

画像生成 AI を用いた建築デザイン支援システムの開発

古庄 玄樹*¹・井坂 匠吾*²

Keywords : image generation AI, architectural perspective drawing, ControlNet, diffusion model

画像生成 AI, 建築パース, コントロールネット, 拡散モデル

1. はじめに

1.1 研究の背景

従来の建築設計の初期段階では、計画の方向性を決定すべく設計のコンセプトの立案、敷地条件の分析、空間構成、建築ボリュームのスタディといった手法を通じて、計画の方向性を段階的に具体化していくアプローチが一般的に採用されてきた。その各フェーズではスケッチやスタディモデル等、抽象的なイメージを共有しながらデザインの方向性を固めていく手法が多く取られている。しかし、これらの表現は抽象性が高く、解釈が設計者個人の主観的な感覚や経験に依存しやすいという課題がある。具体かつ詳細に書かれたスケッチなどが共有されれば、デザインに関する合意形成が容易となるが、デザインの方向性が定まっていないう段階で詳細なスケッチを作成することは時間的、労力的制約から困難である。

近年、Diffusion Model(拡散モデル)を用いた画像生成手法¹⁾が急速に発展しており、建築設計分野においてもデザイン創出の手段として使用されるケースが見られる²⁾。拡散モデルは大量に画像を学習することで、与えられたテキストプロンプト（生成したい画像を表す説明文）から精細かつ写実的な画像の生成が可能である。一方で、拡散モデルによる画像生成は主にプロンプト入力のみによって画像を生成するため、生成画像内の建築の概形を設計者の意図通りに修正するといった操作ができないという制約がある。

ControlNet はこうした課題を克服するため提案された手法³⁾で、ユーザーが入力した線画や深度マップのガイドとなる画像をもとに、拡散モデルの画像生成プ

ロセスを制御する手法で、テキストプロンプトのみでは困難であった設計者の意図通りに概形で建築パースを生成することが可能となる。すでに BIM ソフトや 3D モデリングソフトと ControlNet を連携する取組などが行われており⁴⁾、拡散モデルによる画像生成技術は建築設計プロセスへの導入が急速に進展している。

しかしながら、ControlNet と連携する各種ソフトウェアの利用においては、ユーザーが当該ソフトウェアの操作方法に習熟していない場合、ソフトウェア自体への習熟が必要となるだけでなく、画像生成プロセスに関する専門的な知識も求められるため、総じて高い学習コストが課題となる。そのため、実務に利用できる設計者は限定的である。

Canny アルゴリズム⁴⁾や Depth Anything⁵⁾などカメラ画像を自動的に線画や深度マップに変換する画像処理手法は既に確立されている。既存の ControlNet と連携する各種ソフトウェアにおいてはソフトウェアへの習熟が必要であるが、カメラヘスケッチや写真、模型等の画像を映し自動画像処理によって線画や深度マップに変換した画像を ControlNet に入力するシステムであれば、学習コストはかからない。画像生成プロセスの学習コストについては、GPTs⁶⁾など大規模言語モデルの分野のアプリケーションにみられる、システムプロンプトをあらかじめ設定することで簡易なプロンプトのみで高精度な出力を得る手法⁶⁾の応用が期待できる。画像生成においては特定のワードが生成精度の向上に寄与することが明らかになっており⁷⁾、これらのキーワードをシステムプロンプトとして組み込むことで、ユーザーは簡易なプロンプトのみで高精細な画像の作成が可能となると考えられる。

