Od pikseli do obrazów: Generowanie obrazów

Generowanie obrazów to jedno z wielu zastosowań generatywnej sztucznej inteligencji (AI), pozwalające na tworzenie unikalnych i realistycznych treści za pomocą zaledwie jednego polecenia. Przedsiębiorstwa coraz częściej wdrażają generatywną sztuczną inteligencję (AI), aby opracowywać innowacyjne rozwiązania do generowania i edycji obrazów, co doprowadziło do wielu innowacyjnych zastosowań – od architektury opartej na AI do innowacyjnych projektów budynków, przez projektowanie mody, generowanie awatarów, wirtualne przymierzanie ubrań, po wirtualnych pacjentów w szkoleniach medycznych, by wymienić tylko kilka. Towarzyszą im ekscytujące produkty, takie jak Microsoft Designer i Adobe Firefly, które zostaną omówione. W poprzednich częściach omówiliśmy podstawy generatywnej sztucznej inteligencji (AI) oraz technologię, która umożliwia generowanie tekstu, w tym uzupełnianie i czaty. Tu jednak zmienimy temat i zbadamy, jak generatywną sztuczną inteligencję można wykorzystać do tworzenia i modyfikowania obrazów. Zobaczymy, jak prosty jest proces tworzenia obrazów i zwrócimy uwagę na niektóre trudności związane z ich poprawnym tworzeniem. Początkowo rozdział ten koncentruje się na zrozumieniu metod generatywnej sztucznej inteligencji, które ułatwiają generowanie nowych obrazów, oraz na ogólnym procesie pracy, jaki musi uwzględnić przedsiębiorstwo. Zastosowania tych technik są ogromne i mogą być szczególnie przydatne w sektorach e-commerce, rozrywki i opieki zdrowotnej. Dodatkowo, przyjrzymy się różnym produktom i usługom generatywnej sztucznej inteligencji do manipulacji obrazami. Zaczynajmy!

Modele widzenia

Generatywne modele widzenia AI mogą generować realistyczne, nowe obrazy i nowatorskie koncepcje na podstawie podpowiedzi. Zacznijmy od omówienia kilku przypadków użycia w przedsiębiorstwach i przykładów, jak te generatywne modele widzenia AI mogą pomóc:

* Tworzenie i edycja treści — Istnieje wiele przypadków użycia w różnych branżach, w których generatywne modele widzenia AI mogą pomóc specjalistom z branży medialnej i marketingowej w generowaniu nowych tematów i scenariuszy, usuwaniu zbędnych lub niechcianych elementów z obrazów lub stosowaniu transferu stylu. Konkretne przypadki użycia różnią się w zależności od branży.

* Opieka zdrowotna — Istnieje wiele przypadków użycia AI generującej obrazy w dziedzinie zdrowia, od edukacji i szkolenia studentów medycyny lub stosowania nowych technik (patrz następny punkt) po poprawę diagnozy i rokowania pacjenta poprzez pomoc w ulepszaniu i poprawianiu jakości obrazów medycznych. Przyspiesza również odkrywanie i rozwój leków poprzez analizę nowych cząsteczek, złożonych interakcji molekularnych i ich przewidywań oraz optymalizację formulacji i syntezy.

* Edukacja — Możemy tworzyć interaktywne wizualizacje na bieżąco, w oparciu o postępy studenta i jego aktualną wiedzę. Obejmuje to realistyczne i zróżnicowane scenariusze, symulacje szkoleniowe wykorzystujące rozszerzoną analizę danych oraz pomoc w poprawie jakości nauczania zarówno nauczyciela, jak i ucznia.

* Badania i rozwój — Możemy stworzyć bardziej interpretowalną wizualną reprezentację złożonych struktur danych i relacji, które w innym przypadku mogłyby nie być oczywiste. Te kluczowe elementy pomagają tworzyć nowe projekty produktów w oparciu o trendy, unikalne elementy wizualne, branding i układy, a także odkrywać subtelne wzorce w danych.

* Marketing — generatywne modele wizji AI generują specyficzne wizualizacje dostosowane do konkretnej osoby lub grupy demograficznej, które mogą również obejmować różne zestawy wizualizacji do testów A/B w celu zrozumienia skuteczności kampanii marketingowych.

* Produkcja — generatywne modele wizji AI umożliwiają szybką iterację i wizualizację nowych materiałów i komponentów, w tym procesu montażu.

* Personalizacja — Ten horyzontalny przypadek użycia może obejmować różne wymiary, umożliwiając generowanie spersonalizowanych wizualizacji, na przykład w środowiskach e-commerce, gdzie klient może wizualizować przedmioty, treści, ubrania itp., aby tworzyć wysoce spersonalizowane awatary do gier i platform społecznościowych. Wreszcie moda i dziedziny kreatywne tworzą nowe wzory, układy, projekty ubrań i mebli.

Oto kilka przykładów, jak generować i ożywiać niektóre z tych treści:

* Treści kreatywne — generatywne modele wizji AI mogą tworzyć nowe i różnorodne obrazy lub filmy do celów artystycznych, rozrywkowych lub marketingowych. Niektóre z nich tworzą realistyczne twarze osób, które wydają się prawdziwe, ale nie istnieją, lub modyfikują istniejące twarze, uwzględniając różne cechy, takie jak wiek, płeć, fryzura itp. Rysunek przedstawia misia pandę wygenerowanego z truskawek.

* Edycja obrazu, ulepszanie treści i transfer stylu — generatywne modele wizji AI mogą służyć do ulepszania istniejących obrazów. Mogą one uwzględniać różne artefakty, takie jak poprawa rozdzielczości i jakości oraz usuwanie niepożądanych elementów. Możemy również wykorzystać styl i technikę jednego obrazu i przenieść je na inny. Na przykład rysunek 2 przedstawia obraz olejny przedstawiający Space Needle w Seattle w stylu Vincenta van Gogha.

* Dane syntetyczne — Możemy tworzyć realistyczne, ale syntetyczne obrazy za pomocą generatywnych modeli widzenia AI. Te syntetyczne obrazy mogą być wykorzystywane jako dane treningowe i walidacyjne dla innych modeli AI. Na przykład strona https://thispersondoesnotexist.com/ generuje twarze osób, które nie istnieją w rzeczywistości. Dane syntetyczne wiążą się z wyzwaniami; omówimy je w dalszej części , przy okazji omawiania wyzwań związanych z generatywną AI.

* Inżynieria i projektowanie generatywne — Możemy generować nowe opcje projektowe, obejmujące nowe obiekty i struktury, które pomogą nam zoptymalizować określone kryteria lub ograniczenia, takie jak funkcjonalność, wydajność czy estetyka. Modele te mogą generować unikalne, nowatorskie projekty produktów lub zasobów cyfrowych, redukując czas i zasoby poświęcane na ręczne projektowanie. Rysunek 3 przedstawia krzesło zoptymalizowane pod kątem różnych cech projektowych, takich jak materiał i estetyka. Krzesła te mają unikalne i futurystyczne kształty, różniące się od krzeseł konwencjonalnych.

Cztery główne typy architektury generatywnego modelu sztucznej inteligencji (AI) umożliwiają te przypadki użycia i przykłady: autoenkodery wariacyjne (VAE), generatywne sieci przeciwstawne (GAN), modele dyfuzyjne i transformatory wizji. Każda technika ma swoje mocne i słabe strony, a my przedstawiamy odpowiednie podejście do scenariusza, w którym może być wykorzystana:

* Autoenkodery wariacyjne — VAE generują realistyczne, ale proste obrazy zwierząt, twarzy i innych obiektów. Są one dobre w scenariuszach wymagających generowania danych, czyli nowych punktów danych podobnych do oryginału, ale z wariacjami. Ta właściwość pozwala również na wykorzystanie VAE w systemach wykrywania anomalii i rekomendacji.

* Generatywne sieci przeciwstawne — GAN są wykorzystywane w scenariuszach, w których dane są złożone i zróżnicowane oraz wymagają wysokiego poziomu realizmu. Dzięki temu nadają się do generowania wysokiej jakości obrazów, augmentacji danych i transferu stylów.

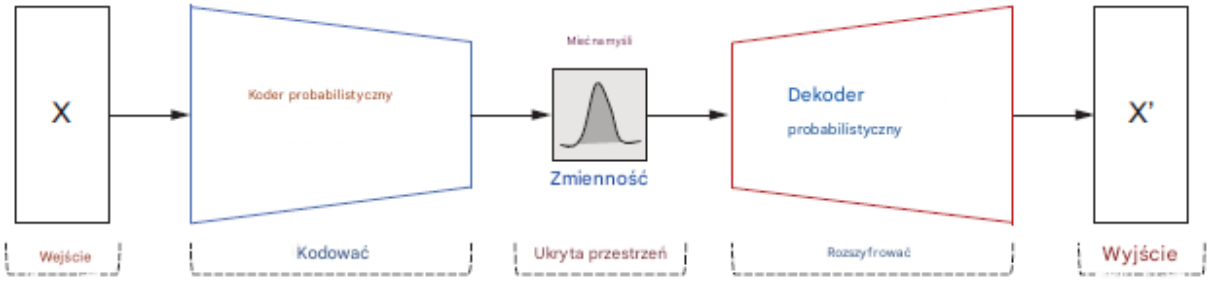
* Dyfuzja — modele oparte na dyfuzji są używane w scenariuszach, w których dane są wielowymiarowe i ciągłe, a my musimy modelować złożony rozkład danych z zachowaniem jakości, przy czym szybkość generowania nie ma znaczenia. Modele te dobrze sprawdzają się w generowaniu mowy i wideo, o których wspomnę później.

* Transformatory wizji — świetnie sprawdzają się, gdy chcemy generować obrazy oparte na sekwencjach, są bardzo elastyczne i adaptowalne do wielu zadań; wymagają znacznych zasobów obliczeniowych.

Przyjrzyjmy się bliżej każdej z tych architektur.

Autoenkodery wariacyjne

Autoenkodery wariacyjne (VAE) to specyficzny model generatywny, który odgrywa kluczową rolę. Reprezentują złożone rozkłady danych, łącząc aspekty głębokiego uczenia się, teorii prawdopodobieństwa i mechaniki statystycznej. Autoenkodery wariacyjne (VAE) składają się z dwóch sieci neuronowych: kodera i dekodera .



Koder odwzorowuje obraz wejściowy na niskowymiarowy wektor utajony (przestrzeń utajoną), który rejestruje jego istotne cechy. Nie tylko znajduje pojedynczy punkt w przestrzeni utajonej, ale także może znaleźć rozkład. Natomiast dekoder pobiera próbki z przestrzeni utajonej i rekonstruuje oryginalny obraz wejściowy, dodając jednocześnie pewną losowość, aby uczynić go bardziej zróżnicowanym. Ta losowość pozwala nam dodawać nowe punkty danych, takie jak dane wejściowe. Następujące dwa parametry definiują stan utajony: średnia i wariancja. Jak sama nazwa wskazuje, średnia to średnia wartość stanu utajonego, a wariancja to różnica miary stanu utajonego od średniej. VAE wykorzystuje te parametry do próbkowania różnych stanów utajonych z rozkładu normalnego – funkcji matematycznej opisującej prawdopodobieństwo wystąpienia różnych wartości. Próbkując różne stany utajone, VAE może generować różne dane wyjściowe, które są podobne do danych wejściowych. Mechanika statystyczna pozwala nam na stworzenie ram do wnioskowania o rozkładzie prawdopodobieństwa zmiennych w zmiennych utajonych na podstawie danych obserwatora.

