamin Hover and Duen Horng Chau, DIFFUSIONDB: A Large-scale Prompt Gallery Dataset for Text-to-Image Generative Models, 2022
- [5] Kelvin Xu, Jimmy Ba, Ryan Kiros, Kyunghyun Cho, Aaron Courville, Ruslan Salakhutdinov, Richard Zemel, Yoshua Bengio and Show, Attend and Tell: Neural Image Caption Generation with Visual Attention, 2015
- [6] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser and Illia Polosukhin, Attention Is All You Need, 2017
- [7] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit and Neil Houlsby, An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale, 2020
- [8] Kishore Papineni, Salim Roukos, Todd Ward and Wei-Jing Zhu, BLEU: a Method for Automatic Evaluation of Machine Translation, 2002
- [9] Satanjeev Banerjee and Alon Lavie, METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments, 2005
- [10] Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q. Weinberger and Yoav Artzi, BERTScore: Evaluating Text Generation with BERT, 2019

Spis rysunków

1.1	Théâtre D'opéra Spatial	9
2.1	Obraz ze zbioru testowego nr 1	11
2.2	Obraz ze zbioru testowego nr 2	12
2.3	Obraz ze zbioru testowego nr 3	12
3.1	Architektura modelu Transformer	15
3.2	Architektura modelu Vision Transformer	16
4.1	Histogram długości promptów	18
4.2	Architektura koder CNN-dekoder LSTM	19
4.3	Krzywe uczenia modelu CNN-LSTM.	19
4.4	Architektura modelu opartego o Transformer	20
4.5	Krzywe uczenia modelu opartego o architekturę Transformer	20
5.1	Sposób obliczania metryki BERT Score	22
5.2	Obraz ze zbioru testowego nr 1	23
5.3	Obraz ze zbioru testowego nr 2	23
5.4	Obraz ze zbioru testowego nr 3	24
5.5	Obraz ze zbioru testowego nr 4	24
5.6	Obraz wygenerowany przez autora nr 1	25
5.7	Obraz wygenerowany przez autora nr 2	26
5.8	Obraz wygenerowany przez autora nr 3	26
5.9	Obraz wygenerowany przez autora nr 4	27
5.10	Obraz wygenerowany przez autora nr 5	27
5.11	Obraz wygenerowany przez autora nr 6	28
5.12	Obraz wygenerowany przez autora nr 7	28

Spis tabel

5.1	Tabela metryk	22
-----	-------------------------	----