* 1 技術センター 都市基盤技術研究部 AI研究室

* 2 設計本部 先端デザイン部

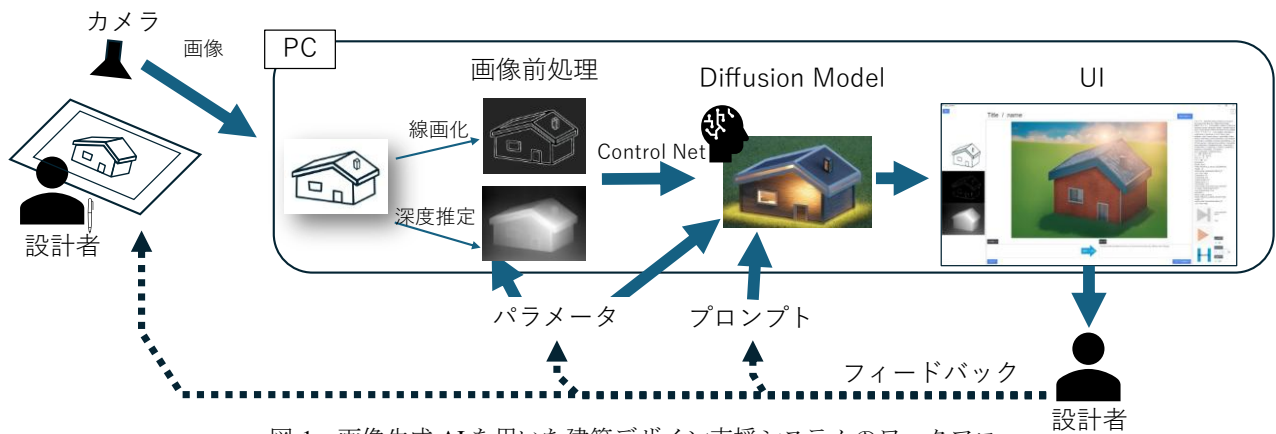


図-1 画像生成 AI を用いた建築デザイン支援システムのワークフロー
Fig.1 Workflow of the architectural design support system with image generation AI

1.2 研究の概要

このような背景から、カメラから参照する画像を入力し、画像生成の精度を高めるプロンプトが自動入力される画像生成システムを構築することで、特別なモデリングソフトや画像生成に習熟しなくとも利用可能な建築パース画像生成システムを開発する。本報では開発したシステムの概要とシステムを用いた建築パース画像生成の検証結果について報告する。検証ではカメラに映す対象はスケッチ、画像、模型とし、それぞれを入力として画像生成した結果を示し、開発したシステムの有用性について議論する。

2. システム

2.1 ワークフロー

開発したシステムを用いて画像生成を行うワークフローを図-1 に示す。カメラからの入力としてはスケッチ、画像、模型が対象となるが(図-2)、説明ではスケッチを取り上げ紹介する。まず、設計者は紙にスケッチを行う。紙の上方にはカメラを設置し、これを通じて PC にスケッチの画像が入力される。入力された画像には画像前処理を実施し、線画もしくは深度マップに変換する。画像前処理した画像は ControlNet に入力し、プロンプトと併せて拡散モデルによる画像生成を行う。本システムでは拡散モデルに stable-diffusion-xl-base-1.0^{注3)}モデルを使用した。生成された画像は UI 上に表示され閲覧可能となる。UI で結果を閲覧した設計者は、フィードバックとしてプロンプトや各種パラメータの調整、スケッチの変更などを実施し再度画像を生成する。本システムではこのワークフローを繰り返すことによって、設計者は意図通りの建築パース画像を生成できる。設計の初期検討への利用では、多様な建築パースを生成し、生成結果から評価の高いものを

デザインの参考とする、といった使用方法を想定している。

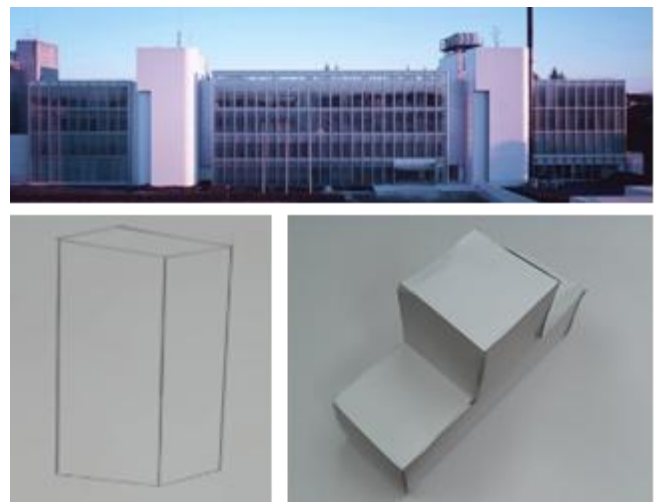


図-2 入力画像の例 (左下) スケッチ, (上) 写真, (右下) 模型
Fig.2 Examples of input images (lower left) Sketch, (upper) Photo, (lower right) Architectural model

