

Valutazione nella pratica dell'applicazione dell'intelligenza artificiale nella generazione di render architettonici

(autore articolo: BIS-Lab®)

Alla pluriennale esperienza in ricerca e sviluppo rivolti all'applicazione della modalità operativa Building Information Modeling (BIM), **BIS-Lab®** ha di recente affiancato un particolare interesse sull'utilizzo dell'Intelligenza Artificiale nei processi operativi. Il progetto di ricerca presentato in questo articolo è stato condotto sotto la supervisione **dell'ing. Michele Carradori, Responsabile scientifico di BIS-Lab®**, con la collaborazione di **Leonardo Giacomobono**, studente del corso di alta formazione "Off-site Technologies for Architecture" – edizione 2024, organizzato da **YACademy**, un istituto postgraduate nato dal confronto e dalla collaborazione fra le più illustri firme dell'architettura contemporanea.

1. Introduzione

I primi generatori di immagine, sviluppati nel 2022, sono modelli basati sull'intelligenza artificiale che consentono di generare nuove immagini tramite due meccanismi: l'imitazione delle immagini di partenza su cui sono stati addestrati e l'implementazione di parametri forniti come input. Il risultato, quindi, dipende dal tipo di immagini su cui il modello è stato addestrato. Per questa ricerca sono stati presi in considerazione modelli con database di immagini architettoniche o fotorealistiche.

Esistono due tipologie di meccanismi di generazione principali: Text-to-Image e Image-to-Image:

- nel primo caso, la generazione parte randomicamente da un prompt testuale fornito, contenente la descrizione del risultato cercato e la definizione dello stile grafico;
- nel secondo caso, oltre a questi parametri, viene fornita un'immagine di partenza che verrà successivamente elaborata insieme al prompt. Il risultato sarà quindi maggiormente controllato e sostanzialmente un'evoluzione dell'immagine iniziale.

Il caso studio selezionato per questa ricerca è la progettazione degli interventi denominati "MAS2-Senior Court, Multilab e Parco Intergenerazionale" di Villa Parma, sviluppata da Contec Ingegneria nel 2023 in collaborazione con Studio Zanlari Architettura. Essendo questa ricerca applicata ad un progetto esistente, sono stati presi in considerazione modelli Image-to-Image, inserendo viste di progetto come input. L'obiettivo era la generazione di immagini che potessero essere considerate "render di progetto"; il risultato viene quindi considerato positivo se fedele alla descrizione assegnata, senza modificare la geometria iniziale del progetto.

2. Overview dei principali modelli generativi

La prima attività è stata l'analisi dei principali generatori Image-to-Image specializzati in architettura. La valutazione si è basata sia sulle principali funzioni disponibili nei modelli, sia sulla qualità dei risultati medi ottenibili.

Sono stati individuati 6 modelli:

- Veras
- Stable Diffusion
- Arko AI
- D5 Renderer
- AI
- Visoid

Da un'analisi preliminare sono state ricavate le principali caratteristiche di questi modelli per capirne le diverse funzionalità. Nello specifico, sono state individuate:

- **Tipologia di software:** modalità di esecuzione del modello. In questo caso erano disponibili due possibili scenari, Webapp (esecuzione da browser) o Plug-in (software aggiuntivo utilizzabile in Revit).
- **Image selection:** possibilità di selezionare determinate regioni dell'immagine in output e continuare l'elaborazione solo su quelle aree.

- **Image seed:** associazione di un seed[1] ad ogni immagine generata. Viene valutata la possibilità di estrapolare un seed e inserirlo come parametro in input di successive elaborazioni, al fine di ottenere risultati simili all'immagine da cui è stato preso.
- **Custom style:** possibilità di attribuire diversi stili grafici all'immagine generata.
- **Maintain geometry:** capacità del modello di individuare la geometria dell'immagine in input in modo da essere consistente con l'immagine generata in output.
- **Real time view:** possibilità di generare immagini dalla vista di progetto corrente in tempo reale senza l'esportazione della vista.

Name	Veras	Stable Diffusion	Arko AI	D5 Render	Lookx.ai	Visoid
WebApp	X	X		X	X	X
Plug-in	X		X			
Image Selection	X	X			X	X
Image Seed	X	X			X	
Custom styles		X	X	X	X	X
Maintain geometry	X	X	X	X		X
Real time view					X	

Figura 1 Schema riassuntivo delle diverse funzionalità di ogni programma.

