

# CLASSIFICAÇÃO NÃO SUPERVISIONADA DE IMAGENS EM PROCESSOS DE DESIGN COMPUTACIONAL GENERATIVO

*UNSUPERVISED CLASSIFICATION OF IMAGES IN GENERATIVE COMPUTATIONAL DESIGN PROCESSES*

*CLASIFICACIÓN NO SUPERVISADA DE IMÁGENES EN PROCESOS DE DISEÑO COMPUTACIONAL GENERATIVO*

Daniel R. A. B. Vianna<sup>1</sup>, Claudio F. de Magalhães<sup>1</sup>, Érico F. Mineiro<sup>2</sup>

## RESUMO:

O uso de técnicas computacionais em processos de Design Generativo permite a criação de inúmeras soluções. Para explorar essas iterações, é necessário adotar abordagens que auxiliem na organização e análise desse vasto espaço de soluções. Este estudo propõe uma metodologia baseada em aprendizado profundo para a classificação não supervisionada de imagens, permitindo a exploração automática das iterações de um sistema generativo sem a necessidade de rotulagem manual. A pesquisa utiliza o programa Grasshopper para gerar as iterações e transformá-las em imagens, um autoencoder convolucional para extrair representações compactas das imagens, técnicas de redução de dimensionalidade como UMAP e o algoritmo de Misturas Gaussianas para agrupamento dos dados. O trabalho demonstra que essa abordagem pode organizar o espaço de soluções com base em características morfológicas e espaciais, contribuindo para novas formas de interação entre o designer e a inteligência artificial no campo do design. Os códigos e imagens utilizadas estão disponíveis aqui: <https://github.com/researchai2/cluster2>.

**PALAVRAS-CHAVE:** aprendizado profundo; exploração do espaço das soluções; *autoencoder* convolucional; estudos tridimensionais.

## ABSTRACT:

The use of computational techniques in Generative Design processes allows for the creation of numerous solutions. To explore these iterations, it is necessary to adopt approaches that help organize and analyze this vast solution space. This study proposes a deep learning-based methodology for unsupervised image classification, enabling the automatic exploration of generative system iterations without the need for manual labeling. The research uses the Grasshopper software to generate the iterations and transform them into images, a convolutional autoencoder to extract compact representations of the images, dimensionality reduction techniques such as UMAP, and the Gaussian Mixture Model algorithm for data clustering. The work demonstrates that this approach can organize the solution space based on morphological and spatial characteristics, contributing to new forms of interaction between the designer and artificial intelligence in the design field. The codes and images used are available here: <https://github.com/researchai2/cluster2>.

**KEYWORDS:** deep learning; exploration of the design space; convolutional autoencoder; three-dimensional studies.

<sup>1</sup> Universidade Católica, Rio de Janeiro. Programa de Pós-Graduação em Design, Departamento de Artes e Design, Pontifícia

<sup>2</sup> Universidade Federal de Minas Gerais

**Fonte de Financiamento:**  
CNPq processo  
167307/2018-5.

**Conflito de Interesse:**  
Não há.

**Submetido em:** 31/07/2024  
**Aceito em:** 03/12/2024

How to cite this article:

VIANNA, D. R. A. B. et. al. Classificação não supervisionada de imagens em processos de design computacional generativo. Título do Artigo. **Gestão & Tecnologia de Projetos**. São Carlos, v19, n3, 2024. <https://doi.org/10.11606/gtp.v19i3.227541>



## **RESUMEN:**

El uso de técnicas computacionales en procesos de Diseño Generativo permite la creación de innumerables soluciones. Para explorar estas iteraciones, es necesario adoptar enfoques que ayuden a organizar y analizar este vasto espacio de soluciones. Este estudio propone una metodología basada en el aprendizaje profundo para la clasificación no supervisada de imágenes, lo que permite la exploración automática de las iteraciones de un sistema generativo sin la necesidad de etiquetado manual. La investigación utiliza el programa Grasshopper para generar las iteraciones y transformarlas en imágenes, un autoencoder convolucional para extraer representaciones compactas de las imágenes, técnicas de reducción de dimensionalidad como UMAP y el algoritmo de Mezclas Gaussianas para el agrupamiento de datos. El trabajo demuestra que este enfoque puede organizar el espacio de soluciones basado en características morfológicas y espaciales, contribuyendo a nuevas formas de interacción entre el diseñador y la inteligencia artificial en el campo del diseño. Los códigos e imágenes utilizados están disponibles aquí: <https://github.com/researchai2/cluster2>.

**PALABRAS CLAVE:** aprendizaje profundo; exploración del espacio de soluciones; autoencoder convolucional; estudios tridimensionales.

## INTRODUÇÃO

Como já ocorreu anteriormente, a introdução de uma nova tecnologia vem prometendo transformar o campo do Design. Dessa vez, a inovação vem do campo da Inteligência Artificial, com as técnicas de aprendizado profundo. Compreender como avanços tecnológicos impactam o Design é uma forma para o campo tornar-se mais consciente do potencial que tem em mãos e assumi-los (NATIVIDADE, 2010). Nesse contexto, esse artigo tem como objetivo investigar o potencial do aprendizado profundo na filtragem ou análise das iterações de um processo generativo. Para isso, esse artigo propõe uma estrutura metodológica para explorar as iterações de um processo de Design Computacional Generativo utilizando uma técnica de classificação não supervisionada de imagens. O estudo apresenta como recorte a classificação das iterações à luz de seus aspectos morfológicos e espaciais.

Esta pesquisa adota como base metodológica a Design Science Research (DSR). Essa metodologia é orientada a pesquisas cujo objetivo é criar, desenvolver e explorar novas soluções. O DSR pode ser utilizado quando existir a intenção de explorar e prescrever novas soluções para problemas relevantes existentes (VAN AKEN, 2005).

O DSR demonstra o uso de um artefato para resolver uma ou mais instâncias de um problema. Isso pode envolver seu uso em experimentação, simulação, estudo de caso, prova ou outra atividade apropriada (PEFFERS et al., 2007). Especificamente, o presente artigo visa aprimorar um método através de um estudo exploratório. O trabalho se estrutura em cinco partes, notadamente: introdução, revisão narrativa da literatura; método; discussão e resultados e as considerações finais.

Essa introdução apresenta a estrutura da pesquisa. A revisão da literatura sintetiza o conhecimento existente sobre o tema de maneira abrangente e embasa a metodologia proposta. A seção de método descreve o estudo exploratório sendo composta pelas seções de desenvolvimento das entradas, extração de dados, agrupamento, visualização e avaliação dos resultados. Em seguida são discutidos alguns pontos do processo e apresentados os resultados do estudo. A parte final aponta aspectos que podem ser refinados no modelo proposto e indica caminhos para pesquisas futuras.

A pesquisa sugere que a estrutura metodológica proposta possibilita a organização do espaço das soluções de um sistema generativo sem a necessidade do designer rotular nenhuma iteração. Acredita-se que o trabalho contribua com a prática e a teoria do Design ao trazer uma técnica pouco utilizada pela área. Em especial, buscou-se aprofundar no estudo a relação iterativa ou o “diálogo” entre o designer e a inteligência artificial para calibrar o resultado do método.

## REVISÃO DA LITERATURA

O Design Computacional Generativo é formado pela adoção de um sistema generativo em meio computacional. A lógica desse tipo de sistema parte do processo de passar por muitos ciclos de aprendizado (gerar, analisar, aprender, modificar) até que se encontre uma solução satisfatória para o design (FISCHER; HERR, 2001). Com o uso da computação associado a um sistema generativo, um designer consegue gerar uma quantidade muito grande de possíveis soluções em um curto espaço de tempo. Logo a utilização de uma ferramenta para explorar esse universo é imprescindível.

Ao longo da história do Design Digital, pesquisadores vêm desenvolvendo e revisando sistematicamente ferramentas para explorar o espaço do design (BROWN; MUELLER, 2017).

Entre as lacunas percebidas nessas pesquisas está a incapacidade desses instrumentos em compreender a qualidade que envolve aspectos perceptuais da forma e do espaço (BHATT; SCHULTZ; HUANG, 2012). De fato, tipicamente, as explorações computacionais são endereçadas a preocupações de otimização estrutural, uso de material e responsividade a fatores ambientais, ao invés de aspectos natureza social (AHLQUIST, 2020). Outro problema observado pelos autores é o fato de que a maioria das pesquisas que buscam compreender e organizar o espaço do design através de técnicas de classificação, adotam meios supervisionados, onde o designer humano precisa indicar exatamente o que máquina deve fazer. Esse controle do designer pode limitar a capacidade da máquina em achar padrões, bem como envolve esforços significativos de programação (JOHANES; HUANG, 2021).