2.2 画像前処理

ControlNet は線画や深度マップなどを画像生成のガイドとする手法であり、線画の入力であれば生成される画像の輪郭が線画に沿うように、深度マップの入力であれば、生成される画像に描画されるものの形状が深度マップに沿うように画像が生成される。カメラ画像を線画化するアルゴリズムには Canny アルゴリズムを採用し(図-3)、深度マップ化する手法は Depth Anything を採用した(図-4)。線画や深度マップのほかにも ControlNet にガイドとして入力可能なデータ形式は存在するが、ControlNet への入力として代表的なガイド且つスケッチや模型などカメラ画像から自動的に生成可能なものとして、線画と深度マップを画像前処理として選定した。

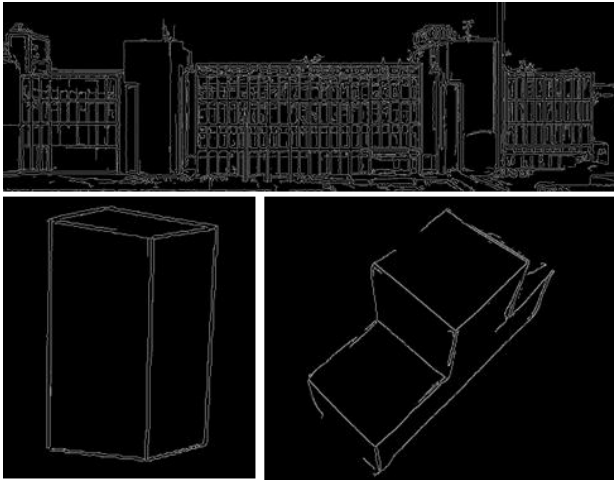


図-3 Canny アルゴリズムによる線画化の結果
Fig-3 Results of line drawing with Canny Algorithm.

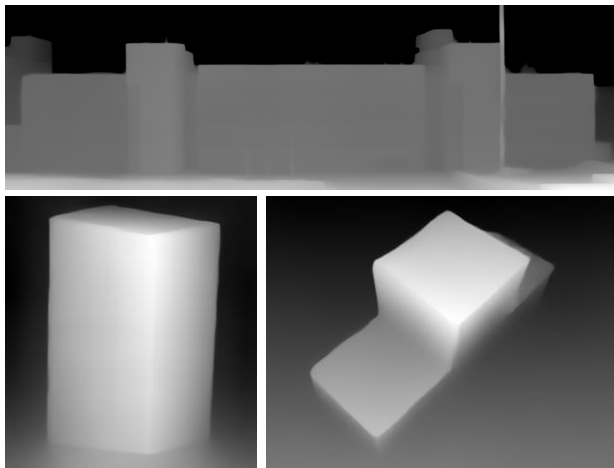


図-4 Depth Anything による深度推定結果
Fig.4 Results of depth map with Depth Anything

2.3 画像生成

画像出力の結果はプロンプトによって大きく結果が変わるが、結果に寄与するプロンプトは拡散モデルの学習データに依拠しているため、直観的な絵を修飾する語彙が必ずしも画像生成の結果に反映されず、習熟していないものには扱いが難しい。本システムでは大まかな生成対象のイメージのみをユーザー入力とし、『建築パースを生成する』、『生成イメージを高精細にする』といった内容はシステムプロンプトに自動で追加される仕様とした。システムプロンプト無しで画像を生成した結果とシステムプロンプトありで画像を生成した結果を図-5 に示す。システムプロンプトなしの生成では直方体のような形状を確認できるが建物として成立しているとは言い難く、背景も現実的な空間を表現していない。システムプロンプトありで生成した画像では中央に建物が描かれ、建築パースとして成立した画像が生成されていることが確認できる。

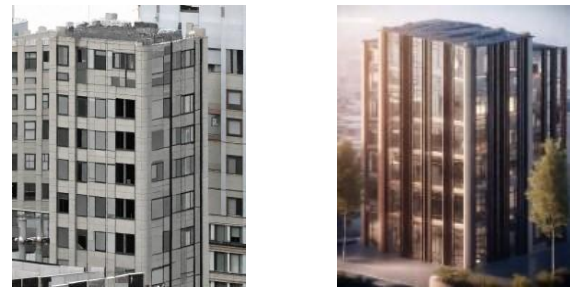


図-5 プロンプトを『Building』とした画像生成の結果
(左) システムプロンプト無し, (右) システムプロンプトあり
Fig.5 Results of image generation with the prompt "Building".
(left) without system prompt, (right) with system prompt