Niektóre z kluczowych zastosowań, jakie umożliwiają VAE, to:

* Generowanie obrazu — VAE są szeroko wykorzystywane do generowania obrazów w celu tworzenia unikalnych obrazów, które mają podobieństwa do danych treningowych, niezależnie od tego, czy są to twarze ludzkie, projekty mody czy dzieła sztuki.

* Rekonstrukcja obrazu i inpainting — Ucząc się podstawowej struktury danych obrazu, VAE mogą rekonstruować brakujące lub uszkodzone fragmenty obrazu. Te właściwości rekonstrukcji lub uzupełniania brakujących aspektów są niezwykle przydatne w niektórych dziedzinach, takich jak obrazowanie medyczne, renowacja starych i archeologicznie ważnych fotografii itd.

* Transfer stylu — VAE pozwalają nam oddzielić treść obrazu od stylu i przenieść elementy stylistyczne z jednego obrazu do drugiego, jak pokazano na rysunku 2.

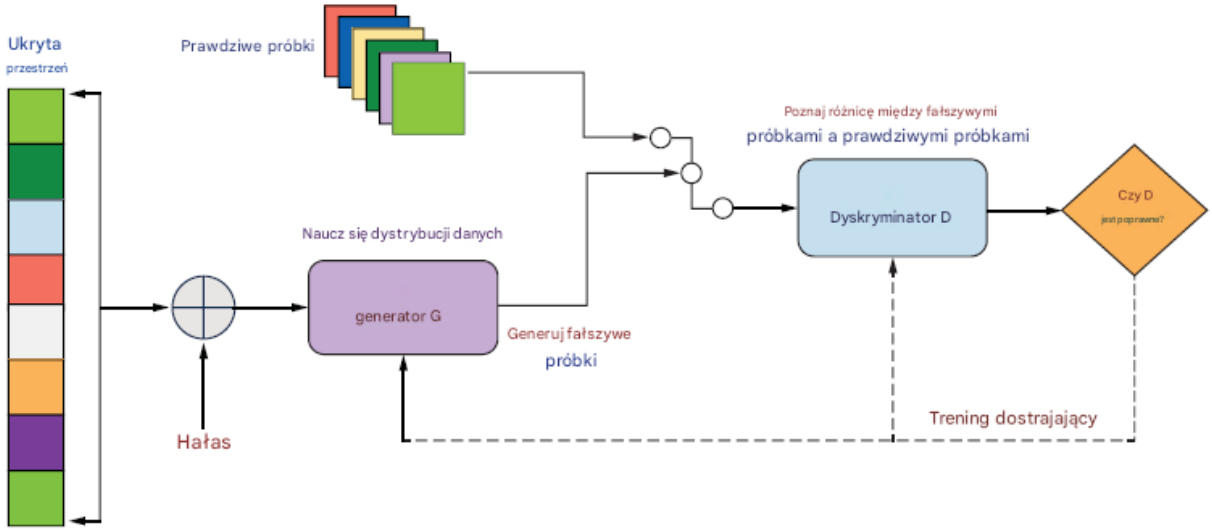
* Semantyczna manipulacja obrazem — jest podobna do rekonstrukcji obrazu. Dzięki poznanej przestrzeni ukrytej, VAE mogą zapewnić nam znacznie bardziej precyzyjną kontrolę nad cechami generowanych obrazów poprzez modyfikowanie określonych aspektów generowanych obrazów, takich jak mimika twarzy, bez wpływu na inne, niezwiązane z nimi cechy.

Choć potężne, modele VAE mają wady, takie jak rozmycie, brak zróżnicowania i trudności w modelowaniu złożonych rozkładów. Ich trenowanie może być wymagające i niestabilne, co prowadzi do załamania się modów. Niezależnie od tych wyzwań, osiągnięcia i potencjał modeli VAE pozostają w czołówce badań nad wizją sztucznej inteligencji, opierając się na złożonych relacjach między danymi, matematyką i kreatywnością.

UWAGA: Przestrzeń utajona reprezentuje złożone dane w prostszy i bardziej zrozumiały sposób. Wyobraź sobie ją jako mapę, na której podobne elementy znajdują się blisko siebie, a różne daleko od siebie. Pomaga nam to znaleźć podobieństwa, generować nowe dane i lepiej je rozumieć.

Generatywne sieci adwersarskie

GAN-y należą do najpopularniejszych technik tworzenia obrazów za pomocą generatywnej sztucznej inteligencji. Składają się one z dwóch sieci neuronowych: generatora, który tworzy nowe przykłady, oraz dyskryminatora, który próbuje odróżnić przykłady rzeczywiste od wygenerowanych. Generator próbuje tworzyć fałszywe obrazy, które wyglądają jak prawdziwe, na podstawie szumu losowego lub danych wejściowych, takich jak tekst czy szkice. Dyskryminator bierze obrazy rzeczywiste od fałszywych i próbuje je rozróżnić. Obie sieci są trenowane poprzez równoczesną rywalizację w sposób oparty na teorii gier, aby z czasem poprawić swoją wydajność. GAN-y działają w oparciu o grę typu min–maks, w której generator stara się zmaksymalizować błędy dyskryminatora, a dyskryminator stara się je zminimalizować. GAN-y wykorzystują podpowiedź jako dane wejściowe dla generatora wraz z szumem losowym. Następnie generator generuje obraz, który próbuje dopasować się do podpowiedzi i wysyła go do dyskryminatora. Dyskryminator porównuje wygenerowany obraz z obrazem rzeczywistym pochodzącym z tego samego podpowiedzi, przyznając ocenę wskazującą, jak realistyczny jest jego zdaniem obraz. Wynik jest następnie wykorzystywany do aktualizacji wag obu sieci za pomocą propagacji wstecznej i metody gradientu prostego. Proces ten jest powtarzany, aż generator będzie w stanie wygenerować obrazy spełniające wymagania i oszukać dyskryminator. Celem GAN jest, aby generator generował realistyczne obrazy, które mogą oszukać dyskryminator. Rysunek 5 przedstawia architekturę modelu GAN w ogólnym zarysie.



Przestrzeń utajona reprezentuje możliwe dane wejściowe dla generatora, a dostrajanie pozwala na dostosowanie parametrów dyskryminatora i generatora.

Sieci GAN oferują wiele podobnych zastosowań, takich jak VAE, ale są szczególnie przydatne w:

* Generowaniu obrazu — Tworzenie realistycznych obrazów z szumu, z konkretnymi zastosowaniami w rozrywce, projektowaniu i sztuce, umożliwia generowanie obrazów wysokiej jakości.

* Transferze stylów — Umożliwianie przenoszenia stylów artystycznych z jednego obrazu do drugiego; działa to tak samo, jak w przypadku VAE.

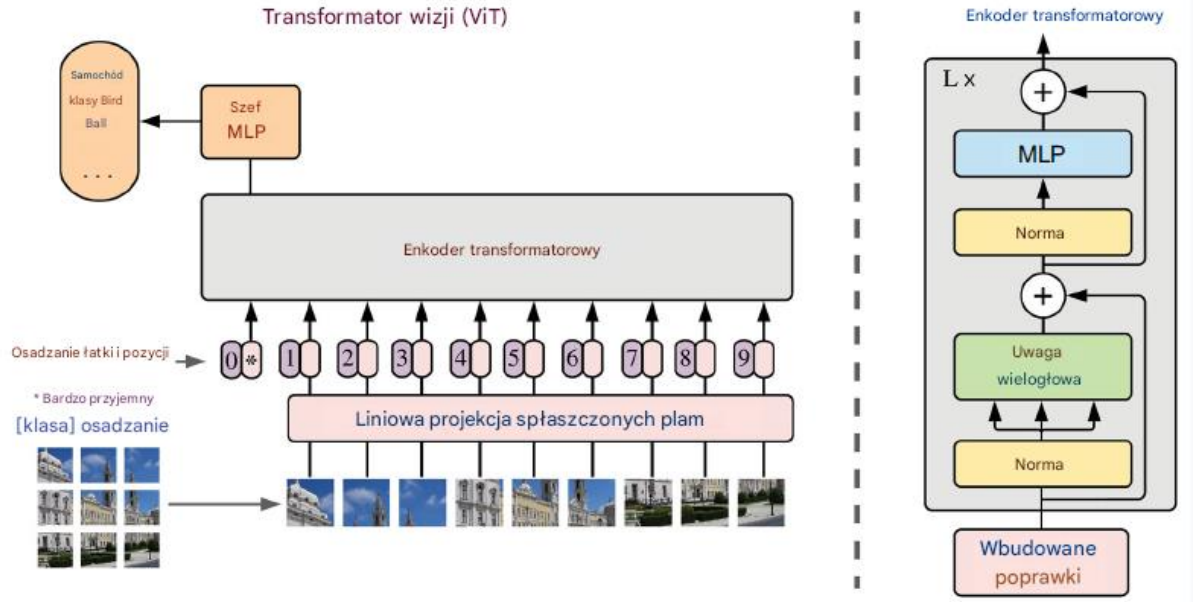
* Superrozdzielczości — Sieci GAN mogą pomóc w zwiększeniu rozdzielczości, dzięki czemu obrazy są bardziej szczegółowe i wyraźniejsze. Jest to bardzo przydatne w niektórych branżach, takich jak obrazowanie medyczne i kosmiczne.

* Rozszerzaniu danych — Podobnie jak sieci VAE do tworzenia danych syntetycznych, sieci GAN pomagają w tworzeniu danych treningowych dla przypadków brzegowych lub w przypadku braku wystarczającej ilości danych lub ich zróżnicowania.

Sieci GAN mogą generować wysokiej jakości obrazy, których nie da się odróżnić od obrazów rzeczywistych. Mają jednak wady, takie jak załamanie modów (tj. model wielokrotnie generuje ten sam wynik), niestabilność i trudności z kontrolowaniem wyników. Budzą one również wątpliwości natury etycznej, ponieważ mogą być dość łatwo wykorzystane do tworzenia deepfake'ów, które mogą prowadzić do naruszenia prywatności, potencjalnej dezinformacji i przekłamań. Wreszcie, podobnie jak wiele innych modeli sztucznej inteligencji, sieci GAN mogą nieumyślnie utrwalać błędy obecne w danych treningowych w generowanych wynikach.