Com a recente introdução do aprendizado profundo no Design, novas formas de exploração das iterações de um sistema vêm surgindo. Junto a elas surgem novas oportunidades para preencher algumas lacunas e aprimorar as antigas técnicas de exploração (HUANG et al., 2021). Antes de qualquer coisa é importante esclarecer a diferença entre o aprendizado de máquina e o aprendizado profundo.

O aprendizado de máquina é um tipo de Inteligência Artificial que usa um grande conjunto de dados para treinar a si mesmo. Ele melhora a sua capacidade de realizar uma operação através da sua experiência a partir de um código escrito para classificar o que se espera que ele faça. Já o aprendizado profundo, se refere a um desenvolvimento mais recente em que a máquina não depende de instruções explícitas programadas para realizar suas tarefas. Em vez disso, a programação identifica e organiza informações, ensinando a si mesmo sobre determinado assunto com base em um conjunto de dados fornecido (FAZI, 2020). Isto é, essas máquinas formulam padrões e relações dentro de um conjunto de dados fornecido, sem intervenção humana direta para definir as regras de suas operações (LEACH, 2022).

Atualmente, as técnicas de aprendizado profundo mais utilizadas no campo do Design são as que auxiliam nas tarefas de síntese de imagens e as de classificação ou regressão (LEACH, 2022). As técnicas de síntese como o sonho profundo (*deep dream*) e as redes adversárias gerativas (*GANs*) possibilitam a geração de imagens. O sonho profundo usa redes neurais convolucionais que são alimentadas com uma imagem e ao invés de apenas reconhecer padrões, ela exagera certas características que já identificou, criando imagens surreais com formas que parecem se repetir e se intensificar, como texturas ou objetos (MORDVINTSEV; OLAH; TYKA, 2015).

As Redes Adversárias Gerativas consistem em duas redes neurais que competem entre si, uma geradora e outra discriminadora. A geradora cria imagens a partir de um ruído aleatório, enquanto a discriminadora tenta distinguir entre imagens reais e as geradas. Esse processo contínuo leva a uma melhoria da qualidade das imagens geradas, que se tornam cada vez mais realistas à medida que a geradora “aprende” a enganar a discriminadora (GOODFELLOW et al., 2014).

As técnicas de classificação ou regressão avaliam um determinado conjunto de dados. A diferença entre as tarefas de classificação de imagem e a regressão de imagem é que a variável alvo (o que estamos tentando prever) na tarefa de classificação não é contínua, enquanto na tarefa de regressão é contínua (LEACH, 2022). Por exemplo, classificar um conjunto de imagens nas categorias cães e gatos é uma tarefa de classificação. Já prever a idade de alguém ou os preços de casas com base nas suas imagens é uma tarefa de regressão.

Existem três tipos de classificações, notadamente: as supervisionadas, as não supervisionadas e as semi-supervisionadas. Na classificação supervisionada, o algoritmo aprende a partir de um conjunto de dados rotulado. Ou seja, cada imagem de treinamento é relacionada a uma categoria de saída. Assim, o modelo é treinado até poder detectar padrões entre as imagens e

a sua classificação, o que permite que ele preveja rótulos de imagens que não foram utilizadas no treinamento. A classificação não supervisionada envolve modelar a estrutura ou distribuição de dados extraídos de um conjunto de imagens para identificar padrões sem um rótulo prévio. Essa classificação primeiro precisa de algum tipo de algoritmo capaz de extrair dados das imagens para depois gerar a classificação propriamente dita. A classificação semi-supervisionada consiste numa mistura dos outros dois processos. Ela usa uma pequena quantidade de dados rotulados junto com uma quantidade maior de dados não rotulados. Se bem-sucedido, o algoritmo consegue aprender a estrutura do conjunto maior de dados não rotulados a partir de uma amostragem de imagens já classificadas que ajudam a direcionar o sistema (SCHMARJE et al., 2021).

A escolha entre essas três abordagens de classificação geralmente depende do problema específico em questão, da natureza dos dados disponíveis e do resultado desejado. A adoção da técnica não supervisionada nesse estudo se dá pelo fato das iterações geradas por um processo de Design Computacional Generativo tenderem a não serem predeterminadas o que pede um processo que não exija dados rotulados. Fora isso, esse tipo de classificação possui a vantagem de revelar ou propiciar descobertas de padrões ocultos em um conjunto de elementos (JOHANES; HUANG, 2021). É válido ressaltar que os processos não supervisionados tendem a ser menos precisos e mais subjetivos em comparação com os outros tipos de classificação.

Apesar de não serem comuns, existem alguns trabalhos que utilizam técnicas de classificação não supervisionada no campo do Design, especialmente na área da Arquitetura. Frederick Chando Kim e Jeffrey Huang explicam as diferentes maneiras de representação de formas tridimensionais em dados computáveis (2022). Ahmed M. Abuzurairq e Halil Erhan apresentam uma série de conceitos relevantes sobre o conceito de similaridade, bem como sobre técnicas de agrupamento e técnicas de redução de dimensionalidade (2020). Mikhael Johannes e Jeffrey Huang propõem uma estrutura experimental de classificação não supervisionada que utiliza um *autoencoder*, um algoritmo de agrupamento (*clustering*) e um processo de visualização, especialmente influentes na elaboração do presente estudo (2021).

Nesse contexto, esse artigo apresenta e discute um método de exploração das iterações de um sistema generativo pouco explorado pelas disciplinas do Design, notadamente aquele que utiliza um algoritmo de auto codificação seguido de outro de classificação não supervisionada de imagens. Enquanto o primeiro, extrai dados de representações bidimensionais das iterações geradas, o segundo organiza essas informações em categorias distintas.

## MÉTODO

Esse estudo desenvolve uma estrutura para classificar de maneira não supervisionada as iterações de um sistema generativo computacional através da captação de imagens. Para isso são utilizadas diferentes ferramentas ao longo das etapas de: Desenvolvimento das entradas, extração de dados, agrupamento, visualização e avaliação dos resultados, como pode-se observar no diagrama da figura 1. A seta indicada com a palavra retorno sinaliza que existe uma iteração entre a estrutura do método e as avaliações de possíveis resultados. Essa iteração entre método e resultado torna-se uma via de mão dupla, onde ambos se influenciam mutuamente em um processo de co-evolução. Esse processo contínuo garante que o artefato final seja tanto uma solução prática quanto uma contribuição teórica robusta.

Em outras palavras, cada ajuste ao método envolve uma reflexão que permite identificar padrões, limitações e oportunidades, que podem beneficiar tanto aplicações práticas quanto a construção de um corpo teórico mais sólido e expansivo

Todo o processo que leva a preparação do conjunto de imagens é realizado usando o programa *Grasshopper*. As etapas de extração de dados e agrupamento são conduzidas na plataforma *Google Colaboratory*.

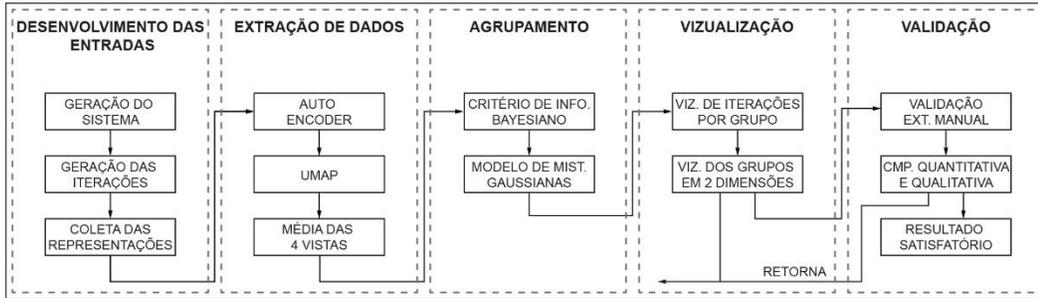


Figura 1. Diagrama da estrutura metodológica do estudo exploratório.

Fonte:  
Autores

## DESENVOLVIMENTO DAS ENTRADAS

A primeira etapa do estudo consiste em desenvolver as representações das iterações que serão usadas como entrada no processo de classificação dos grupos. Essa fase pode ser subdividida na criação de um sistema generativo, na geração das suas iterações e na coleta das suas representações em uma determinada mídia e formato. As iterações são composições tridimensionais geradas por um sistema generativo computacional, as representações se dão por quatro vistas ortogonais laterais e a mídia adota o meio digital em formato JPEG (*Joint Photographic Experts Group*).

## EXTRAÇÃO DE DADOS

A próxima etapa do estudo consiste em extrair as características das imagens de entrada e as transformar em um conjunto de dados matemáticos. Nessa fase é utilizado um *autoencoder*, uma técnica de redução de dimensões e uma técnica de encontro da média das quatro imagens.