2.4 UI

2.4.1 カメラ範囲選択画面

UIはカメラ範囲選択画面、画像生成画面、画像生成結果一覧画面の三画面作成した。カメラ範囲選択画面ではカメラの可視範囲から画像生成ワークフローに入力する範囲を設定する。図-6 にはカメラの可視範囲全体が映っている。可視範囲全体を入力するとスケッチ以外の映り込みが発生し、意図しない画像生成結果を引き起こす。本システムではカメラ範囲選択画面上で入力したい範囲の左上と右下の二点（図-6 内、赤点）を選択することで入力範囲（図-6 内、水色の枠内）を設定する UI を開発した。

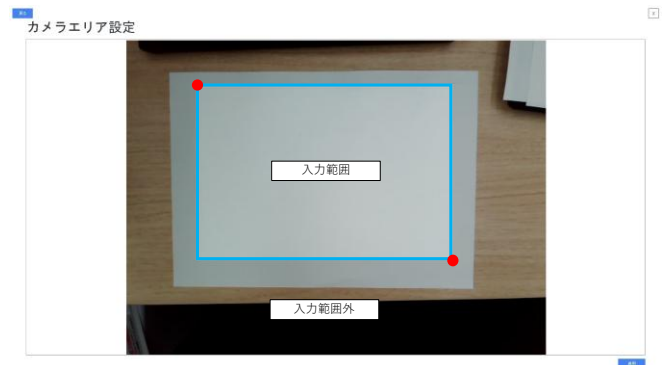


図-6 カメラ範囲選択画面 (紙上の明るい範囲が画像前処理に入力される範囲)
Fig.6 Camera range selection UI (bright area on paper is input into image preprocessing)

2.4.2 画像生成画面

画像生成画面を図-7 に示す。画面左にはカメラ画像と画像前処理結果、画面中央には画像生成結果、画面中央下には日本語プロンプト入力フィールドと英語プロンプト入力フィールド、画面右には生成時のパラメータの表示と画像生成の開始ボタンを設置した。使用した画像生成モデルでは英語以外のプロンプトでは画

像生成精度が低くなるため、日本語プロンプト入力フィールドに書かれたプロンプトは機械翻訳し、英語のフィールドに追加される仕様とした。画像生成の開始ボタンを押下すると、ユーザーが入力したプロンプトにシステムプロンプトが追加されて画像生成が実行される。これによりユーザーは簡易に高精細な画像生成が可能となる。

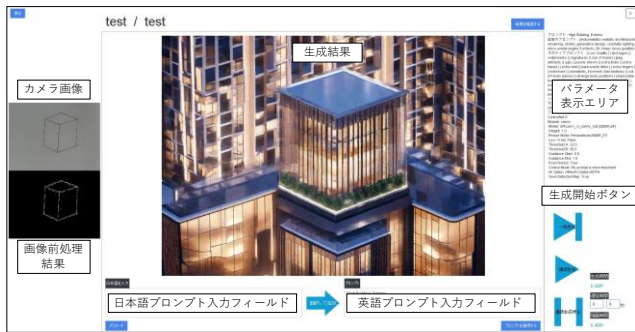


図-7 画像生成画面
Fig.7 Image generation UI

2.4.3 画像生成結果の一覧画面

画像生成の結果は画像生成結果一覧画面で確認できる(図-8)。一覧から画像を選択すると選択した画像の拡大表示と生成時のカメラ画像と実行時のプロンプトとパラメータが表示される(図-9)。

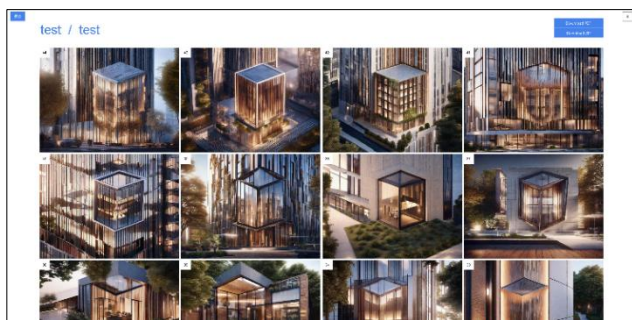


図-8 画像生成結果の一覧画面
Fig.8 UI for displaying a list of image generation results



図-9 個別の生成結果と生成時の入力画像とパラメータの表示画面
Fig.9 Screen showing individual generation results, input images, and parameters