Modele transformacji wizji

Transformatory to kolejna architektura modeli, która może tworzyć obrazy. Tę samą architekturę widzieliśmy wcześniej w kontekście zadań przetwarzania języka naturalnego (NLP). Transformatory mogą również obsługiwać zadania związane z wizją i są nazywane transformatorami wizji (ViT). Transformatory to sieci neuronowe wykorzystujące mechanizmy uwagi do przetwarzania danych sekwencyjnych, takich jak tekst lub mowa, i mogą być używane do generowania podpowiedzi graficznych. Są one również bardzo skuteczne w przypadku określonych zadań, takich jak rozpoznawanie obrazów, i przewyższyły poprzednie wiodące architektury modeli. Architektura modelu ViT jest podobna do architektury NLP, aczkolwiek z pewnymi różnicami — ma większą liczbę warstw samouwagi i globalny mechanizm uwagi, pozwalający modelowi na jednoczesne zajmowanie się wszystkimi częściami obrazu. Transformatory obliczają, w jakim stopniu każdy token wejściowy jest powiązany z każdym innym tokenem wejściowym. Nazywa się to uwagą. Im więcej tokenów, tym więcej obliczeń uwagi jest potrzebnych. Liczba obliczeń uwagi rośnie wraz z kwadratem liczby tokenów, czyli jest kwadratowa. W przypadku obrazów podstawową jednostką analizy jest jednak piksel, a nie token. Relacje dla każdej pary pikseli w typowym obrazie są obliczeniowo niemożliwe. Zamiast tego, ViT oblicza relacje między pikselami w różnych małych sekcjach obrazu (zazwyczaj o rozmiarze 16 × 16 pikseli), co pomaga zmniejszyć koszt obliczeniowy. Te sekcje o rozmiarze 16 × 16, wraz z ich osadzeniami pozycyjnymi, są umieszczone w sekwencji liniowej i stanowią dane wejściowe dla transformatora. Jak pokazano na rysunku 6, model ViT składa się z trzech głównych sekcji: lewej, środkowej i prawej.



Lewa sekcja przedstawia klasy wejściowe, takie jak Klasa, Ptak, Piłka, Samochód itd. Są to możliwe etykiety, które model może przypisać do obrazu. Środkowa sekcja przedstawia liniową projekcję spłaszczonych fragmentów, które przekształcają obraz wejściowy w sekwencję wektorów, które można przekazać do enkodera transformatora. Ostatnia sekcja to enkoder transformatora. Składa się ona z kilku wielogłowicowych warstw uwagi i normalizacji i służy do poznawania relacji między różnymi częściami obrazu. Modele ViT są wykorzystywane w różnych zastosowaniach obrazów, takich jak segmentacja, klasyfikacja i detekcja, i często są dokładniejsze niż poprzednie techniki. Obsługują również dostrajanie, które można stosować w kilku próbkach z mniejszymi zbiorami danych, co czyni je bardzo przydatnymi w zastosowaniach korporacyjnych, gdzie może brakować danych. Model ViT ma na celu wygenerowanie ostatecznej reprezentacji wektorowej dla tokena klasy, który zawiera informacje o całym obrazie. Modele ViT wiążą się również z wyzwaniami, takimi jak wysokie koszty obliczeniowe, niedobór danych i kwestie etyczne. Są one złożone obliczeniowo zarówno z perspektywy uczenia, jak i wnioskowania, a także charakteryzują się niską interpretowalnością – oba te obszary są aktywnymi obszarami badawczymi. Modele multimodalne, z modelami ViT, takimi jak GPT-4, są bardzo obiecujące i otwierają nowe możliwości dla przedsiębiorstw.

Modele dyfuzyjne

Modele dyfuzyjne to generatywne modele uczenia maszynowego, które mogą tworzyć realistyczne dane z losowego szumu, takiego jak obrazy lub dźwięk. Ich celem jest poznanie ukrytej struktury zbioru danych poprzez modelowanie, w jaki sposób punkty danych dyfundują w tej ukrytej przestrzeni. Model jest trenowany poprzez powolne dodawanie szumu do obrazu i uczenie się odwracania tego procesu poprzez usuwanie szumu z wejścia, aż będzie on przypominał pożądany wynik. Na przykład, model dyfuzji może generować obraz pandy, zaczynając od losowego obrazu, a następnie powoli usuwając szum, aż będzie wyglądał jak panda. Modele dyfuzji wizji zazwyczaj składają się z dwóch części: procesu dyfuzji w przód i w tył. Proces dyfuzji w przód odpowiada za stopniowe dodawanie szumu do ukrytej reprezentacji obrazu, co powoduje uszkodzenie tej przestrzeni utajonej. Proces dyfuzji w tył działa odwrotnie — odpowiada za rekonstrukcję oryginalnego obrazu z uszkodzonej ukrytej reprezentacji. Proces dyfuzji w przód jest zazwyczaj implementowany jako łańcuch Markowa (tj. system bez pamięci o swojej przeszłości, a prawdopodobieństwo następnego kroku zależy od bieżącego stanu). Oznacza to, że uszkodzona ukryta reprezentacja w każdym kroku zależy tylko od ukrytej reprezentacji z

poprzedniego kroku, co sprawia, że proces dyfuzji w przód jest wydajny i łatwy do trenowania. Proces dyfuzji odwrotnej jest zazwyczaj implementowany jako sieć neuronowa, co oznacza, że sieć neuronowa uczy się odwracać proces dyfuzji do przodu, przewidując oryginalną, utajoną reprezentację na podstawie uszkodzonej. Ten proces dyfuzji odwrotnej jest powolny, ponieważ polega na powtarzaniu krok po kroku. Oto niektóre z zalet modeli dyfuzji:

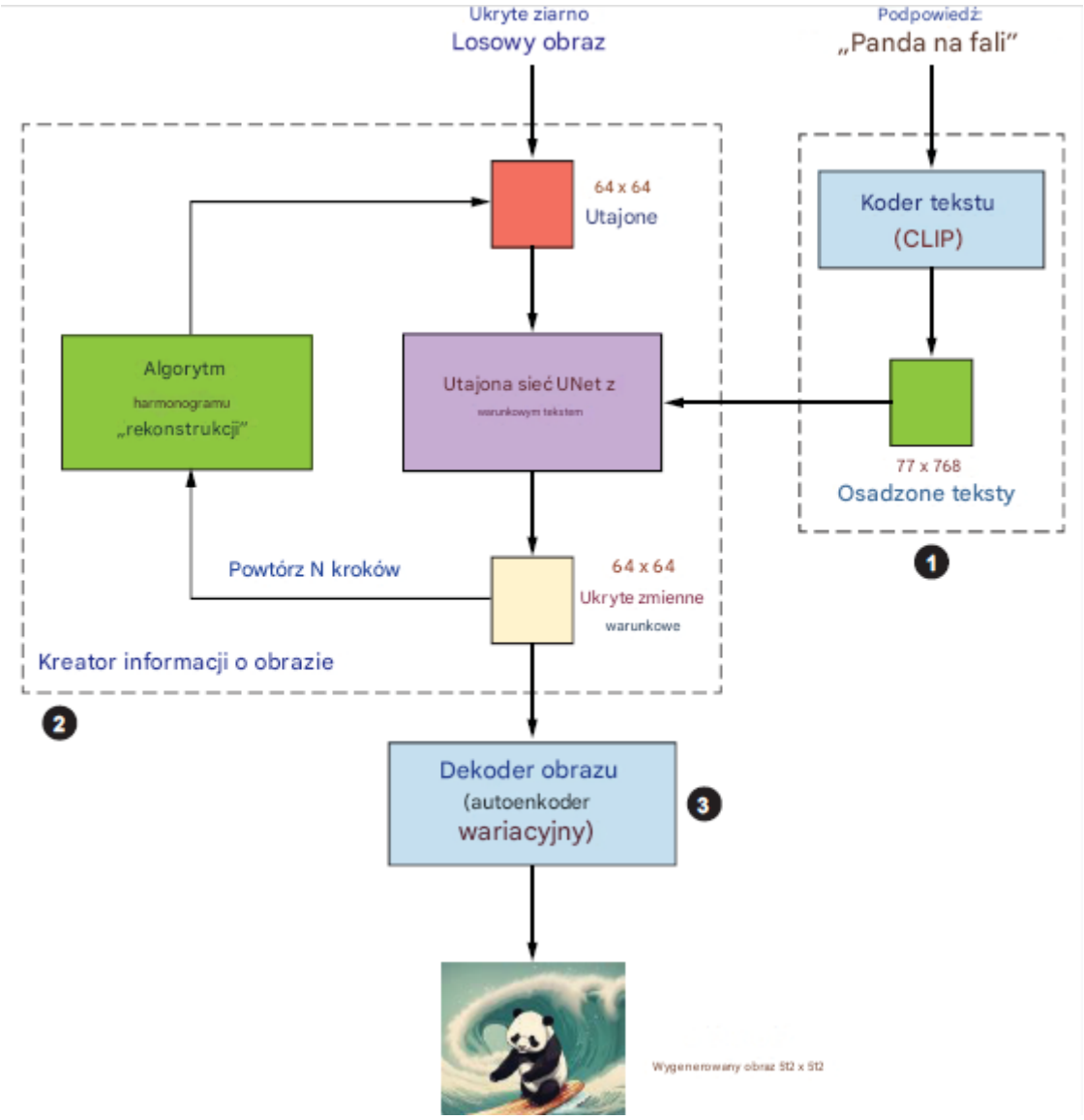
* Mogą generować wysokiej jakości obrazy, które dorównują lub przewyższają obrazy generowane przez GAN, szczególnie w przypadku złożonych scen, ale ich generowanie zajmuje znacznie więcej czasu.

* Nie występują w nich zjawiska załamania modów, które jest częstym problemem w sieciach GAN. Załamanie modów występuje, gdy generator generuje jedynie ograniczoną liczbę sygnałów wyjściowych, ignorując niektóre mody rozkładu danych.

* Modele dyfuzji mogą uchwycić pełną różnorodność rozkładu danych, wykorzystując proces łańcucha Markowa, który dodaje szum do danych wejściowych.

* Modele dyfuzji można łączyć z innymi modelami, takimi jak modele języka naturalnego, w celu tworzenia systemów generowania sterowanych tekstem.

Stabilna dyfuzja to jeden z najpopularniejszych modeli generowania obrazu opartych na dyfuzji. Jego architektura składa się z trzech głównych części (patrz rysunek 7):



* Koder tekstu, który konwertuje znak zachęty użytkownika na reprezentację wektorową.

* Autokoder odszumiający (zwany UNet), który służy do rekonstrukcji obrazu z przestrzeni latencji, oraz algorytm harmonogramu, który pomaga w rekonstrukcji obrazu oryginalnego. Nazywamy go kreatorem informacji o obrazie. UNet to autokoder odszumiający, ponieważ uczy się usuwać szum z obrazu wejściowego i generować czysty obraz wyjściowy. Jest to sieć neuronowa o strukturze koder-dekoder. Część kodująca zmniejsza rozdzielczość obrazu wejściowego i wyodrębnia jego cechy. Z drugiej strony, część dekodująca zwiększa rozdzielczość i rekonstruuje obraz wyjściowy.

* Autokoder wariacyjny (VAE), który tworzy obraz o rozkładzie możliwie najbardziej zbliżonym do rozkładu normalnego.