*Autoencoder* é um tipo de rede neural convolucional projetada para aprender a codificação de dados de entrada em uma representação interna de dimensão inferior e reconstrução de volta para corresponder aos dados de entrada (GOODFELLOW et al., 2014). Ou seja, ele aprende padrões visuais hierárquicos importantes de uma imagem, transforma esses padrões em uma representação matemática compacta e informativa e depois tenta recriar a imagem através dessa representação.

Existem diversos tipos de *autoencoders* como o *sparse autoencoder*, o *variational autoencoder* (VAE) e o *convolutional autoencoder* (CAE), por exemplo. Aqui foi utilizado o *convolutional autoencoder* (CAE) pois esta variedade é especialmente útil para tarefas onde a preservação da estrutura visual da imagem é crítica (JANNANI; IDRISSEI; CHAKIB, 2023). O *autoencoder* foi implementado com as bibliotecas *Tensorflow* e *Keras* (ABADI et al., 2016; CHOLLET, 2015).

Redução de dimensionalidade é um conjunto de técnicas que auxiliam na análise de dados multidimensionais ao minimizar as variáveis de um sistema ou ao mapear os dados para um novo espaço, menor (ABUZURAIQ; ERHAN, 2020). Isso não apenas facilita a visualização e

interpretação dos dados (permitindo-nos visualizar agrupamentos ou padrões) mas também pode melhorar a eficiência e performance de algoritmos de aprendizado de máquina subsequentes, que frequentemente têm desempenho reduzido em espaços de alta dimensão (MCINNES; HEALY; MELVILLE, 2018).

Existem diversas técnicas que reduzem as dimensões de conjuntos de dados com múltiplas dimensões, como a PCA (Análise de Componentes Principais), a t-SNE (Incorporação Estocástica de Vizinhos distribuída em t) ou a UMAP (Aproximação e Projeção de *Manifold* Uniforme). Aqui definiu-se pelo emprego da UMAP devido a sua eficácia tanto na preservação das estruturas locais (relações de proximidade entre pontos semelhantes) quanto das globais (a estrutura geral dos dados) dos dados (MCINNES; HEALY; MELVILLE, 2018).

A última etapa dessa fase envolve combinar os dados das imagens captadas. Isso acontece porque cada elemento tridimensional foi representado por quatro imagens sem uma ordem específica. Assim é preciso que a notação matemática delas seja unida em uma só.

## AGRUPAMENTO

A etapa de agrupamento compara e classifica os elementos gerados pelo sistema generativo através dos dados extraídos e processados de suas imagens. Aqui foi utilizado o Critério de Informação Bayesiano (BIC) (SCHWARZ, 1978) e o algoritmo de agrupamento do modelo de Misturas Gaussianas (PEDREGOSA et al., 2011). Enquanto o primeiro método analisa os dados e sugere um número de grupos para a classificação, o segundo gera o agrupamento propriamente dito. O BIC foi utilizado porque o Modelo de Misturas Gaussianas pede que o usuário insira o número de grupos desejado.

O algoritmo de agrupamento classifica itens com base em sua similaridade. A similaridade entre duas alternativas pode ser definida em termos de um subconjunto ou de todos os seus dados associados, como parâmetros de entrada, métricas de desempenho ou suas características geométricas (ABUZURAIQ; ERHAN, 2020).

Existem diversos algoritmos de agrupamento (*clustering*) e no presente estudo, optou-se pelo uso do Modelo de Misturas Gaussianas em detrimento de métodos como *K-means*, *DBSCAN* e do agrupamento hierárquico, por exemplo. Essa decisão se justifica pela capacidade superior da técnica escolhida modelar grupos com tamanhos variados e a sobreposição inerentes aos dados envolvidos no estudo (XU; WUNSCH, 2005). Aqui foi utilizada a biblioteca *scikit-learn* com a classe *GaussianMixture* (PEDREGOSA et al., 2011)

## VISUALIZAÇÃO

A próxima etapa consiste na elaboração de duas ferramentas empregadas na visualização da classificação. A primeira permite a visualização de um determinado número de iterações por grupo e a outra localiza todas as iterações mapeadas em um espaço bidimensional. Ambas as ferramentas pegam dados gerados nas fases anteriores e geram as interfaces através de um algoritmo desenvolvido no programa *Grasshopper*.

## AValiação DOS RESULTADOS

A última etapa consiste na avaliação dos resultados. Na classificação não supervisionada, não existem dados rotulados para testar a precisão como se faria no caso de um processo de

classificação supervisionado. Nesses casos, uma das maneiras de avaliação, é utilizando métodos de validação externa, que são aqueles que se baseiam em critérios ou informações externas ao conjunto de dados original sobre o qual o modelo foi treinado.

Assim, o trabalho adota o método de validação externa manual, em que especialistas humanos analisam os agrupamentos ou padrões identificados pela máquina e tentam replicá-los. A classificação manual então é comparada quantitativamente e qualitativamente com as classificações ou agrupamentos realizados pelo modelo. A comparação quantitativa, mede a concordância entre os agrupamentos da máquina e os rótulos fornecidos pelos especialistas humanos com base nos mesmos rótulos da máquina. Já a comparação qualitativa discute a coerência, a relevância e a interpretabilidade dos agrupamentos ou padrões identificados.

## DISCUSSÃO E RESULTADOS

O ponto da estrutura proposta é a criação de um sistema generativo, concebido para produzir iterações de geometrias simples, facilitando a compreensão do designer e assegurando um maior controle do estudo. O sistema generativo utilizado foi baseado na metodologia de ensino de fundamentos do Design da professora Rowena Reed (ROWENA REED KOSTELLOW FUND - 3-D DESIGN EDUCATION RESOURCE, [s.d.]). Essa metodologia foi desenvolvida ao longo de cinco décadas, principalmente no Pratt Institute, onde ainda hoje, os experimentos são ministrados.

O método de ensino de Reed consiste em um processo cíclico de divergência e convergência, em que os alunos criam alternativas continuamente e escolhem as mais adequadas com base nas relações entre os elementos do Design. Essas relações são comportamentos ou propriedades formais que podem emanar certos estímulos sensoriais e significados incorporados. Assim, elas servem como recursos analíticos para perceber particularidades das composições formais e como a percepção sensorial responde a elas (HANNAH, 2002). A adoção dessa metodologia se deve ao seu foco na lógica generativa, bem como no fato dela fornecer uma lógica de classificação clara e concisa. Isso é essencial para compreender os fenômenos associados ao estudo sem a interferência de ruídos.

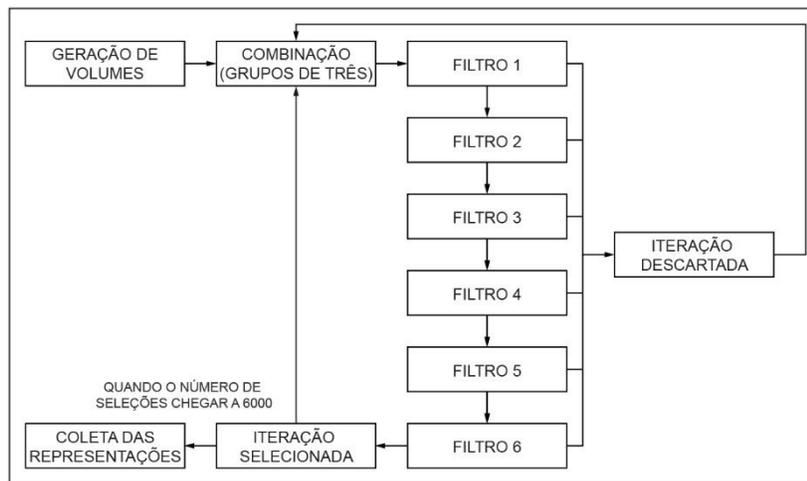
O sistema generativo desenvolvido para o estudo se baseia no primeiro problema da metodologia da professora Rowena Reed, chamado de volumes retilíneos. Ele consiste na organização de três volumes retilíneos, e o seu objetivo é familiarizar-se com volumes simples em uma declaração unificada (HANNAH, 2002), seguindo os seguintes princípios:

- Os volumes devem variar em caráter. O caráter de cada volume é definido pelas suas proporções tridimensionais. Neste exercício, o caráter de cada forma pode variar entre: linear, planar e volumétrico.
- Todas as dimensões lineares e proporções planares que compõem os volumes e os resultados de suas combinações devem variar. Isso inclui as dimensões e proporções dos espaços negativos e a massa total de toda a composição. As proporções quadradas ou cúbicas devem sempre ser evitadas.
- Os volumes devem constituir um relacionamento hierárquico, escolhendo formas dominantes, subdominantes e subordinadas. Onde o dominante é a maior forma, o mais impactante do conjunto e deve ser colocado em uma posição de destaque. A subdominante deve ser a segunda maior peça e deve criar uma relação contrastante com a peça dominante. O subordinado é o menor volume e muito dependente dos outros dois. Ele deve tornar o design mais tridimensional, complementar os outros volumes e completar a unidade do design.