3. 画像生成検証

3.1 検証手法

前章に示したシステムを用いて、図-2 に示したスケッチ、画像、模型のそれぞれを入力とした画像生成実験を行う。その結果から本システムの実用性について検証する。検証では画像前処理による生成結果の違い、入力したプロンプトによる生成結果の違いについてそれぞれ検証を行う。図-1 のワークフローではパラメータのフィードバックについても示しているが、画像前処理のパラメータはシステムの稼働場所の光環境などの外的要因に対応するためのアルゴリズムの閾値の調整が主であり、画像生成時のパラメータは画像生成の結果を固定化するためのシード値の固定が主な目的であるため検証は値の違いによる比較は行わない。

3.2 画像前処理による生成結果の違い

画像前処理を Canny アルゴリズム、Depth Anything で実施した結果を図-10、11 に示した。画像生成時のパラメータは同一とし、プロンプトを『Building』とした。図-10、11 の画像はいずれも入力である図-2 の画像で示す概形が反映されていることが確認できる。

模型を入力した画像生成結果(図-10 右下と図-11 右下)を比較すると画像前処理が Canny アルゴリズムである図-10 右下の方が建物の中央部分が出っ張る形状になるなど建物の形状が不正確である。要因としては線画による入力では二次元の情報から解釈するため、想起される形状に別の解釈余地が発生していることが考えられる。このことから模型の形状を正しく反映させるためには Depth Anything を採用が適していると考えられる。

スケッチ入力に対する Depth Anything による深度推定結果はスケッチの精度に左右された(図-12)。Depth Anything による深度マップの精度が低い場合、画生成される建築パースの精度も低くなることが確認された。一方でスケッチ入力に対する Canny アルゴリズムでは線で描かれたものを線画化しているため画像前処理結果で想定外の結果になることは少ない。これらの結果からスケッチについては Canny アルゴリズムを、模型については Depth Anything を前処理に使用することが適していると考えられる。

画像入力においては Canny アルゴリズム、Depth Anything のどちらの画像前処理結果を用いても画像生成の結果に形状を反映させることができおり、どちらも適していると考えられる。



図-10 画像前処理に Canny アルゴリズムを採用した
画像生成結果

Fig.10 Results of image generation with Canny Algorithm



図-11 画像前処理に Depth Anything を採用した画像生成結果

Fig.11 Results of image generation with Depth Anything



図-12 Depth Anything の推定精度が低く画像生成結果に形状
の反映がされなかった例, (左) スケッチ, (中央) Depth
Anything による深度マップ, (右) 画像生成結果

Fig.12 Result of image generation with low accuracy depth map
(left) sketch, (center) depth map by depth anything, (right) Result of
image generation

3.3 プロンプトによる生成結果の違い

プロンプトによる生成結果の違いについて検討するために、図-10 の結果を基準としプロンプトのみを変更し画像を生成する検証を実施した。

素材に言及する例として『Wood Facade Building』をプロンプトとして入力した結果を図-13 に示す。どの結

果もプロンプトが反映されているが、模型の画像が入力の結果（図-13）ではどこまでが建物か判別がつかない破綻した絵となった。破綻した要因には、線画の入力であるため線の入力に解釈の余地が発生し、現実的な建物の形に収束できなかったことが考えられる。木の配置が屋上部分とファサード部分で同じ向きになっていることから、全体として形状の整合性が取れないまま画像の生成が進行したと考えられる。3.2 で述べた通り、画像前処理に Depth Anything を用いることでこのような破綻は対応が可能である。

直接的な見た目に言及するプロンプトの例として『Colorful Building』を入力した結果を図-14 に示す。こちらの検証では形状の破綻なく、プロンプトが反映されることを確認した。

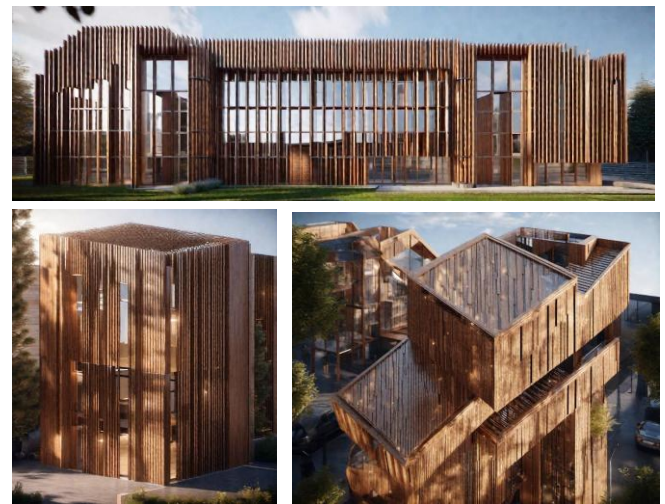


図-13 プロンプトを『Wood Facade Building』として
画像生成した結果

Fig.13 Results of image generation with prompt
“Wood Facade Building”