Wybór pomiędzy tymi modelami zależy od konkretnego zastosowania, dostępności zasobów obliczeniowych, danych szkoleniowych oraz wymagań niefunkcjonalnych, takich jak jakość obrazu, szybkość itd. Tabela 1 przedstawia niektóre z bardziej popularnych generatywnych systemów wizyjnych AI, które mogą tworzyć obrazy z tekstu.

Narzędzie AI Vision: Opis

Imagen: Imagen to opracowany przez Google model dyfuzji tekstu do obrazu, który umożliwia generowanie realistycznych obrazów z opisów tekstowych. Jest dostępny w ograniczonym podglądzie i, jak wykazano, generuje obrazy nieodróżnialne od prawdziwych fotografii.

DALL-E: OpenAI opracował model języka transformującego, który umożliwia tworzenie różnorodnych, oryginalnych, realistycznych i kreatywnych obrazów oraz dzieł sztuki na podstawie podpowiedzi. Umożliwia edycję obrazów w oparciu o kontekst, na przykład poprzez dodawanie, usuwanie lub zmianę określonych części. Wygenerował różnorodne obrazy, od przedmiotów codziennego użytku po sztukę surrealistyczną, na podstawie prostych podpowiedzi tekstowych.

DALL-E 3 to ulepszona wersja, która może generować bardziej realistyczne i dokładne obrazy o czterokrotnie większej rozdzielczości.

Midjourney: Generator sztuki oparty na sztucznej inteligencji, który wykorzystuje głębokie uczenie i sieci neuronowe do tworzenia dzieł sztuki na podstawie podpowiedzi oraz innych obrazów i filmów. Dostęp do niego można uzyskać wyłącznie za pośrednictwem serwera Discord, a wyniki można dostosować do dowolnej estetyki, od abstrakcyjnej po realistyczną, oferując w ten sposób nieograniczone możliwości twórczej ekspresji. Adobe Firefly: Adobe Firefly to rodzina kreatywnych, generatywnych modeli dyfuzji AI, zaprojektowanych z myślą o pomaganiu projektantom i profesjonalistom kreatywnym w tworzeniu obrazów i efektów tekstowych oraz ich edycji i rekolorowaniu. Jest łatwy w użyciu z innymi narzędziami Adobe, takimi jak Photoshop i Illustrator. Adobe oferuje zarówno modele tekstu do obrazu, jak i generatywne modele wypełnień.

Stabilna dyfuzja: Popularne modele obejmują wersje Stable Diffusion XL i v1.6, model generujący obraz, który wykorzystuje modele dyfuzji do tworzenia wysokiej jakości obrazów za pomocą monitów z możliwościami fotorealizmu na najwyższym poziomie. Potrafi również generować nowe obrazy z opisów tekstowych. Nowsza rodzina modeli v3 jest dostępna w rozmiarach dużym i średnim, odpowiednio z parametrami 8B i 2B.

Wiele modeli wizji AI wymienionych w tabeli 1 jest dostępnych tylko dla tych, którzy zostali zaproszeni do ich przetestowania. Jest to wciąż nowa dziedzina i większość dostawców działa powoli, ucząc się od garstki klientów przed wdrożeniem. Tworzenie i manipulowanie obrazami za pomocą generatywnej AI to ekscytujący i wymagający obszar badań o wielu potencjalnych zastosowaniach i implikacjach. Rodzi to jednak pytania etyczne i społeczne dotyczące własności, autentyczności i skutków generowanych treści. Dlatego ważne jest, aby korzystać z generatywnej AI w sposób odpowiedzialny i etyczny, uwzględniając korzyści i zagrożenia dla społeczeństwa.

Modele multimodalne

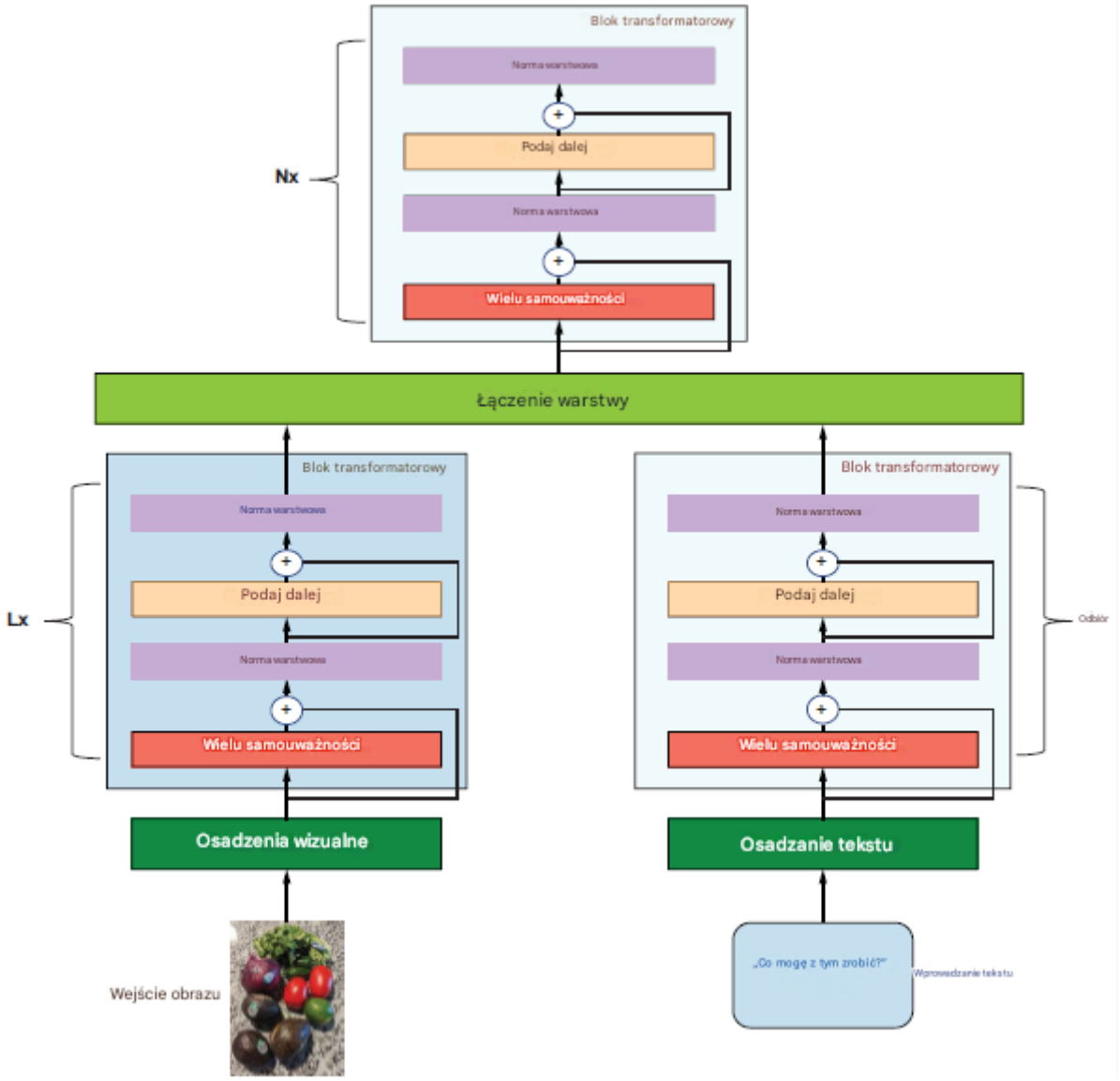
Model multimodalny może obsługiwać różne typy danych wejściowych. „Modalny” odnosi się do trybu lub rodzaju danych, a „multimodalny” odnosi się do wielu typów danych. Typy te obejmują tekst, obrazy, dźwięk, wideo i inne. Na przykład GPT-4 ma wariant modelu multimodalnego, który wykorzystuje zarówno obraz, jak i powiązany z nim komunikat, aby formułować przewidywania lub wnioskowania. Bing Chat niedawno włączył tę multimodalną funkcję, umożliwiając nam używanie obrazów i tekstu w monicie. Na przykład, jak pokazano na rysunku 8, przekazujemy modelowi dwie rzeczy: obraz i powiązany z nim monit. W tym przypadku pokazujemy produkt i pytamy model, co możemy z niego ugotować.

Co mogę z tym zrobić?

Z tych składników możesz przygotować wiele różnych dań! Oto kilka pomysłów:

- Guacamole (z awokado, pomidorami, czerwoną cebulą, kolendrą, limonką i papryczkami jalapeno)
- Salsa (z pomidorami, czerwoną cebulą, kolendrą, limonką i papryczkami jalapeno)
- Grzanki z awokado (z awokado i czerwoną cebulą)
- Sałatka z pomidorów i awokado (z awokado, pomidorami, czerwoną cebulą i kolendrą)
- Faszerowane papryczki jalapeno (z wykorzystaniem papryczek jalapeno, awokado i czerwonej cebuli)

W tym przypadku model musi zrozumieć obraz i jego poszczególne części (czyli składniki w naszym przykładzie) oraz odnieść się do polecenia, aby wygenerować odpowiedź. Odpowiedź widzimy w zacieniowanym tekście, pokazującym, że możemy przygotować guacamole, salsę, tosty z awokado i tak dalej. Modele multimodalne często wykorzystują różne techniki sztucznej inteligencji. Chociaż mogą one wykorzystywać różne kombinacje architektury modelu, w naszym przykładzie GPT-4 łączy różne bloki transformatorów (rysunek 9).



UWAGA : Podczas prezentacji bloków transformatorów, jak na rysunku 4.9, konwencja polega na użyciu Nx, odnosząc się do bloku transformatora powtarzającego się wielokrotnie; innymi słowy, jest on ułożony w stos x razy. W naszym przykładzie multimodalnym dotyczy to wszystkich trzech bloków transformatorów: obrazu po lewej stronie (Lx), tekstu po prawej stronie (Rx) i warstwy łączącej (Nx).

Modele multimodalne są szczególnie przydatne w złożonych aplikacjach rzeczywistych, gdzie dane występują w różnych formach. Na przykład:

* Internet — analizuje tekst i obrazy pod kątem moderacji treści i analizy sentymentu

* Handel elektroniczny — rekomenduje produkty, wykorzystując zarówno zdjęcia, jak i opisy tekstowe

* Opieka zdrowotna — wykorzystuje dane tekstowe (wywiad medyczny pacjenta) i obrazowanie medyczne (dane obrazowe) do diagnozy

* Jazda autonomiczna — integruje dane z czujników (radar i lidar) z danymi wizualnymi (kamery) w celu zapewnienia świadomości sytuacyjnej i podejmowania decyzji

Teraz, gdy poznaliśmy już kilka modeli, ich wyniki i ogólne zasady działania modeli sztucznej inteligencji (AI) w zakresie wizji, wygenerujmy obrazy za pomocą technologii Stable Diffusion.

Generowanie obrazu za pomocą Stable Diffusion

Stability AI, firma stojąca za Stable Diffusion, opracowała zaawansowane modele oparte na dyfuzji, a SDXL jest ich najnowszym i najpotężniejszym modelem. Oferują nam wiele opcji do wykorzystania:

* Hosting własny — model i powiązane wagi zostały opublikowane i są dostępne za pośrednictwem Hugging Face . Można je hostować samodzielnie, co wymaga odpowiedniego sprzętu obliczeniowego, w tym procesorów graficznych (GPU).