- Nenhuma visão deve ser desinteressante repetida ou espelhada.
- Os eixos principais das formas devem ser perpendiculares entre si para dar mais tridimensionalidade ao projeto.
- Os encontros entre as peças devem ocorrer de três formas: a cunha, o berço e a perfuração. Os encontros não devem se repetir e devem ser estruturais.
- As composições devem apresentar um equilíbrio estrutural e visual.

Com essas diretrizes, foi elaborado um sistema generativo de expressão paramétrica, desenvolvido no programa Grasshopper, capaz de gerar peças variadas atendendo aos requisitos e restrições impostas pelo exercício. O algoritmo começa criando uma grande quantidade de volumes retilíneos randomicamente, sem a necessidade do usuário fornecer variáveis de entrada. A geração aleatória dos volumes é controlada por uma semente (*seed*) definida a partir de um contador baseado no tempo (em segundos).

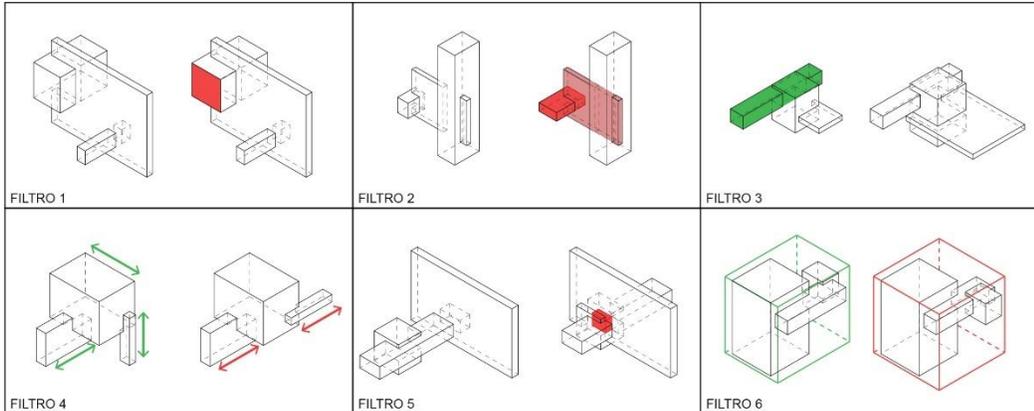
Em seguida três volumes são selecionados randomicamente e combinados de modo a formar uma peça única. Depois, essas agregações são analisadas e selecionadas através de fórmulas matemáticas simples. Ou seja, selecionam-se aqueles que atendem aos requisitos e as restrições do problema e descartam-se as demais. A figura 2 apresenta o diagrama do método do sistema generativo proposto.



**Figura 2.** Diagrama da estrutura processual do sistema generativo.

**Fonte:**  
Autores

As seleções são feitas através de seis filtros. O primeiro, descarta grupos que possuam uma de suas faces quadradas; o segundo elimina os grupos que possuam dois ou três blocos com caracteres iguais; o terceiro descarta aquele conjunto que não apresenta uma relação hierárquica de dominância clara; o quarto elimina os agrupamentos cujas peças repetem a sua direção; o quinto descarta grupos que possuam interseções entre os três blocos; e o último elimina aqueles conjuntos que formam uma proporção próxima a quadrada ou cúbica na proporção geral combinada. A Figura 3 exemplifica os casos em que as peças são aceitas (à esquerda) e descartadas (à direita).



**Figura 3.** Exemplificação dos filtros.

**Fonte:**  
Autores

Quando a peça é excluída por um dos filtros, o algoritmo retorna a etapa de combinação. Quando a peça não é filtrada, ela é armazenada e o processo retorna a mesma etapa de combinação. O processo iterativo só acaba quando o número de itens armazenados chega a 6000.

Com todas as iterações armazenadas, pode-se coletar os dados que serão utilizados como entrada para a próxima etapa. Essa etapa é complicada porque, esbarra em um dos desafios fundamentais da visão computacional, que é tirar inferências sobre o mundo tridimensional a partir de imagens bidimensionais. Vários métodos foram desenvolvidos para esse fim através de diferentes formatos de dados, como múltiplas imagens bidimensionais, *voxels*, nuvens de pontos ou *wireframes* (KIM; HUANG,2022). O estudo experimental proposto aqui utiliza o método de múltiplas imagens bidimensionais devido ao fato dessa técnica demandar um menor poder de computação.

Com a mídia escolhida, foi preciso definir uma série de parâmetros que afetam diretamente o resultado do estudo. Esses parâmetros são as suas dimensões (224 x 224, 112 x 112 ou 56 x 56 pixels), o número de imagens (1, 4, 6 ou 8), o tipo de lente (paralela ou perspectivada), as vistas (isométrica, ortogonal ou mista), o sombreado (com ou sem), o fundo (branco ou preto) e a coloração (cinza, branca ou colorida).

As escolhas dessa etapa se deram por experiências prévias de outros pesquisadores (LI; CORNEY, 2023) e experimentações. Devido ao grande número de combinações possíveis entre os parâmetros utilizados neste estudo, mais precisamente 1152, seria impraticável realizar testes com todas as combinações. A multiplicidade de variáveis envolvidas, quando cruzadas, resulta em um espaço de experimentação muito extenso, o que tornaria o processo inviável. Além disso, as avaliações das representações foram feitas no final da estrutura processual, na fase de visualização, o que tornou essa tarefa muito dispendiosa em termos de tempo e recursos.

Assim, optou-se por uma abordagem de amostragem. Foram selecionados subconjuntos representativos das combinações de parâmetros, de forma a garantir a exploração de diferentes cenários relevantes. Essa seleção foi realizada tentando priorizar variáveis com mais impacto, ao mesmo tempo visando que as combinações selecionadas cobrissem de forma equilibrada o espaço das possíveis soluções.

Analisando os resultados através das ferramentas de visualização definiu-se que as representações seriam constituídas por um grupo de quatro imagens separadas; de dimensões de 56 x 56 pixels; captadas por lentes paralelas; em disposições de vistas laterais ortogonais sem sombreado, com fundo branco, sem arestas marcadas e com cores identificando a dominância dos volumes. As versões coloridas apresentam os elementos dominantes na cor

verde, os elementos subdominantes na cor azul e os elementos subordinados na cor vermelha. A figura 4 reúne quatro exemplos de representações que foram testadas. O exemplo de número 4 foi o selecionado para o estudo por apresentar os melhores resultados analisados observando as ferramentas de visualização.

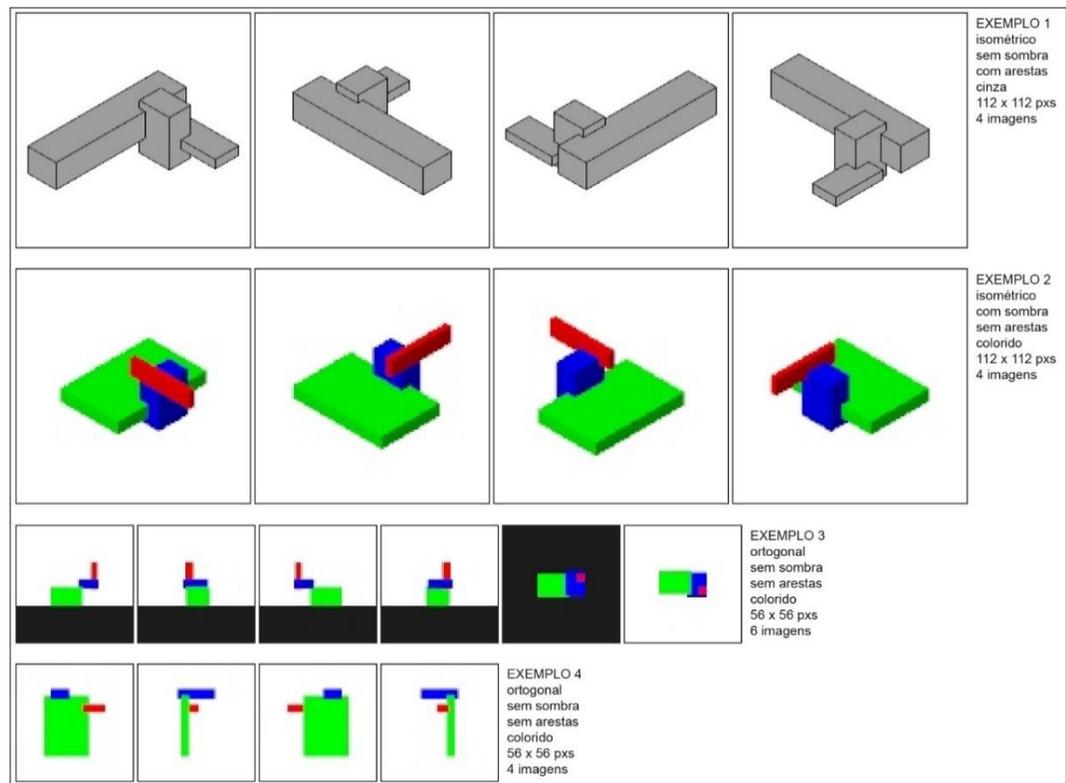


Figura 4. Modos de representação das iterações.