Come è possibile vedere nella Figura 1, i programmi che contengono la maggior parte delle funzionalità e che sono quindi anche maggiormente personalizzabili con possibilità di avere risultati migliori sono *Stable Diffusion*, *Veras* e *Visoid*. La successiva fase di test è stata quindi limitata a questi tre motori.

2.1 Stable Diffusion

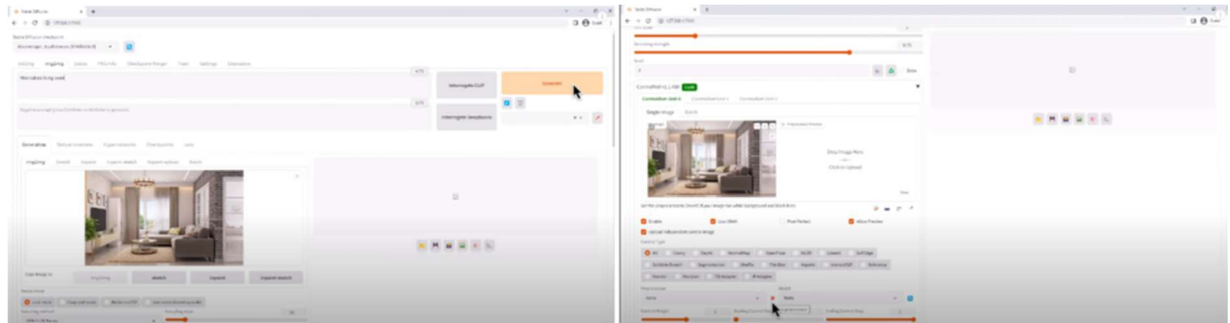


Figura 2 Interfaccia web di Stable Diffusion.

Stable Diffusion è risultato essere il più personalizzabile tra i tre programmi selezionati. Viene eseguito localmente da interfaccia web e, contrariamente agli altri, sfrutta la potenza di calcolo del computer. Aprendo la finestra di generazione *img2img* (*Image-to-Image*) è possibile analizzare le impostazioni su cui si può agire e i parametri di input richiesti.

Al fine di comprendere le variabili modificabili, viene riportato il funzionamento generale del modello. Partendo dall'immagine iniziale, viene aggiunta casualmente una quantità di rumore per gradi. In questa fase, il modello immagazzina dati su come l'aggiunta del rumore modifichi l'immagine; questo processo, dall'immagine al rumore, viene effettuato per sviluppare successivamente il processo contrario, ovvero dal rumore all'immagine. Una volta terminato il processo di aggiunta di rumore, il modello inverte il processo, cercando di rimuoverlo per tornare all'immagine originaria. In questo modo viene addestrato a creare nuovi contenuti partendo da una quantità casuale di rumore iniziale (dipendente dal *seed*).

Le principali variabili della sezione *img2img* sono:

- **Checkpoint:** modello su cui il programma è stato addestrato, ovvero il database di immagini di riferimento che viene usato nel processo di generazione. Modificando questo parametro è possibile influenzare lo stile del risultato ottenuto.
- **Prompt:** descrizione dell'output desiderato.
- **Negative prompt:** descrizione degli elementi indesiderati nell'immagine in output.
- **Sampling method:** tipo di algoritmo utilizzato per la generazione dell'immagine.
- **Resize to:** dimensione dell'immagine in input (è importante mantenere il rapporto originale dell'immagine in input e non l'effettiva risoluzione; è conveniente utilizzare una risoluzione inferiore in modo da avere risultati in tempi brevi).
- **Batch count:** quantità di immagini in output.
- **CFG scale:** definisce quanto il modello si attiene al prompt. A valori più bassi sono collegati output più creativi.
- **Denoise strength:** definisce la quantità di rumore aggiunta all'immagine prima di iniziare il processo iterativo di generazione dell'immagine. A valori bassi corrispondono immagini in output simili a quella in input, mentre a valori alti corrispondono risultati che se ne discostano maggiormente.
- **Seed:** gestisce il rumore iniziale aggiunto all'immagine in input. Diverse immagini generate con lo stesso input mantengono lo stesso carattere.

Queste sono le impostazioni di base più importanti, su cui si può agire per modificare l'output, in modo da avere un risultato congruo al prompt e all'immagine inserita. Oltre a questi valori è stata aggiunta un'estensione (ControlNet 1.5) in grado di estrapolare il *lineart* dell'immagine in input e riutilizzarlo come linea guida all'interno del processo di generazione, in modo da avere risultati che non alterino la geometria di partenza.