* DreamStudio — ta aplikacja konsumencka StabilityAI jest skierowana do użytkowników indywidualnych. Jest to prosty interfejs internetowy, który generuje obrazy. Firma oferuje również wersję open source o nazwie StableStudio, rozwijaną przez społeczność.

* Interfejsy API platformy — Stability AI posiada interfejs API platformy , z którego będziemy korzystać , ponieważ większość przedsiębiorstw preferuje interfejs API, którym można lepiej zarządzać w dużej skali. W naszym przykładzie wykorzystamy API REST, ponieważ oferuje ono największą elastyczność na wszystkich platformach. Stabilna wersja Diffusion posiada również API gRPC, które jest dość podobne.

Zależności

Będziemy bazować na pakietach wymaganych wcześniej i założymy, że zainstalowane są: Python, środowisko programistyczne IDE oraz środowisko wirtualne (takie jak Conda). W przypadku stabilnej wersji Diffusion potrzebujemy:

* Konta Stability AI i powiązanego klucza API; można go uzyskać na stronie konta pod adresem https://platform.stability.ai/account/keys. Dane rozliczeniowe należy również skonfigurować w tym samym miejscu. Instalujemy pakiet Python stable-sdk za pomocą pip: pip install stable-sdk.

* Zachowaj poufność klucza API i postępuj zgodnie z najlepszymi praktykami zarządzania sekretami.

Do bezpiecznego przechowywania klucza użyjemy zmiennych środowiskowych, które można skonfigurować w następujący sposób:

– Windows — setx STABILITY API KEY „twój-klucz-openai”

– Linux/Mac — export STABILITY API KEY=twój-punkt-końcowy-openai

–Bash — echo export STABILITY_API_KEY="YOUR_KEY" >> /etc/environment && source /etc/environment

Zaczynamy od pobrania listy wszystkich modeli dostępnych za pomocą API silników, w tym wszystkich dostępnych silników (tj. modeli).

Listing 1: Stabilna dyfuzja: Wyszczególnienie modeli

```
import os
import requests
import json

api_host = "https://api.stability.ai"
url = f"{api_host}/v1/engines/list"

response = requests.get(url, headers={
    "Authorization": f"Bearer {api_key}"
})

payload = response.json()

# format the payload for printing
payload = json.dumps(payload, indent=2)
print(payload)
```

REST API call for getting the models

HTTP header for authorization

Response back from the API

Making the JSON more human-readable

Wynik tego kodu przedstawiono na poniższym listingu. Pokazuje on nam, jakich silników musimy użyć, i pomaga w kompleksowym testowaniu, aby potwierdzić, że wywołanie API działa, a my możemy się uwierzytelnić i uzyskać odpowiedź.

Listing 2: Dane wyjściowe: Listy stabilnego modelu dyfuzji

[

{

"description": "Real-ESRGAN_x2plus upscaler model",

"id": "esrgan-v1-x2plus",

"name": "Real-ESRGAN x2",

"type": "PICTURE"

},

{

"description": "Stability-AI Stable Diffusion XL v1.0",

"id": "stable-diffusion-xl-1024-v1-0",

"name": "Stable Diffusion XL v1.0",

"type": "PICTURE"

},

{

"description": "Stability-AI Stable Diffusion v1.5",

"id": "stable-diffusion-v1-5",

"name": "Stable Diffusion v1.5",

"type": "PICTURE"

},

…

]

Generowanie obrazu

Do generowania obrazu używamy punktu końcowego generowania obrazu Stable Diffusion (REST API). W momencie publikacji będziemy korzystać z najnowszego modelu, modelu SDXL. Odpowiedni identyfikator silnika dla tego modelu to stable-diffusion-xl-1024-v1-0, jak pokazano w poprzednim przykładowym zestawieniu modeli. Ten identyfikator silnika jest wymagany jako część parametru ścieżki interfejsu API REST . Listing 3 przedstawia przykład użycia tego interfejsu API do generowania obrazu. Należy pamiętać, że w przykładach w tym rozdziale używamy wersji 1.0 interfejsu API. Aby korzystać z nowszych modeli, w większości przypadków wystarczy zmienić ścieżkę interfejsu API REST. Na przykład, aby wykorzystać nowsze modele, które zostały właśnie zapowiedziane, Stable Diffusion 3 i są obecnie w fazie beta.

Listing 3: Stabilna dyfuzja: Generowanie obrazu

```
import base64
import os
import requests
import datetime
import re

engine_id = "stable-diffusion-xl-1024-v1-0"
api_host = "https://api.stability.ai"
api_key = os.getenv("STABILITY_API_KEY")

prompt = "Laughing panda in the clouds eating bamboo"

# Set the folder to save the image; make sure it exists
image_dir = os.path.join(os.curdir, 'images')
if not os.path.isdir(image_dir):
    os.mkdir(image_dir)

# Function to clean up filenames
def valid_filename(s):
    s = re.sub(r'[^\w_.)( -]', '', s).strip()
    return re.sub(r'[\s]+', '_', s)

response = requests.post(
    f"{api_host}/v1/generation/{engine_id}/text-to-image",
    headers={
        "Content-Type": "application/json",
        "Accept": "application/json",
        "Authorization": f"Bearer {api_key}"
    },
    json={
        "text_prompts": [
            {
                "text": f"{prompt}",
            }
        ],


        "cfg_scale": 7,
        "height": 1024,
        "width": 1024,
        "samples": 1,
        "steps": 50,
    },
)

data = response.json()

for i, image in enumerate(data["artifacts"]):
    filename = f"sd_{valid_filename(prompt)}_{i}_{ [CR]
                datetime.datetime.now().strftime('%Y%m%d_%H%M%S')}.png"
    image_path = os.path.join(image_dir, filename)
    with open(image_path, "wb") as f:
        f.write(base64.b64decode(image["base64"]))
```

Choose the model we want to use.

Prompts used to generate the image

Helper functions to create filenames

API call for generating the image

The REST API Endpoint includes the engine ID.

Parameters controlling the model generation

Response from the API once the generation finishes

Saves the image locally

Wygenerowano obraz „śmiejącej się pandy w chmurach jedzącej bambus”, jak pokazano na rysunku 10.

Jest to całkiem wesoła i realistyczna panda. Niektóre parametry interfejsu API Stability Diffusion są podobne do tych, które już widzieliśmy. Niektóre jednak różnią się, ponieważ architektura bazowego modelu różni się od tej przedstawionej w tabeli 2.

Parametr: Typ: Wartość domyślna: Opis

Akceptuj: Ciąg: Pusty (application/json): Format odpowiedzi może być domyślny (pusty) JSON lub ustawiony na image/png dla obrazu PNG.

Organizacja: Ciąg: Null (opcjonalnie): Tag, który pozwala na ograniczenie zakresu żądań do organizacji innej niż domyślna organizacja użytkownika. Ten parametr może pomóc w debugowaniu, monitorowaniu i wykrywaniu nadużyć.

Stabilność-ID klienta: Ciąg: Null (opcjonalnie): Ten parametr służy do identyfikacji źródła żądań, takiego jak aplikacja kliencka lub podorganizacja. Może pomóc w debugowaniu, monitorowaniu i wykrywaniu nadużyć.

Stability-Client-Version: String: Null (opcjonalnie): Ten parametr identyfikuje wersję aplikacji lub usługi wysyłającej żądania. Może pomóc w debugowaniu, monitorowaniu i wykrywaniu nadużyć.

Autoryzacja: String: Nośnik API_KEY: Klucz wymagany do uwierzytelnienia wywołania API

Ponieważ korzystamy z interfejsu API REST, istnieją również dwa zestawy parametrów – jeden zestaw to parametry nagłówka, a drugi to parametry treści.

Tabela 3 przedstawia parametry, które stanowią treść wywołania API. Parametry te pozwalają na precyzyjne dostrojenie modelu i ukierunkowanie go na to, co chcemy wygenerować.

Parametr: Typ : Wartość domyślna: Opis

wysokość i szerokość: Liczba całkowita: 512 (opcjonalnie): Wysokość i szerokość obrazu w pikselach muszą być mierzone w krokach co 64 i muszą być jedną z następujących kombinacji: 1024 x 1024, 1152 x 896, 1216 x 832, 1344 x 768, 1536 x 640, 640 x 1536, 768 x 1344, 832 x 1216 i 896 x 1152. Należy pamiętać, że niektóre z nich różnią się w zależności od używanego silnika.

text_prompts: Ciąg: Null (wymagane): Do wygenerowania obrazu używana jest tablica monitów tekstowych. Każdy element tej tablicy składa się z dwóch właściwości — jednej dotyczącej samego

monitu, a drugiej odnoszącej się do niego wagi. W przypadku podpowiedzi negatywnych wagi powinny być ujemne. Na przykład:

"text_prompts": [{

"text": "Pies na macie",

"weight": 0,7

}]

Właściwość text może mieć do 2000 znaków.

cfg_scale : String: 7 (opcjonalnie): Wartość może mieścić się w zakresie od 0 do 35; definiuje ona, jak ściśle proces dyfuzji podąża za podpowiedzią. Wyższe wartości utrzymują obraz bliżej podpowiedzi.

clip_guidance _preset : String : None (opcjonalnie): Różne wartości kontrolują stopień wykorzystania podpowiedzi CLIP oraz jakość i trafność generowanego obrazu. Wartości to NONE, FAST_BLUE, FAST_GREEN, SIMPLE, SLOW, SLOWER i SLOWEST.

sampler : String : Null (opcjonalnie): Definiuje próbnik używany w procesie dyfuzji. Jeśli ta wartość zostanie pominięta, API automatycznie wybierze odpowiedni sampler. Dostępne wartości to DDIM, DDPM, K_DPMPP_2M, K_DPM_2, K_EULER K_DPMPP_2S_ANCESTRAL, K_HEUN, K_DPM_2_ANCESTRAL, K_LMS, K_EULER_ANCESTRAL.

sample: Liczba całkowita: 1 (opcjonalnie): Określa liczbę obrazów do wygenerowania. Wartości muszą mieścić się w zakresie od 1 do 10.

seed: Liczba całkowita: 0 (opcjonalnie): Losowe ziarno to liczba określająca wygląd szumu. Pozostaw 0 jako wartość losowego ziarna. Możliwy zakres wartości wynosi od 0 do 4294967295.

steps: Liczba całkowita: 50 (opcjonalnie): Określa liczbę kroków dyfuzji do uruchomienia. Zakres wartości wynosi od 10 do 150.

style_preset: Ciąg: Null (opcjonalnie): Służy do ukierunkowania modelu obrazu na określony styl predefiniowany. Wartości to: model 3D, film analogowy, anime, kinowy, komiksowy, sztuka cyfrowa, ulepszony, fantasy, izometryczny, grafika liniowa, low-poly, modelowanie złożone, neon punk, origami, fotograficzny, pikselowy i tekstura kafelkowa. Uwaga: Ta lista predefiniowanych stylów może ulec zmianie w czasie.