Fonte:  
Autores

Ainda utilizando o programa *Grasshopper*, o algoritmo gerou as quatro imagens para cada uma das seis mil iterações geradas, resultando em um total de vinte e quatro mil imagens. Cada imagem recebeu um nome formado pelo código numérico do seu grupo (de 0 a 5999) seguido do código numérico de identificação de cada imagem dentro de cada grupo (de 0 a 3).

É importante destacar que um conjunto de iterações pode ser classificado de diversas maneiras, dependendo dos aspectos observados. Por exemplo, pode-se focar no encontro da peça com o solo, na sua verticalidade ou horizontalidade ou no tipo de encontro entre os blocos de uma composição entre outros aspectos. Nesse sentido, uma representação orienta a máquina sobre os aspectos que devem ser priorizados e, conseqüentemente, influencia a direção que a classificação irá tomar (ABUZURAIQ; ERHAN, 2020).

Em especial, duas variáveis das representações se mostraram muito influentes na melhora do desempenho do agrupamento: As versões coloridas e as versões com as menores dimensões. Acredita-se que a distinção das cores dos volumes aqui desempenhou um papel crucial ao sinalizar para a máquina a relevância do caráter e do tamanho dos volumes na classificação. Da mesma forma, foi observado um melhor desempenho do agrupamento gerado com as representações de imagens de dimensões menores. Isso pode ser atribuído a classificação ter se concentrado nas características gerais dos elementos, ao invés de detalhes locais ou específicos. Essa percepção vem da ideia de que a diminuição das dimensões de uma imagem faz com que um modelo se concentre em características mais amplas é um conceito comum no campo da visão computacional (CHEN et al., 2024; ZHANG et al., 2020).

Com as imagens prontas, a próxima etapa transforma os conjuntos de imagens em dados através do uso de um *autoencoder*, da técnica de redução de dimensão UMAP e do encontro da média da versão numérica das quatro imagens.

Tanto o *autoencoder* quanto o UMAP requerem a definição de variáveis. No caso do autoencoder, é preciso definir a arquitetura da rede, a função de perda personalizada (*ssim\_loss*), o tipo de otimizador (Adam), a sua taxa de aprendizado (0.0002), o número de épocas e do *batch* do treinamento (40 e 40). No UMAP, o número de componentes (2), o número de vizinhos (5), a distância mínima entre os pontos (0.8) e a métrica de similaridade (euclidean) foram ajustadas para reduzir a dimensionalidade e preservar a estrutura dos dados.

No *autoencoder*, a arquitetura da rede define quantas camadas e neurônios haverá na rede neural, influenciando como os dados serão comprimidos e reconstruídos. A função de perda personalizada, neste caso chamada de *ssim\_loss*, avalia o erro entre a imagem original e a reconstruída, ajudando a treinar o *autoencoder* para melhorar a qualidade da reconstrução. O otimizador, definido como *Adam*, é o algoritmo que ajusta os pesos da rede neural durante o treinamento para minimizar a função de perda. A taxa de aprendizado (0.0002) controla o quão grandes são as atualizações dos pesos em cada iteração, influenciando a velocidade e a precisão do treinamento. O número de épocas (40) refere-se a quantas vezes o modelo terá todos os seus dados de treinamento ajustados, enquanto o *batch size* (40) indica quantos exemplos de treinamento são processados antes da atualização dos pesos da rede.

No caso do UMAP, a técnica de redução de dimensionalidade, os parâmetros ajustados incluem o número de componentes (2), que define para quantas dimensões os dados serão reduzidos (neste caso, de um espaço de alta dimensão para um espaço bidimensional). O número de vizinhos (5) determina quantos pontos próximos são considerados para construir a estrutura local dos dados, influenciando a forma como os clusters são preservados na redução de dimensionalidade. A distância mínima entre os pontos (0.8) controla a dispersão entre os pontos no espaço reduzido, afetando a separação entre diferentes grupos de dados. Por fim, a métrica de similaridade, definida como *euclidean*, especifica a forma como a distância entre os pontos é medida, sendo a distância euclidiana (comum) escolhida para calcular as proximidades no espaço de dados.

Com os dados extraídos dos conjuntos de imagens extraídos o próximo passo é operar o algoritmo de agrupamento. Porém, como o Modelo de Misturas Gaussianas precisa que o usuário selecione quantos grupos serão formados, antes disso é preciso definir essa quantidade. A seleção do número de grupos é crucial para o desempenho do agrupamento pois garante que o modelo não seja demasiadamente simples nem excessivamente complexo. Existem diferentes métodos que auxiliam na definição de um número adequado a classificação de um conjunto e nesse trabalho foi adotado o Critério de Informação Bayesiano (BIC).

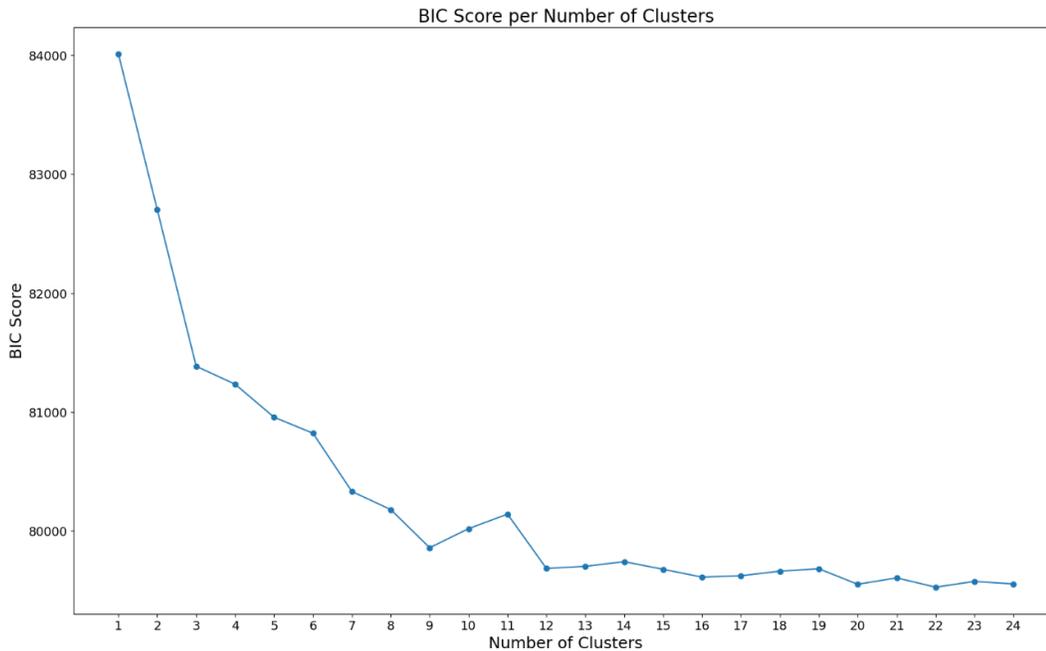


Figura 5. Critério de Informação Bayesiano (BIC) para quantificação de agrupamentos.

Fonte: Autores

Este gráfico demonstra o BIC (eixo Y) em função do número de grupos (eixo X). Valores mais baixos no eixo Y indicam que o número de grupos é mais adequado e valores mais altos indicam que ele é menos adequado para os dados, possivelmente por ser muito simples ou por não capturar adequadamente as características dos dados. O número de grupos escolhido foi 12, pois neste ponto o gráfico começa a se estabilizar, indicando que adicionar mais clusters não melhoraria muito o modelo.

Aqui também, o BIC e o algoritmo de agrupamento requereram a definição de variáveis. No caso do BIC é preciso definir o intervalo de possíveis números de grupos a ser testado (1 a 24) e o parâmetro de *covariance\_type (full)*. No algoritmo de agrupamento, foi preciso fornecer o número de grupos selecionado (12) e o parâmetro de *covariance\_type (full)*.