図-14 プロンプトを『Colorful Building』として
画像生成した結果

Fig.14 Results of image generation with prompt
“Colorful Building”

4. まとめ

画像前処理の検証によって、カメラに映す対象がスケッチであれば Canny アルゴリズム，モデルであれば Depth Anything が適切であるとの結果を得た。プロンプトの検証では素材や直接の見た目に言及したプロンプトが画像の生成結果に反映されることを確認した。

本報では、従来、特定のソフトウェアや画像生成に関する知識を必要としていた画像生成 AI を用いた設計検討プロセスに対し、特定のソフトウェアや画像生成に習熟せずとも利用可能な建築パース画像を生成するシステムの構築を試みた。開発したシステムは、システムプロンプトの導入により、簡易なプロンプトの入力のみで高精細な建物の画像の生成が可能となった。プロンプトの変更のみで簡易に多様な建築パースを生成できることから、設計者は大量に生成した建築パースの中から評価の高いものを建築デザインの参考にするなど、建築設計における初期検討への活用が期待できる。また、画像の生成に特別な知識や技能が無くとも画像を生成できる点から、設計者だけでなく、絵を描くのが不得手な発注側の人物がデザインの要望を伝える手段としても活用が期待できる。デザイン要望の伝達手段として具体の建築パースが利用できることで合意形成を確実にすることに寄与すると考えられる。

研究で開発した画像を生成するシステムは運用を開始しており、今後ユーザーの要望に応じて高機能化を予定している。現状一枚ずつ画像を生成しており、編集できないが、一度生成した画像をブラッシュアップする手段として、画像内の特定部分を範囲選択し再生成するなどの機能拡充、さらなる操作性および柔軟性の向上に向けた改良を継続的に進める予定である。

注

- 注1) 3D モデリングソフトである Rhinoceros 3D 上で ControlNet を用いた画像生成を行う Plugin (<https://www.food4rhino.com/en/app/ambrosinus-toolkit> 2025 年 7 月 1 日閲覧)
- 注2) OpenAI 社の提案する,システムプロンプトをプリセットすることで,目的に特化したチャットシステムを作成するサービス (<https://openai.com/index/introducing-gpts/> 2025 年 7 月 1 日閲覧)
- 注3) Stability AI 社が開発した operail++ ライセンスの Diffusion モデル (<https://huggingface.co/stabilityai/stable-diffusion-xl-base-1.0> 2025 年 7 月 1 日閲覧)

参考文献

- 1) Ho, J., Jain, A. and Abbeel, P., Denoising Diffusion Probabilistic Models, Advances in Neural Information Processing Systems 33, pp.6840-6851, 2020.
- 2) Ville, P., Jonas, O. and Aku V., Using text-to-image generation for architectural design ideation, International Journal of Architectural Computing, Vol. 22, No. 3, pp.458-474, 2024.
- 3) Zhang, L., Rao, A. and Agrawala, M., Adding Conditional Control to Text-to-Image Diffusion Models, 2023 IEEE/CVF International Conference on Computer Vision, pp3813-3824, 2023.
- 4) Canny, J., A Computational Approach To Edge Detection, IEEE Trans. Pattern Analysis and Machine Intelligence, Vol.8, No.6, pp679-698, 1986.
- 5) Yang, L., Kang, B., Huang, Z., Xu, X., Feng, J. and Zhao, H., Depth Anything: Unleashing the Power of Large-Scale Unlabeled Data, 2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp.10371-10381, 2024.
- 6) Tamenaoul, H., Hamlaoui, M., E., Nassar, M., Prompt Engineering: User Prompt Meta Model for GPT Based Models, Artificial Intelligence, Data Science and Applications, pp.428-433, 2024.
- 7) Liu, V. and Chilton, L., B., Design Guidelines for Prompt Engineering Text-to-Image Generative Models, Proceedings of the 2022 CHI Conference on Human Factors in Computing Systems, No.384, 2022.