Przyjrzyjmy się teraz innym sposobom tworzenia obrazów.

Generowanie obrazów z innymi dostawcami

Jeśli chcemy generować obrazy, kilku innych dostawców również oferuje generatywne modele sztucznej inteligencji (AI); jednak nie mają oni platformy ani interfejsu API. W tej sekcji pokażemy inne platformy, które umożliwiają tworzenie obrazów, ale nie mają interfejsów API i w większości przypadków dostęp do nich wymaga interfejsu graficznego (GUI).

OpenAI DALLE 3

DALLE 3 to nowszy model generowania obrazów firmy OpenAI, który umożliwia tworzenie obrazów z poziomu wiersza poleceń. Był to jeden z pierwszych modeli generowania obrazów, z którym mogła współpracować większość użytkowników. DALLE to skrót od Discrete Autoencoder Language Latent Encoder, co oznacza, że wykorzystuje specjalny typ sieci neuronowej do kodowania obrazów i tekstu na tokeny, a następnie używa tych tokenów do tworzenia obrazów. DALLE może być używany zarówno

za pośrednictwem interfejsu API, jak i interfejsu graficznego (GUI). Biorąc pod uwagę, że obrazy generowane za pomocą DALLE są podobne do Stable Diffusion, nie będziemy tutaj omawiać szczegółów API. Repozytorium kodu GitHub dołączone do książki (https://bit.ly/GenAIBook) zawiera API i przykłady kodu DALLE.

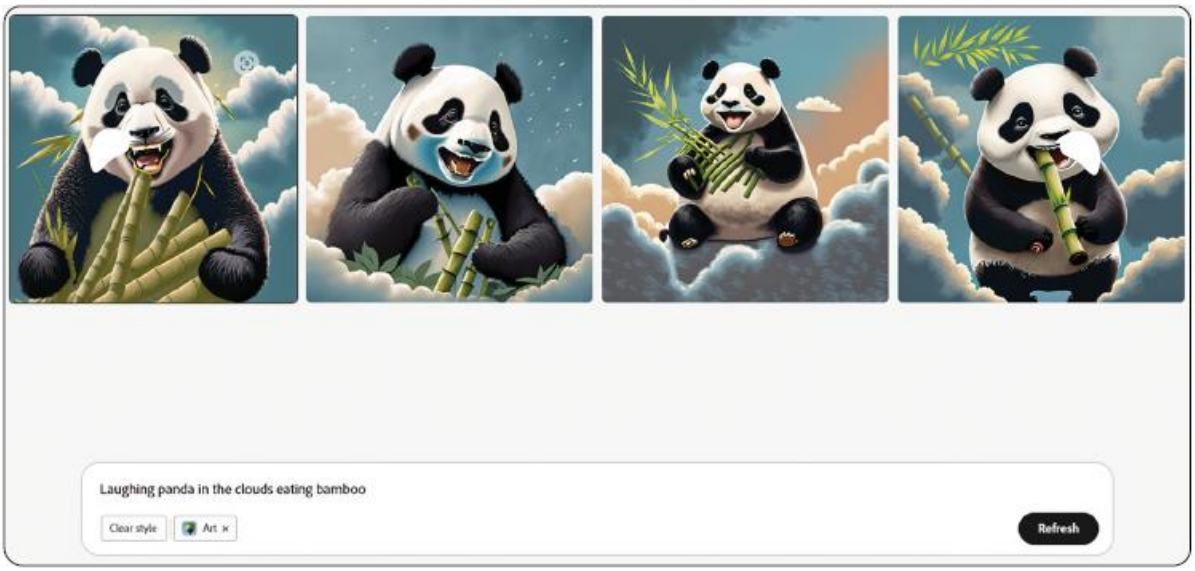
Kreator obrazów Bing

Bing oferuje aplikację do tworzenia obrazów, która wewnętrznie wykorzystuje DALLE, ale tworzone przez nią obrazy są ulepszone i nieco inne. Do korzystania z niej potrzebujemy tylko przeglądarki internetowej; API nie jest dostępne. Możemy generować obrazy, przechodząc na stronę https://www.bing.com/create i wprowadzając polecenie. Nie ma wielu modyfikacji poza tymi określonymi w samym poleceniu. Rysunek 11 przedstawia generowanie „spokojnego domku nad jeziorem, akwareli z psem". Później wykorzystamy jeden z tych obrazów, aby zobaczyć, jak edytować obraz.

Adobe Firefly

Adobe oferuje zestaw narzędzi generatywnej sztucznej inteligencji (Generative AI), a Firefly to rodzina modeli generatywnej sztucznej inteligencji (Generative AI). Jest on integrowany z różnymi produktami Adobe, takimi jak Photoshop, i jest dostępny pod adresem https://firefly.adobe.com/. Chociaż nie ma API, ogólny proces i sposób działania są takie same, jak te, które widzieliśmy wcześniej w OpenAI. Po zalogowaniu pojawia się interfejs użytkownika, w którym wprowadzamy monit i generujemy obrazy. Skorzystajmy z jednego z poprzednich przykładów:

„śmiejąca się panda w chmurach jedząca bambus". Domyślnie tworzone są cztery obrazy (rysunek 12).

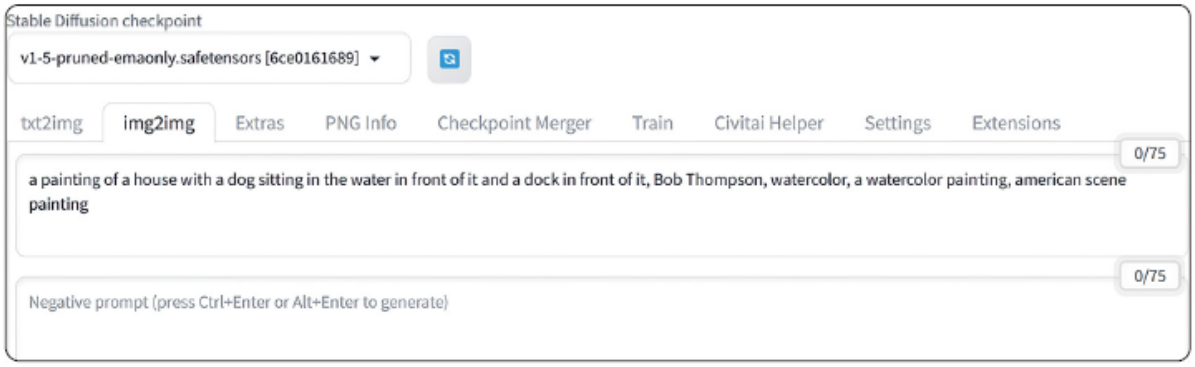


UWAGA: Google niedawno ogłosiło swój generatywny zestaw interfejsów API AI o nazwie Vertex AI; w momencie publikacji interfejsy Vision API, które również bazują na modelach dyfuzji, nie były dostępne. Skoro stworzyliśmy już obraz, zobaczmy, jak go edytować i udoskonalać.

Edycja i udoskonalanie obrazów za pomocą funkcji Stable Diffusion

Oprócz generowania obrazów, funkcja Stable Diffusion umożliwia ich edycję i udoskonalanie. Używamy interfejsu internetowego Stable Diffusion, jednego z interfejsów internetowych typu open source dla Stable Diffusion, aby pokazać, jak korzystać z funkcji inpaintingu i udoskonalać obrazy. Interfejs internetowy jest opakowaniem modelu i chociaż nie wywołuje interfejsu API, ma te same właściwości. Zaczynamy od użycia jednego z obrazów akwareli, które wygenerowano wcześniej. W tym przykładzie maskujemy dwa obszary: psa i różne kolory w lewym dolnym rogu obrazu (rysunek 13).

Podczas przesyłania obrazu do Inpaint, jedną z funkcji aplikacji internetowej jest użycie modelu CLIP do analizy obrazu i odgadnięcia podpowiedzi. Chociaż znamy podpowiedź z pierwotnej generacji, jest to inny model i wskazane byłoby, aby program Stable Diffusion ją odgadł. Wyniki przedstawiono na rysunku 14.