O *covariance\_type* especifica como a forma e a dispersão de cada grupo é modelado. Se o tipo de covariância utilizado no BIC fosse diferente do usado no algoritmo de agrupamento, a escolha do número ideal de grupos com base no BIC não refletiria corretamente o comportamento esperado pelo agrupamento. Isso levaria inconsistências no processo de agrupamento. Por isso, ambos precisam usar o mesmo tipo de covariância para garantir que o modelo avaliado e o modelo de agrupamento sejam compatíveis.

Com o agrupamento feito, a próxima etapa consiste na elaboração das ferramentas de visualização. As duas ferramentas de visualização da classificação foram vitais não só para a tomada de decisões do processo de Design, mas também para calibrar o próprio processo de classificação. Elas possibilitaram o “diálogo” entre o designer e a máquina para que se chegasse a uma classificação compreensível.

A primeira ferramenta utilizou o programa *Grasshopper* para desenvolver uma interface que selecionasse randomicamente um determinado número de iterações de cada grupo criado e os exibisse através de uma vista isométrica e de quatro vistas ortogonais laterais, iguais às que alimentaram o sistema. A figura 6 abaixo ilustra essa interface. Devido a limitação de espaço, a figura 6 apresenta apenas seis exemplos de instâncias produzidas pelas iterações para doze agrupamentos de sólidos. Ao longo da calibragem do processo de classificação foram selecionadas vinte iterações para serem exibidas.

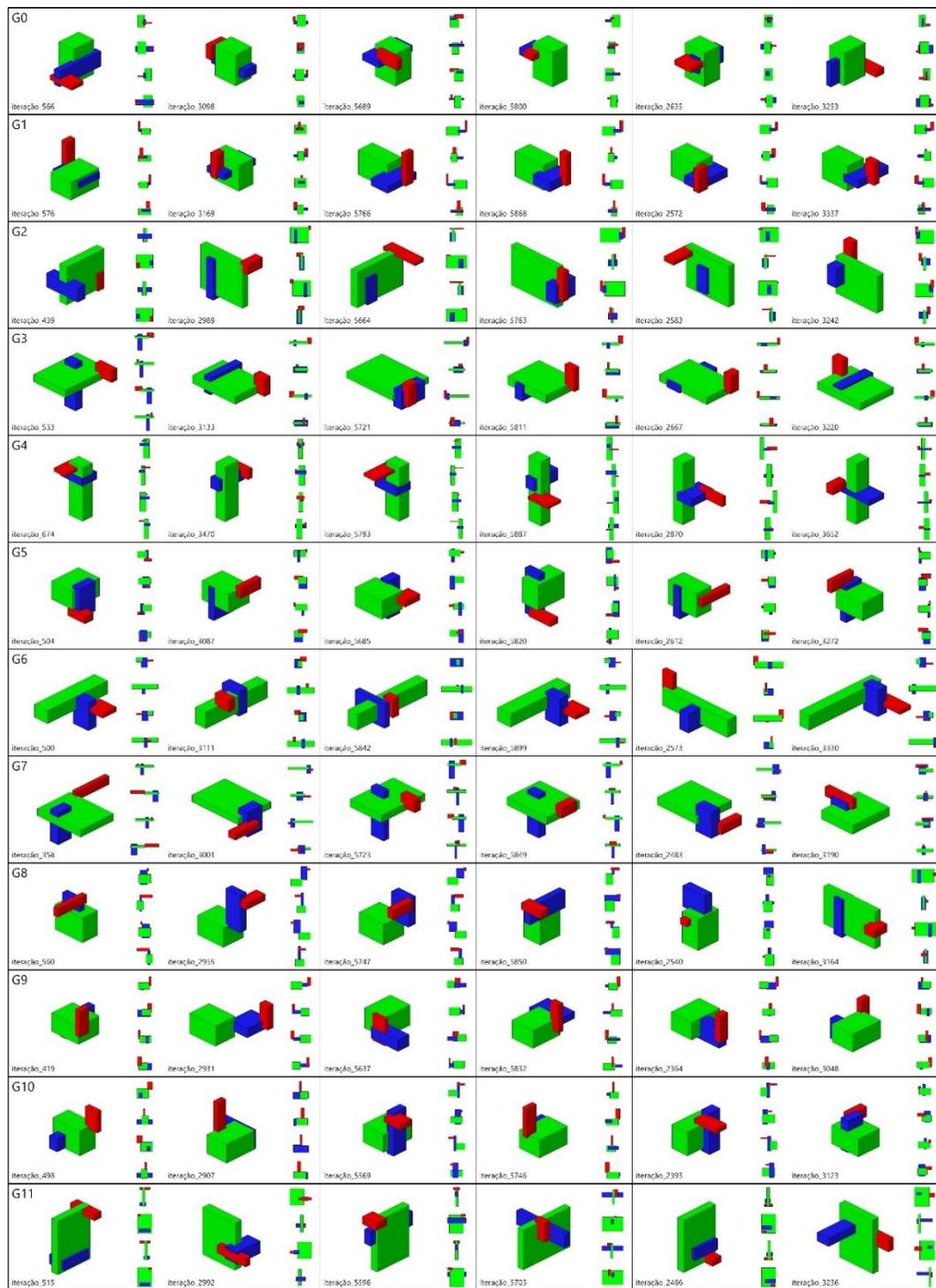


Figura 6. Interface simplificada para visualização das iterações agrupadas.

Fonte: Autores.

A segunda ferramenta de visualização gera uma imagem que possibilita a observação de todos os agrupamentos e seus elementos de uma só vez em um espaço bidimensional. Essa imagem também foi desenvolvida no programa *Grasshopper* e é gerada com a entrada dos dados processados no método de redução de dimensionalidade UMAP, descrito anteriormente. A Figura 7 ilustra a segunda interface com os dados finais do estudo.

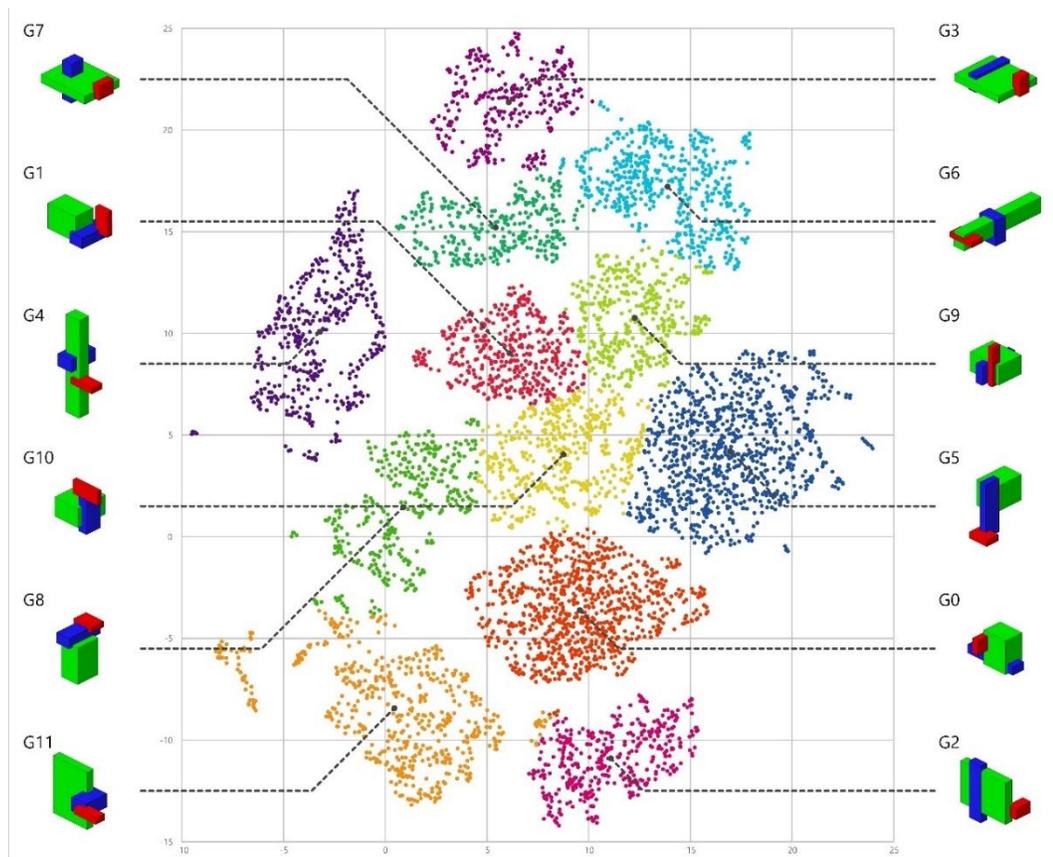


Figura 7. Representação dos agrupamentos em duas dimensões através do método UMAP.