2.2 Veras

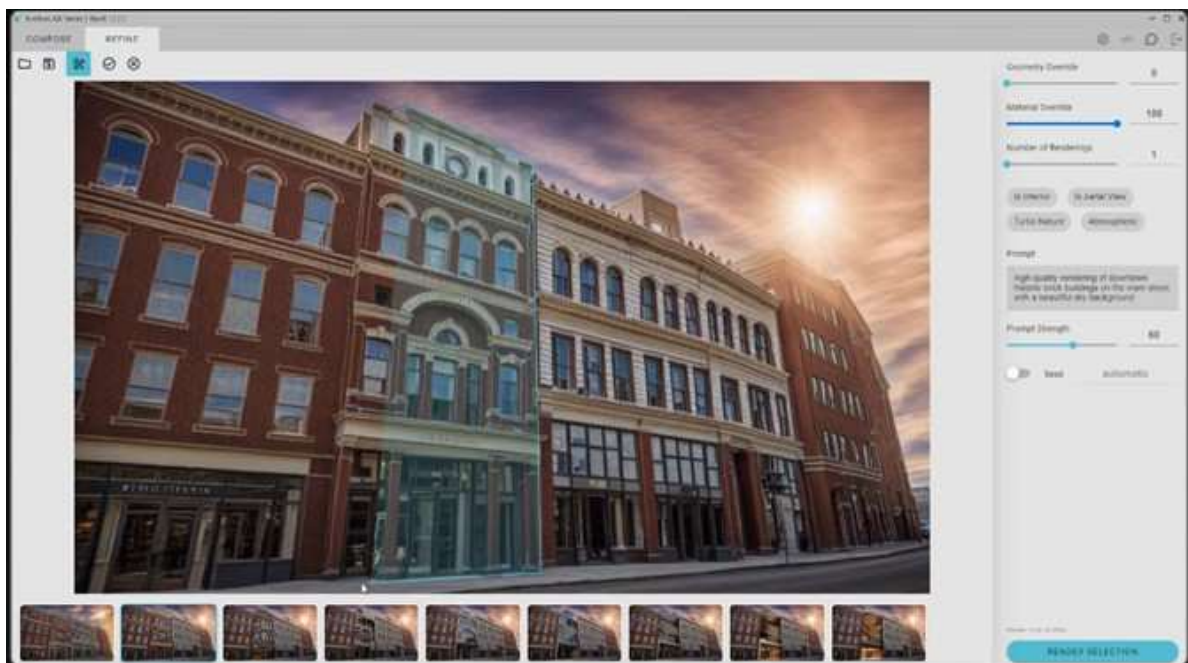


Figura 3 Interfaccia web di Veras.

Nasce come plug-in di render con intelligenza artificiale per Revit e SketchUp; attualmente è possibile utilizzarlo anche tramite interfaccia web. Al contrario di *Stable Diffusion*, *Veras* non utilizza la potenza locale del computer. In Figura 3 si può notare come l'interfaccia sia notevolmente semplificata rispetto a quella di *Stable Diffusion*: molti parametri minori non sono presenti, mentre vengono mostrati solo quelli che influenzano maggiormente l'output. Partendo dall'alto si trova:

- **Geometry override:** determina la fedeltà dell'output alla geometria dell'immagine iniziale.
- **Material override:** determina quanto i materiali dell'immagine originale vengano modificati per rispettare la descrizione del prompt.
- **Tipo di vista:** è possibile scegliere tra Is Interior (viste interne), Is Aerial View (viste dall'alto), Turbo Nature (esterni), Atmospheric (aggiunge nebbia ed effetti ambientali).
- **Prompt**
- **Prompt strenght:** attinenza dell'output al prompt dato.
- **Seed:** possibilità di estrarre il *seed* dell'immagine per avere risultati simili.

2.3 Visoid



Figura 4 Interfaccia web di Visoid.

Come Veras, Visoid non sfrutta la potenza di calcolo locale ed è utilizzabile da web. Presenta un'interfaccia molto semplificata in cui le principali variabili sono:

- **Prompt**
- **Negative prompt**
- **Visualization type:** tipo di vista inserita in input (esterno, interno o dall'alto).
- **Input detail level:** tipo di immagine inserita in input (solo linee, volumi con ombre, render monomaterico, immagine con texture).
- **Style:** stile dell'immagine di output cercato.
- **Input image blend:** controlla quanto l'immagine iniziale sia visibile nell'output; se impostato al massimo, l'immagine originaria non è visibile.
- **Geometric freedom:** determina la fedeltà dell'output alla geometria dell'immagine iniziale.