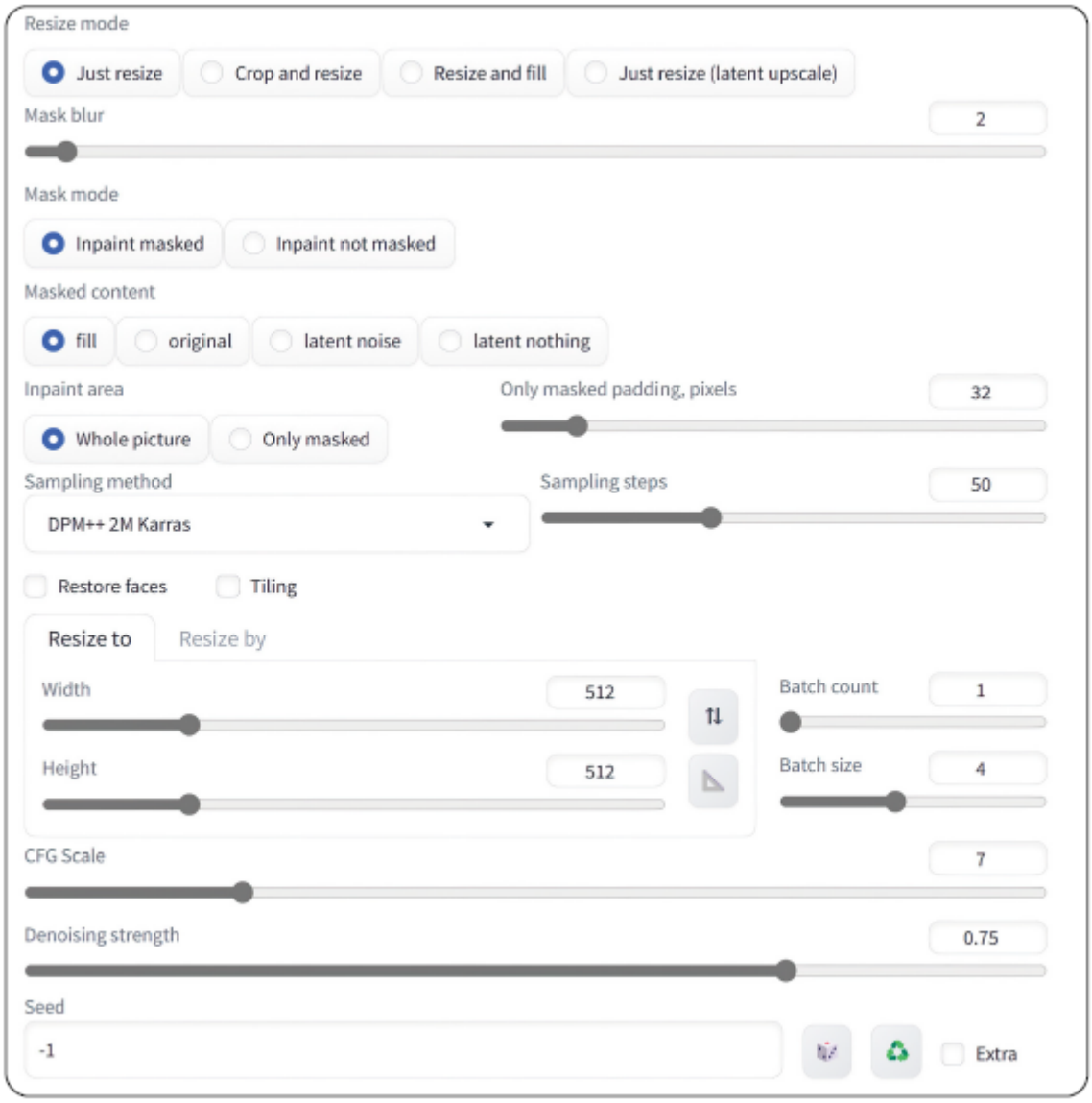


Model CLIP

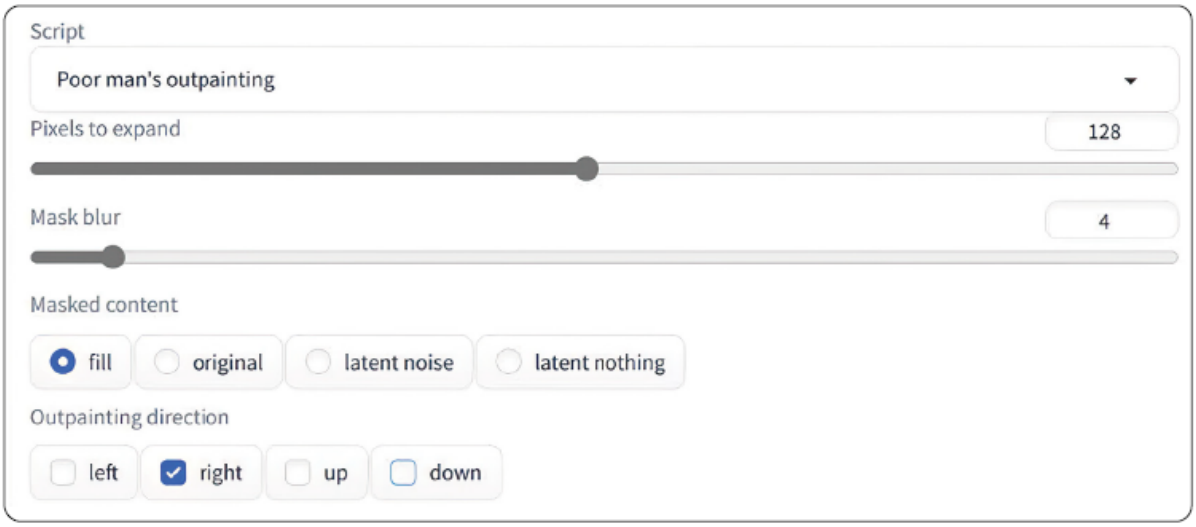
CLIP (Contrastive Language–Image Pre-training) to sieć neuronowa stworzona przez OpenAI, która łączy tekst i obrazy. Potrafi ona rozumieć i klasyfikować obrazy, dopasowując je do opisów w języku naturalnym. Odbywa się to za pomocą techniki zwanej uczeniem kontrastywnym, w której model uczy się z dużej liczby obrazów i powiązanych par tekstowych pochodzących z internetu. Unikalna zdolność

CLIP do uczenia się „bez przerw” oznacza, że może on precyzyjnie oznaczać obrazy, z którymi nigdy wcześniej się nie zetknął, na podstawie samego tekstu, bez konieczności bezpośredniego dostrajania do konkretnego zadania. Na przykład, CLIP może otrzymać nazwy klas wizualnych i zidentyfikować je na obrazach, nawet jeśli nie był specjalnie na nich trenowany. CLIP koduje zarówno tekst, jak i obrazy we wspólnej przestrzeni reprezentacji. Potrafi oszacować najbardziej odpowiedni fragment tekstu dla obrazu i odwrotnie. Daje mu to dużą elastyczność i możliwość obsługi różnych rodzajów zadań wizualnych bez konieczności korzystania z danych treningowych specyficznych dla każdego zadania.

Jak pokazano na rysunku 15, dodatkowe ustawienia dopełniania obrazu pozwalają na dokładniejszą kontrolę. Niektóre z nich są takie same jak w przypadku generowania obrazu i są równie ważne, na przykład liczba kroków próbkowania i metod.



„Outpainting” to dodatkowe ustawienie, które generuje i rozszerza obraz w wybranym kierunku. Opcję tę wybiera się z menu rozwijanego „Skrypt” na tej samej karcie ustawień (rysunek 16).



Przechodzimy przez iterację inpaintingu, usuwając żądane obszary za pomocą maski, regenerując je, a następnie dodając nowe elementy. Końcowy wynik tych iteracji pokazano na rysunku 17.

UWAGA: Szczegóły dotyczące interfejsu użytkownika Stable Diffusion, w tym konfiguracja, konfiguracja i wdrożenia, wykraczają poza zakres tej książki; jest to jednak jedna z bardzo popularnych aplikacji, które umożliwiają samodzielne hostowanie w systemach Windows, Linux i MacOS.

Generowanie za pomocą interfejsu API „obraz do obrazu”

Obraz do obrazu to potężne narzędzie do generowania lub modyfikowania nowych obrazów, które wykorzystują istniejące obrazy jako punkt wyjścia i tekst zachęty. Za pomocą tego interfejsu API możemy wygenerować nowy obraz, zmieniając jednocześnie styl i nastrój oraz dodając lub usuwając aspekty. Wykorzystajmy nasz wcześniejszy przykład spokojnego jeziora, a następnie użyjmy interfejsu API obraz-obraz, aby wygenerować nowy obraz. Bazujemy na obu przykładach, które widzieliśmy wcześniej – używamy spokojnego jeziora jako danych wejściowych i prosimy model o wygenerowanie „szczęśliwej pandy jedzącej bambus na niebie”.

Listing 4: Generowanie obrazu do obrazu

```
import base64
import os
import requests
import datetime
import re
engine_id = "stable-diffusion-xl-1024-v1-0"
api_host = "https://api.stability.ai"
api_key = os.getenv("STABILITY_API_KEY")
orginal_image = "images/serene_vacation_lake_house.jpg"
#helper functions
...
response = requests.post(
f"{api_host}/v1/generation/{engine_id}/image-to-image",
headers={
"Accept": "application/json",
"Authorization": f"Bearer {api_key}"
},
files={
"init_image": open(orginal_image, "rb")
},
data={
"image_strength": 0.35,
"init_image_mode": "IMAGE_STRENGTH",
"text_prompts[0][text]": "A happy panda eating bamboo in the sky",
"cfg_scale": 7,
"samples": 1,
"steps": 50,
"sampler": "K_DPMPP_2M"
}
)
```

```
data = response.json()

for i, image in enumerate(data["artifacts"]):

filename = f"{valid_filename(os.path.basename(orginal_image))}_

➥img2img_{i}_{datetime.datetime.now().

➥strftime('%Y%m%d_%H%M%S')}.png"

image_path = os.path.join(image_dir, filename)

with open(image_path, "wb") as f:

f.write(base64.b64decode(image["base64"]))
```

Widzimy wygenerowany obraz, jak pokazano po lewej stronie na rysunku 18 wywołania API „obraz-do-obrazu”; widzimy pandę i bambus oraz sposób użycia obrazu wejściowego do ustawienia sceny, a także typ i estetykę wygenerowanego obrazu. Nie jest on jednak zgodny z aspektem chmury w monicie. Możemy zmodyfikować parametry, aby bardziej odpowiadał monitowi, a mniej obrazowi wejściowemu, jak pokazano po prawej stronie rysunku 18. Przykładem jest panda na niebie jedząca bambus; ogólnie estetyka obrazu jest zgodna z obrazem wejściowym.

Korzystanie z API maskowania

Stable Diffusion posiada również API maskowania, które pozwala programowo edytować fragmenty obrazu. API jest bardzo podobne do API tworzenia, jak pokazano w przykładzie na listingu 4.5. Ma jednak kilka ograniczeń: obraz maski musi mieć takie same wymiary jak obraz oryginalny i być plikiem PNG o rozmiarze mniejszym niż 4 MB. API ma te same parametry nagłówka, które omówiliśmy wcześniej w rozdziale, gdy omawialiśmy generowanie obrazu; unikniemy powielania.

Listing 5: Przykład API stabilnego maskowania dyfuzyjnego

```
import base64
import os
import requests
import datetime
import re

engine_id = "stable-inpainting-512-v2-0"
api_host = "https://api.stability.ai"
api_key = os.getenv("STABILITY_API_KEY")

orginal_image = "images/serene_vacation_lake_house.jpg"
mask_image = "images/mask_serene_vacation_lake_house.jpg"
prompt = " boat with a person fishing and a dog in the boat"
```

Selects the inpainting model we want to use

Image we want to edit

Masks that we want to apply

```
# helper functions
...

response = requests.post(
    f"{api_host}/v1/generation/{engine_id}/image-to-image/masking",
    headers={
        "Accept": 'application/json',
        "Authorization": f"Bearer {api_key}"
    },
    files={
        'init_image': open(orginal_image, 'rb'),
        'mask_image': open(mask_image, 'rb'),
    },
    data={
        "mask_source": "MASK_IMAGE_BLACK",
        "text_prompts[0][text]": prompt,
        "cfg_scale": 7,
        "clip_guidance_preset": "FAST_BLUE",
        "samples": 4,
        "steps": 50,
    }
)

data = response.json()

for i, image in enumerate(data["artifacts"]):
    filename = f"{valid_filename(os.path.basename(orginal_image))}_
                masking_{i}_{datetime.datetime.now().
                strftime('%Y%m%d_%H%M%S')}.png"
    image_path = os.path.join(image_dir, filename)
    with open(image_path, "wb") as f:
        f.write(base64.b64decode(image["base64"]))
```

Masks API call

Selects the black pixels of the image to be replaced

Prompts for the generation

Specifies the number of images to generate

Determines the number of steps for each of the images

Gets the response from the API

Saves the edited image to disk

Tabela 4 przedstawia wszystkie parametry API. Jeśli chodzi o opcje sterowania modelem, wiele z nich jest podobnych do tych z poprzedniego tworzenia obrazu.

Parametr: Typ: Wartość domyślna: Opis

init_image: Ciąg: Binarny (wymagany): Początkowy obraz, który chcemy edytować

mask_source: Ciąg: Null (wymagany): Szczegóły maski określające obszary generacji i powiązane z nimi siły. Może to być jedna z następujących wartości:

MASK_IMAGE_WHITE — użyj białych pikseli jako maski; białe piksele zostaną zmodyfikowane; czarne piksele pozostaną niezmienione.