Fonte: Autores

Através dessas interfaces foi possível fazer uma primeira avaliação da precisão das classificações e assim seguir com o processo para a parte de validação dos resultados. O processo seguiu para a etapa final quando não foram verificados mais de 10% (2 em cada grupo de 20) de erros nas classificações de cada grupo observados na primeira ferramenta de visualização mostrada anteriormente e quando a segunda ferramenta mostrou um mapa com uma lógica compreensível para os autores.

## RESULTADOS

O primeiro passo da avaliação consiste em replicar os padrões de agrupamento da máquina observados nas ferramentas de visualização. Para isso os autores analisaram 600 composições selecionadas randomicamente, o que corresponde a 10% do número total de iterações. Com essas análises foi possível comparar a classificação da máquina com a classificação elaborada pelos autores a partir da lógica percebida na classificação da máquina.

Quantitativamente, das 600 classificações selecionadas, a verificação identificou 51 casos em que os autores não identificaram uma lógica por trás da classificação da máquina e classificaram uma composição de maneira diferente que a máquina. Isso representa 8,5% das iterações selecionadas.

De todas as incompatibilidades encontradas, 30 envolveram o grupo 10. Dessas, 22 envolveram classificações do grupo 10 pela máquina e 8 envolveram classificações do grupo 10 pela avaliação humana. Em todos esses 30 casos, o grupo 10 foi confundido com um dos



Os grupos que possuem blocos planares dominantes se diferenciam principalmente pela orientação do bloco dominante, pela orientação dos blocos subdominantes e subordinados e pela distância do bloco dominante até o solo. O grupo 2 possui composições com blocos dominantes horizontais em pé; o grupo 11 possui composições com blocos dominantes na vertical; e os grupos 3 e 7 possuem composições com blocos dominantes na horizontal deitados. O que diferencia esses dois últimos é que o grupo 3 engloba as peças em que existe maior chance do bloco subdominante estar na horizontal e o bloco dominante estar próximo da base ou solo. Já no grupo 7, as peças são tipicamente descoladas do solo com o bloco subdominante azul na orientação vertical.

Os grupos lineares se diferenciam somente pela orientação do bloco dominante. Dessa maneira, o grupo 4 possui composições com blocos dominantes lineares verticais e o grupo 6 engloba aquelas peças com blocos dominantes lineares horizontais.

Diferente do que se observa nos grupos que possuem blocos planares e lineares dominantes, aqueles que possuem blocos volumétricos dominantes apresentam uma lógica de classificação menos evidente. Isso porque, alguns grupos, principalmente o 10, possuem características que outros grupos possuem. Como pode-se observar na Figura 6, à primeira vista o grupo 0 é aquele que engloba as peças dominantes verdes verticais; o grupo 1 engloba as peças horizontais em pé; o grupo 5 se caracteriza por possuir peças subdominantes azuis verticais e peças verdes longe do solo; o grupo 8 engloba as composições com peças azuis no alto; o grupo 9 engloba as peças dominantes horizontais deitadas; e o grupo 10 engloba as composições com peças vermelhas no alto. Apesar dessas impressões preliminares parecerem suficientes para compreender a lógica por trás das classificações, existem alguns casos que elas não bastam.

Nos casos, em que as fronteiras entre grupos estão mais difusas é importante levar em consideração a soma da magnitude de cada aspecto. Por exemplo, apesar do grupo 5 ser fortemente caracterizado por possuir peças azuis verticais que elevam o bloco verde dominante, o grupo 10 também possui uma covariância que inclui essas características. Assim, a altura em que a peça verde está do solo, e a altura da peça vermelha devem ser consideradas. Isso porque, quanto menor for a distância entre a peça verde e o solo e mais alta for o posicionamento da peça vermelha, maior a chance da composição ser do grupo 10.

Em outro exemplo, por mais que o grupo 0 seja composto apenas por peças verdes verticais, o grupo 8 também possui algumas peças verdes nessa orientação. Assim, o desempate deve ficar a cargo dos outros parâmetros. Especificamente, nesse caso, na localização das peças azuis em relação a verde. Isso porque, quanto mais altas essas peças estiverem, maior a chance da composição ser do grupo 8.

Tecnicamente falando, a sobreposição de características nos grupos gerados por modelos de misturas Gaussianas é uma consequência natural da modelagem probabilística flexível que permite capturar a complexidade e as nuances dos dados reais. Esse modelo assume que cada componente possui um conjunto de intensidades de cada parâmetro analisado (PEDREGOSA et al., 2011). Assim, componentes com distribuições semelhantes tenderam a estar no mesmo grupo. Ou seja, a distribuição está influenciando a Conceção, que poderíamos até chamar de caráter da composição, diferentemente do caráter do sólido isoladamente. Cada grupo possui uma média e uma covariância. Ou seja, uma combinação média que represente o elemento mais representativo do grupo e uma covariância que engloba os limites do grupo. Essa abordagem probabilística permite que os pontos de dados tenham uma probabilidade não-nula de pertencer a mais de um cluster, o que pode levar a alguma sobreposição entre os clusters.

## CONCLUSÕES

Tanto do ponto de vista quantitativo quanto do ponto de vista qualitativo, esse trabalho sugere que, mesmo com uma pequena quantidade de dados, gerados pelos próprios autores, é possível desenvolver uma ferramenta de classificação não supervisionada de imagens. Mesmo atingindo um resultado satisfatório, onde o sistema conseguiu criar agrupamentos de maneira lógica, observou-se uma incompatibilidade principalmente em relação aos grupos com peças dominantes de caráter volumétrico e em especial em relação ao grupo 10.

Como sugestões para trabalhos futuros, recomenda-se aumentar o número de agrupamentos. Por ser um estudo inicial, foi definido limitar o número de agrupamentos gerados pela máquina. Entretanto, com mais grupos, o estudo poderá mostrar uma segmentação mais detalhada dos dados, possibilitando a identificação de subgrupos menores ou padrões mais específicos que não foram detectados aqui. Outra sugestão seria explorar abordagens que venham ser capazes de reduzir as incompatibilidades observadas. Além disso, uma possibilidade seria conduzir aplicações desse tipo de estudo experimental em um âmbito mais próximo da prática profissional do Design. Um estudo assim permitiria avaliar como os métodos de classificação e agrupamento podem contribuir para a criação de soluções mais personalizadas, otimizar processos de design, e identificar tendências estéticas ou funcionais com maior precisão no desenvolvimento de produtos. Por último, outro caminho seria investigar explorações de variações metodológicas pela inclusão de outros métodos de análise de iterações junto ao apresentado aqui, como avaliações através de processos de regressão que simulem a avaliação humana a respeito de diferentes critérios.

Do ponto de vista prático, o artigo contribui diretamente para a solução de um problema comum em processos de design generativo: a necessidade de explorar e organizar vastos conjuntos de iterações sem uma programação direta e onerosa do designer. A metodologia apresentada demonstra como otimizar esse processo, permitindo que designers explorem soluções de forma mais eficiente. A implementação prática da metodologia, utilizando ferramentas como *Grasshopper* e *Google Colab*, destaca a aplicabilidade real dessa abordagem, facilitando a adoção por profissionais e pesquisadores do Design Computacional.

Outra contribuição está na forma de avaliação dos resultados gerados pela classificação não supervisionada. Ao empregar métodos de validação manual externa, além das análises através das ferramentas de visualização. Esse tipo de avaliação garante uma análise mais robusta e criteriosa dos resultados, permitindo não apenas identificar possíveis inconsistências nos agrupamentos, mas também oferecer uma camada adicional de interpretação e refinamento dos dados.

No campo teórico, o estudo amplia o uso de técnicas de classificação não supervisionada no design, um campo onde essas técnicas ainda são pouco exploradas. A pesquisa oferece uma contribuição significativa ao demonstrar como características morfológicas e espaciais das iterações podem ser analisadas e organizadas automaticamente por algoritmos de inteligência artificial. A escolha do modelo de *autoencoder*, em conjunto com técnicas de redução de dimensionalidade e do algoritmo de agrupamento cria um framework metodológico que pode ser replicado e adaptado para diferentes contextos de design.

O estudo também contribui ao reforçar a investigação por meio de um processo iterativo entre o método e os resultados. Essa abordagem iterativa permite que o artefato, representado pela estrutura metodológica, seja constantemente avaliado e refinado com base no desempenho observado em cada ciclo de análise. Ao longo das iterações, o método é ajustado em resposta aos resultados intermediários, garantindo que a solução se torne cada vez mais eficaz na organização das iterações do sistema generativo. Esse processo contínuo de retroalimentação

não apenas aprimora o artefato em termos práticos, mas também gera novos insights teóricos sobre como características morfológicas e espaciais podem ser melhor exploradas e interpretadas por técnicas de aprendizado profundo. Dessa forma, a interação entre método e resultado, ou entre o designer pesquisador e a máquina, torna-se uma via de mão dupla, onde ambos se influenciam mutuamente, gerando avanços tanto na solução prática do problema quanto no desenvolvimento de conhecimento científico no campo do design computacional.