3. Generazione dei render

Sono state impostate le due camere di progetto da cui estrarre le viste da renderizzare. Sono state testate una vista esterna e una vista interna; per ogni vista sono state poi definite cinque diverse impostazioni grafiche al fine di comprendere quali siano le condizioni migliori per ottenere render soddisfacenti. In particolare, riferendosi ai nomi delle viste in Revit, le viste contengono le seguenti impostazioni:

- Linea nascosta (No ombre, No verde, Linee di contorno, No colori superfici)
- Colori omogenei (No ombre, No verde, Linee di contorno, Colori superfici)
- Colori omogenei con ombre (Ombre ambientali, No verde, Linee di contorno, Colori superfici)
- Colori omogenei con verde (No ombre, Verde, Linee di contorno, Colori superfici)
- Ombreggiata (Ombre portate, No verde, Linee di contorno, Colori superfici)

Si è cercato di non studiare casi contenenti più variabili attive in contemporanea, in modo da comprendere l'incidenza della singola impostazione sull'output del render. Il prompt usato è stato scritto in maniera discorsiva e comprende una breve descrizione degli elementi nella vista. Il prompt è simile per tutti i motori per capire le effettive differenze dei modelli. Vengono successivamente riportati i risultati migliori per ogni motore.

3.1. Stable Diffusion

Linea nascosta



Colori omogenei



Colori omogenei con ombre



Colori omogenei con verde



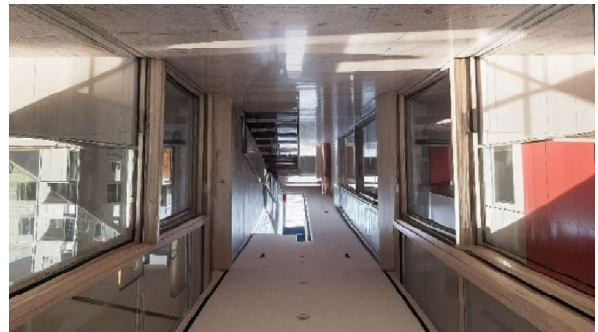
Ombreggiato



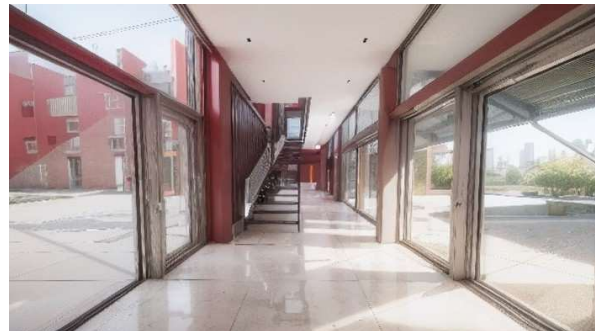
Parametri usati per le viste esterne:

- **Model:** Photon
- **Prompt:** (photorealistic) perspective view of a residential building in a park with bushes on the streets, (group of trees in foreground), smooth red plaster walls, wooden balaustrade, soft colors, lot of grass and vegetation in background
- **Negative prompt:** white walls, wood, cartoon, painting, illustration, (worst quality, low quality, normal quality:1.8)
- **Sampling method:** DPM++2M Karras
- **Denoising:**95
- **CFG scale:** 6
- **Controlnet:** lineart, balanced with 1 of weight

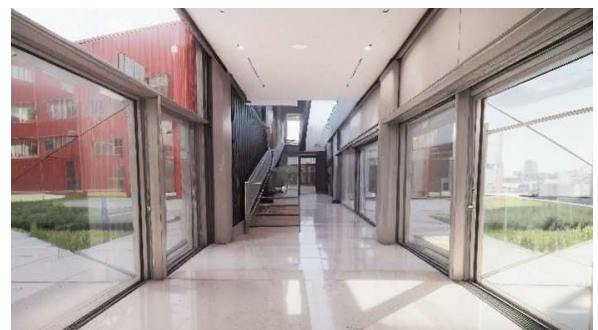
Linea nascosta



Colori omogenei



Colori omogenei con ombre



Colori omogenei con verde



Ombreggiato

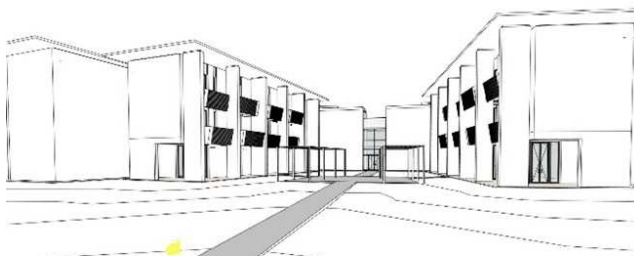


Parametri usati per le viste interne:

- **Model:** Photon
- **Prompt:** (photorealistic) perspective view of a residential building's entrance hall, ((vegetation outside)), tall windows on the sides, steel staircase, red buildings outside, white internal walls and concrete finishing on the floor
- **Negative prompt:** cartoon, painting, illustration, (worst quality, low quality, normal quality:1.8)
- **Sampling method:** DPM++2M Karras
- **Denoising:**65
- **CFG scale:** 6
- **Controlnet:** lineart, balanced with 1.2 of weigh

3.2 Veras

Linea nascosta



Colori omogenei



Colori omogenei con ombre



Colori omogenei con verde



Ombreggiato



Parametri usati per le viste esterne:

- **Geometry override:** 0
- **Prompt:** photorealistic perspective view of a red residential building in a park with bushes on the streets, group of trees in foreground, smooth red plaster walls, wooden railings, soft colors, lot of grass and vegetation in background
- **Material override:** 100
- **Mode:** Turbo nature
- **Prompt strength:** 70

Linea nascosta



Colori omogenei



Colori omogenei con ombre



Colori omogenei con verde



Ombreggiato

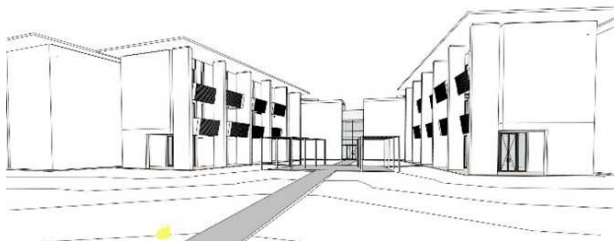


Parametri usati per le viste interne:

- **Geometry override:** 0
- **Prompt:** photorealistic perspective view of a residential building's entrance hall, vegetation outside, tall windows on the sides, steel staircase, red buildings outside, white internal walls and concrete finishing on the floor
- **Material override:** 100
- **Mode:** Is inside
- **Prompt strength:** 50

3.3 Visoid

Linea nascosta



Colori omogenei



Colori omogenei con ombre



Colori omogenei con verde



Ombreggiato



Parametri usati per le viste esterne:

- **Visualization type:** Exterior
- **Prompt:** photorealistic perspective view of a red residential building in a park with bushes on the streets, group of trees in foreground, smooth red plaster walls, wooden railings, soft colors, lot of grass and vegetation in background, in the wood
- **Negative prompt:** white walls, cartoon, painting, illustration, worst quality, low quality, normal quality
- **Input detail level:** Drawings (linea nascosta) -Textured
- **Style:** Render
- **Input image blend:**9
- **Geometric freedom:**2 – 1.8 per la vista con verde