MASK_IMAGE_BLACK — użyj czarnych pikseli jako maski; czarne piksele zostaną zmodyfikowane; białe piksele pozostaną niezmienione

INIT_IMAGE_ALPHA — użyj kanału alfa jako maski. Edytuj w pełni przezroczyste piksele i pozostaw w pełni nieprzezroczyste piksele bez zmian.

mask_image: String: Binarny (wymagany): Obraz maski, który wskazuje model, w którym piksele wymagają modyfikacji. Ten parametr jest używany tylko wtedy, gdy mask_source to MASK_IMAGE_BLACK lub MASK_IMAGE_WHITE.

text_prompts: String: Null (wymagany): Do wygenerowania obrazu używana jest tablica monitów tekstowych. Każdy element tej tablicy zawiera dwie właściwości — jedną dotyczącą samego monitu i drugą dotyczącą powiązanej wagi. Wagi powinny być ujemne w przypadku monitów o wartości ujemnej. Monity muszą być zgodne z następującym formatem:

text_prompts[index][text|weight], przy czym indeks musi być unikalny i nie musi być sekwencyjny.

cfg_scale: String: 7 (opcjonalnie): Może zawierać się w zakresie od 0 do 35; definiuje, jak ściśle proces dyfuzji podąża za monitami. Wyższe wartości utrzymują obraz bliżej monitu.

clip_guidance _preset : String :None (opcjonalny): Różne wartości kontrolują, jak często używane są wskazówki CLIP i wpływają na jakość i trafność generowanego obrazu. Możliwe wartości to NONE, FAST_BLUE, FAST_GREEN, SIMPLE, SLOW, SLOWER i SLOWEST.

sampler : String : Null (opcjonalny): Definiuje sampler używany w procesie dyfuzji. Jeśli ta wartość zostanie pominięta, API automatycznie wybierze odpowiedni sampler. Możliwe wartości to DDIM, DDPM, K_DPMPP_2M, K_DPM_2, K_EULER K_DPMPP_2S_ANCESTRAL, K_HEUN, K_DPM_2_ANCESTRAL, K_LMS i K_EULER_ANCESTRAL.

samples : Integer :1 (opcjonalny): Definiuje liczbę generowanych obrazów. Wartości muszą mieścić się w zakresie od 1 do 10.

seed: Liczba całkowita: 0 (opcjonalnie): Losowe ziarno to liczba określająca wygląd szumu. Pozostaw 0 jako wartość losowego ziarna. Możliwe wartości mieszczą się w zakresie od 0 do 4294967295.

steps: Liczba całkowita: 50 (opcjonalnie): Określa liczbę kroków dyfuzji do uruchomienia. Możliwe wartości mieszczą się w zakresie od 10 do 150.

style_preset: Ciąg: Null (opcjonalnie): Służy do kierowania modelem obrazu w kierunku określonego stylu predefiniowanego. Możliwe wartości to: model 3d,

film analogowy, anime, kinowy, komiksowy, sztuka cyfrowa, ulepszony, sztuka fantasy, izometryczny, grafika liniowa, low-poly, modelingcompound, neon-punk, origami, fotograficzny, sztuka pikselowa i tekstura kafelkowa.

Uwaga: Ta lista stylów predefiniowanych może ulec zmianie w czasie.

Zmiana rozmiaru za pomocą API upscale

Ostatnie API Stable Diffusion, które omówimy, służy do skalowania obrazu w górę, czyli generowania obrazu o wyższej rozdzielczości. Domyślnie obraz wejściowy jest skalowany dwukrotnie, z maksymalną liczbą pikseli 4 194 304, co odpowiada maksymalnym wymiarom 2048 × 2048 i 4096 × 1024. API jest

proste, jak pokazano na poniższym listingu. Najważniejsze, o czym należy pamiętać, to użycie odpowiedniego modelu za pomocą parametru engine_id.

Listing 6: Stabilny interfejs API zmiany rozmiaru dyfuzji

```
import base64

import os

import requests

import datetime

import re

engine_id = "esrgan-v1-x2plus"

api_host = "https://api.stability.ai"

api_key = os.getenv("STABILITY_API_KEY")

orginal_image = "images/serene_vacation_lake_house.jpg"

# helper functions

...

response = requests.post(

f"{api_host}/v1/generation/{engine_id}/image-to-image/upscale",

headers={

"Accept": "image/png",

"Authorization": f"Bearer {api_key}"

},

files={

"image": open(orginal_image, "rb")

},

data={

"width": 2048,

}

)

filename = f"{valid_filename(os.path.basename(orginal_image))}_

➥upscale_{datetime.datetime.now().

➥strftime('%Y%m%d_%H%M%S')}.png"

image_path = os.path.join(image_dir, filename)
```

```
with open(image_path, "wb") as f:

f.write(response.content)
```

Teraz, gdy omówiliśmy już liczne opcje generowania obrazów z wykorzystaniem interfejsów graficznych i interfejsów API, przyjrzyjmy się najlepszym praktykom dla przedsiębiorstw.

Znak wodny dla obrazów generowanych przez sztuczną inteligencję

Ponieważ obrazy generowane przez sztuczną inteligencję są coraz lepsze i często nie potrafimy odróżnić obrazów rzeczywistych od generowanych przez sztuczną inteligencję, coraz popularniejsze staje się znakowanie wodne obrazów generowanych przez sztuczną inteligencję. Obecnie istnieją dwa główne sposoby: widoczne znaki wodne, takie jak te stosowane przez Bing i DALLE, oraz niewidoczne znaki wodne, które nie są dla nas widoczne, ale są osadzone w obrazie i można je wykryć za pomocą specjalnych narzędzi. Google poszedł o krok dalej i opracował nowy typ znaku wodnego o nazwie SynthID. Niewidoczny znak wodny jest osadzany w każdym pikselu obrazu, co czyni go bardziej odpornym na manipulacje obrazem, takie jak filtry, zmiana rozmiaru i kadrowanie. Dzieje się to bez zauważalnego pogorszenia jakości obrazu i bez znaczącej zmiany jego rozmiaru. Znakowanie wodne obrazów generowanych przez sztuczną inteligencję ma wiele zalet. Oprócz wskazywania pochodzenia i ewentualnego właściciela obrazów, pomaga to zniechęcić do nieautoryzowanego użycia i dystrybucji oraz zapobiega rozprzestrzenianiu się dezinformacji.

Wskazówki dotyczące generowania obrazu

W tej sekcji przedstawiono kilka najlepszych praktyk w zakresie generowania obrazu. W kontekście przedsiębiorstw, poza niektórymi funkcjami, takimi jak graficy i artyści, wiele osób o różnych umiejętnościach potrzebuje pomocy. Poniższe sugestie pomogą im rozpocząć pracę. Więcej szczegółów omówimy w dalszej części książki, omawiając inżynierię wizualizacji:

* Opisz szczegółowo — opisz szczegółowo główny temat, który chcesz wygenerować. Elementy wizualne, które sobie wyobrażamy lub chcemy, mogą nie odpowiadać interpretacji modelu, dlatego dodanie szczegółów i wskazówek może nakierować model na to, czego oczekujesz. Wiele osób zapomina również o opisaniu tła; ważne jest, aby dodać te szczegóły.

* Klimat i styl artystyczny — określ styl klimatu lub sztuki, który jest Twoim zamierzeniem; na przykład, w poprzednich wskazówkach nakreśliliśmy zarys obrazu. Lista jest nieskończona i w pewnym sensie zależy od Twojej wyobraźni, od malarstwa olejnego, przez steampunk, po fotografię akcji.

* Określ emocje, energię i nastrój — dodaj przymiotniki i czasowniki, które oddają nastrój, energię i ogólne emocje — na przykład, wygenerowany obraz ma być pozytywny i wysokoenergetyczny, pozytywny, ale niskoenergetyczny itd.

* Generowanie dłoni i twarzy — stanowią one problem dla wielu modeli i chociaż stają się coraz lepsze, czasami lepiej jest dodać do wygenerowanych obrazów obrazy stockowe lub inne.

* Struktura, rozmiar, światło i perspektywy oglądania — myśląc o klimacie i stylu obrazu docelowego, należy również wziąć pod uwagę rozmiar i strukturę artefaktów. Na przykład, czy oczekujemy czegoś małego i skomplikowanego, czy dużego i wolnostojącego? I z jakiej perspektywy oglądane są artefakty — czy jest to zbliżenie, ujęcie z daleka, szeroki kąt, plener, czy światło naturalne? Oczywiście, biorąc pod uwagę, że mówimy o podpowiedzi, może ona łączyć wiele z tych rzeczy.

* Słowa, logo i znaki — modele obrazów nie są dużymi modelami językowymi i zazwyczaj mają problemy z obrazami, w których oczekujemy generowania słów (np. salon dla zwierząt z nazwą na

zewnątrz). Najlepiej dodawać je ręcznie podczas edycji obrazów. Po dodaniu możemy skorzystać z funkcji „inpaint”.

* Unikaj jednoczesnego stosowania wielu znaków — dodawanie wielu znaków w tym samym poleceniu i zadaniu generowania często prowadzi do pomyłki w modelu. Lepiej zacząć od mniejszych zadań, a następnie skorzystać z funkcji „inpaint” lub ręcznie edytować te elementy.

Podsumowanie

* Generatywne modele AI oparte na wizji pozwalają nam tworzyć unikalne i realistyczne treści, a wszystko to za pomocą prostego polecenia. Modele te mogą generować nowe treści, edytować i ulepszać istniejące obrazy oraz korzystać z prostych poleceń.

* Generatywne modele AI do wizji mają wiele zastosowań, w których można je wykorzystać do tworzenia treści kreatywnych, edycji obrazów, tworzenia danych syntetycznych i projektowania generatywnego.

* Istnieją cztery główne architektury generatywnych modeli AI, z których każda ma swoje mocne i słabe strony. Wyjaśniliśmy autoenkodery wariacyjne (VAE), generatywne sieci przeciwstawne (GAN), modele transformacji wizji (ViT) i modele dyfuzyjne.

* Modele multimodalne to różne generatywne modele AI, które pozwalają nam jednocześnie obsługiwać różne typy danych wejściowych, w tym tekst, obrazy, dźwięk i wideo.

* DALLE firmy OpenAI, Bing, Adobe i Stable Diffusion firmy Stability AI to jedne z bardziej znanych i powszechnych generatywnych modeli AI do generowania i edycji obrazów wykorzystywanych przez przedsiębiorstwa. Większość elementów udostępnianych za pośrednictwem API posiada również odpowiednie interfejsy GUI.

* Wiele generatywnych modeli wizji AI obsługuje inpainting (modyfikowanie fragmentów obrazu), outpainting (rozszerzanie obrazu poza jego pierwotne granice) oraz tworzenie wariacji obrazu.

* Modele dyfuzyjne są bardziej niezawodne w modelowaniu zapadania się obrazu i obsłudze różnych wyników.

* Wreszcie, jeśli chodzi o obrazy, musimy wziąć pod uwagę scenę, głównego bohatera, strukturę i elementy takie jak tekst i twarze, które lepiej jest tworzyć ręcznie i edytować w obrazie. Te aspekty muszą zostać dodane do monitu w celu generowania. W dalszej części książki omówimy ten temat w ramach inżynierii monitu.