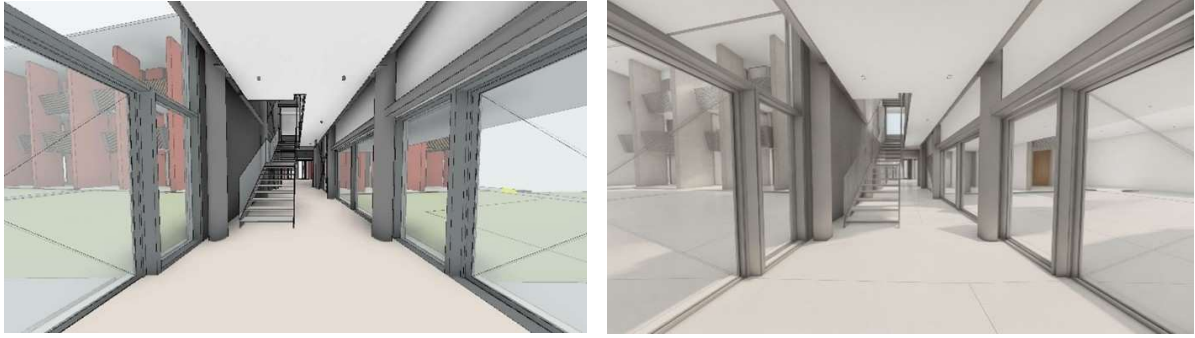
Linea nascosta



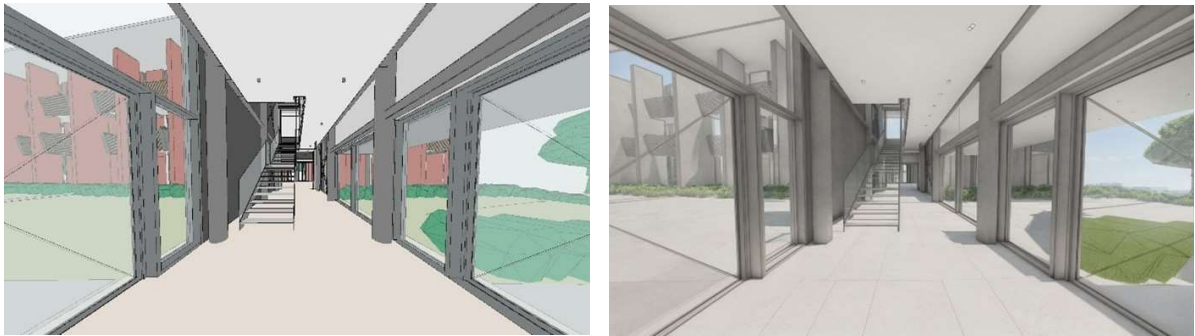
Colori omogenei



Colori omogenei con ombre



Colori omogenei con verde



Ombreggiato



Parametri usati per le viste interne:

- **Visualization type:** Interior
- **Prompt:** photorealistic perspective view of a residential building's entrance hall, vegetation outside, tall windows on the sides, steel staircase, red buildings outside, white internal walls and concrete finishing on the floor
- **Negative prompt:** cartoon, painting, illustration, worst quality, low quality, normal quality
- **Input detail level:** Drawings (linea nascosta) -Textured
- **Style:** Render
- **Input image blend:**8
- **Geometric freedom:** 1

4. Conclusioni

- I render esterni sono risultati più accurati rispetto a quelli interni: tutti e tre i modelli hanno faticato a riconoscere ed elaborare gli elementi nella vista.
- Per quanto riguarda i render esterni, i tre motori hanno dato output abbastanza congruenti con le immagini inserite. Si può notare che impostare come parametro iniziale la congruenza della geometria finale con quella iniziale rende impossibile l'aggiunta di elementi di vegetazione, anche se specificatamente richiesto nel prompt, a meno che questi non siano nella vista originale. Se gli elementi di vegetazione non sono stati modellati nella vista, il

massimo che si riesce ad ottenere sono superfici con erba e bassi cespugli (ad eccezione di *Visoid* dove è stato difficile ottenere anche queste).

- L'utilizzo della vista lineare nascosta rende più difficile ai programmi riconoscere i diversi elementi; mediamente, sia nelle viste esterne che in quelle interne, i render effettuati da vista a linee nascoste presentano più imperfezioni.
- L'utilizzo di viste con colori omogenei facilita il riconoscimento degli elementi. Si può notare come aggiungere le ombre ambientali non migliori l'output generato; invece, aggiungere elementi di vegetazione consente al motore di render di includerli nel risultato.
- La presenza di colori diversi all'interno della vista non corrisponde alla renderizzazione degli stessi colori all'interno dell'output, ma serve solo a facilitare l'elaborazione dell'immagine dal modello. La presenza di un determinato materiale all'interno dell'output dipende solamente dal prompt inserito.
- Nel caso dei render interni si ha maggiore difficoltà a far corrispondere l'elemento in foto con il suo effettivo materiale. Una volta specificato nel prompt i materiali di ogni singolo elemento, non è possibile avere certezza che la corrispondenza definita sia rispettata.
- La generazione di un risultato favorevole è altamente dipendente dal *seed*. È possibile avere risultati sfavorevoli e favorevoli con gli stessi parametri in input, al solo variare del *seed* iniziale.
- Entrando nello specifico dei tre motori usati, sicuramente *Stable Diffusion* è quello che consente un margine di miglioramento dell'immagine più elevato, essendo anche il più personalizzabile. Pur avendo parametri di input simili, *Veras* ha dato risultati migliori di *Visoid* sia per le viste esterne che interne (dove *Visoid* ha avuto difficoltà anche solo ad inserire i materiali).

[1] Il *seed* è un parametro numerico che inizializza il generatore di numeri casuali del modello. Questo numero non ha un significato intrinseco, ovvero non è riconducibile a determinati parametri; si tratta solamente di un parametro di partenza per il processo di generazione.