## **Referências Bibliográficas**

ABADI, Martín et al. TensorFlow: A system for large-scale machine learning. *Em: PROCEEDINGS OF THE 12TH USENIX SYMPOSIUM ON OPERATING SYSTEMS DESIGN AND IMPLEMENTATION, OSDI 2016 2016*, **Anais [...]**. : USENIX Association, 2016. p. 265–283. Disponível em: <https://arxiv.org/abs/1605.08695v2>. Acesso em: 10 out. 2024.

ABUZURAIQ, Ahmed M.; ERHAN, Halil. The many faces of similarity: A visual analytics approach for design space simplification. *Em: RE: ANTHROPOCENE, DESIGN IN THE AGE OF HUMANS - PROCEEDINGS OF THE 25TH INTERNATIONAL CONFERENCE ON COMPUTER-AIDED ARCHITECTURAL DESIGN RESEARCH IN ASIA, CAADRIA 2020 2020*, **Anais [...]**. : The Association for Computer-Aided Architectural Design Research in Asia (CAADRIA), 2020. p. 485–494. DOI: 10.52842/CONF.CAADRIA.2020.1.485. Acesso em: 31 mar. 2024.

AHLQUIST, Sean. Reciprocal Relationships of Materiality and Human Engagement. *Em: Paradigms of Performativity in Design and Architecture*. New York: Routledge, 2020.

BHATT, Mehul; SCHULTZ, Carl; HUANG, Minqian. The shape of empty space: Human-centred cognitive foundations in computing for spatial design. *Em: PROCEEDINGS OF IEEE SYMPOSIUM ON VISUAL LANGUAGES AND HUMAN-CENTRIC COMPUTING, VL/HCC 2012*, **Anais [...]**. [s.l.: s.n.] p. 33–40. DOI: 10.1109/VLHCC.2012.6344477. Acesso em: 31 mar. 2024.

BROWN, Nathan;; MUELLER, Caitlin. Designing with data: Moving beyond the design space catalog. *Em: ACADIA 2017 DISCIPLINES & DISRUPTION 2017*, **Anais [...]**. [s.l.: s.n.] p. 154–163.

CHEN, Haoyu; LI, Wenbo; GU, Jinjin; REN, Jingjing; SUN, Haoze; ZOU, Xueyi; ZHANG, Zhensong; YAN, Youliang; ZHU, Lei. Low-Res Leads the Way: Improving Generalization for Super-Resolution by Self-Supervised Learning. *Em: PROCEEDINGS OF THE IEEE/CVF CONFERENCE ON COMPUTER VISION AND PATTERN RECOGNITION (CVPR) 2024*, **Anais [...]**. [s.l.: s.n.] p. 25857–25867. Disponível em: <https://arxiv.org/abs/2403.02601v1>. Acesso em: 9 out. 2024.

CHOLLET, François. **Keras: The Python Deep Learning Library**. 2015. Disponível em: <https://keras.io/>. Acesso em: 10 out. 2024.

FAZI, M. Beatrice. Beyond Human: Deep Learning, Explainability and Representation. **Algorithmic Thought**, [S. l.], v. 38, 2020. DOI: 10.1177/0263276420966386. Acesso em: 6 out. 2021.

FISCHER, Thomas; HERR, Christiane M. Teaching Generative Design. **Proceedings of the 4th International Conference on Generative Art**, [S. l.], 2001. Disponível em: [https://www.researchgate.net/publication/30869860\\_Teaching\\_Generative\\_Design](https://www.researchgate.net/publication/30869860_Teaching_Generative_Design). Acesso em: 21 out. 2021.

GOODFELLOW, Ian; POUGET-ABADIE; MIRZA, Mehdi; XU, Bing; WARDE-FARLEY, David; OZAIR, Sherjil; COURVILLE, Aaron; BENGIO, Yoshua. Generative Adversarial Nets. **Proceedings of the International Conference on Neural Information Processing Systems**, [S. l.], 2014. Disponível em:

<https://proceedings.neurips.cc/paper/2014/hash/5ca3e9b122f61f8f06494c97b1afccf3-Abstract.html>. Acesso em: 3 mar. 2022.

HANNAH, Gail Greet. **Elements of Design**. New York: Princeton Architecture Press, 2002.

HUANG, Jeffrey; JOHANES, Mikhael; KIM, Frederick Chando; DOUMPIOTI, Christina; HOLZ, Georg Christoph. On GANs, NLP and Architecture: Combining Human and Machine Intelligences for the Generation and Evaluation of Meaningful Designs. **Technology|Architecture + Design**, [S. l.], v. 5, n. 2, p. 207–224, 2021. DOI: 10.1080/24751448.2021.1967060. Disponível em: <https://www.tandfonline.com/doi/abs/10.1080/24751448.2021.1967060>. Acesso em: 7 out. 2024.

JANNANI, Oussama; IDRISSEI, Najlae; CHAKIB, Houda. An Image Compression Approach Based on Convolutional AutoEncoder. **Lecture Notes in Networks and Systems**, [S. l.], v. 806 LNNS, p. 78–91, 2023. DOI: 10.1007/978-3-031-46584-0\_7. Disponível em: [https://link.springer.com/chapter/10.1007/978-3-031-46584-0\\_7](https://link.springer.com/chapter/10.1007/978-3-031-46584-0_7). Acesso em: 7 out. 2024.

JOHANES, Mikhael; HUANG, Jeffrey. Deep Learning Isovist: Unsupervised Spatial Encoding in Architecture. *Em*: PAPER PRESENTED AT 2021 ASSOCIATION FOR COMPUTER AIDED DESIGN IN ARCHITECTURE ANNUAL CONFERENCE, ACADIA 2021, VIRTUAL, ONLINE. 2021, **Anais [...]**. [s.l.: s.n.] Disponível em: <https://scholar.ui.ac.id/en/publications/deep-learning-isovist-unsupervised-spatial-encoding-in-architectu>. Acesso em: 31 mar. 2024.

LEACH, Neil. **Architecture in the age of artificial intelligence : an introduction to AI for architects**. London: Bloomsbury, 2022. . Acesso em: 3 mar. 2022.

LI, Shuang; CORNEY, Jonathan. Multi-view expressive graph neural networks for 3D CAD model classification. **Computers in Industry**, [S. l.], v. 151, p. 103993, 2023. DOI: 10.1016/J.COMPIND.2023.103993. Acesso em: 31 mar. 2024.

MCINNES, Leland; HEALY, John; MELVILLE, James. UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. **Journal of Open Source Software**, [S. l.], 2018. DOI: 10.21105/joss.00861. Disponível em: <https://arxiv.org/abs/1802.03426v3>. Acesso em: 7 out. 2024.

MORDVINTSEV, Alexander; OLAH, Christopher; TYKA, Mike. **Inceptionism: Going Deeper into Neural Networks**. 2015. Disponível em: <https://research.google/blog/inceptionism-going-deeper-into-neural-networks/>. Acesso em: 7 out. 2024.

NATIVIDADE, Verônica Gomes. **Fraturas metodológicas nas arquiteturas digitais**. 2010. UNIVERSIDADE DE SÃO PAULO , São Paulo, 2010. Acesso em: 2 maio. 2020.

PEDREGOSA, Fabian et al. Scikit-learn: Machine Learning in Python. **Journal of Machine Learning Research**, [S. l.], v. 12, n. 85, p. 2825–2830, 2011. Disponível em: <http://jmlr.org/papers/v12/pedregosa11a.html>. Acesso em: 10 out. 2024.

PEFFERS, Ken; TUUNANEN, Tuure; ROTHENBERGER, Marcus A.; CHATTERJEE, Samir. A design science research methodology for information systems research. **Journal of Management Information Systems**, [S. l.], v. 24, n. 3, p. 45–77, 2007. DOI: 10.2753/MIS0742-1222240302. Acesso em: 7 out. 2024.

**Rowena Reed Kostellow Fund - 3-D Design Education Resource**. [s.d.]. Disponível em: <http://rowenafund.org/>. Acesso em: 21 nov. 2024.

SCHMARJE, Lars; SANTAROSSA, Monty; SCHRÖDER, Simon Martin; KOCH, Reinhard. A Survey on Semi-, Self- and Unsupervised Learning for Image Classification. **IEEE Access**, [S. l.], v. 9, p. 82146–82168, 2021. DOI: 10.1109/ACCESS.2021.3084358. Acesso em: 7 out. 2024.

SCHWARZ, Gideon. “Estimating the Dimension of a Model.” **The Annals of Statistics**, [S. l.], v. 6, n. 2, p. 461–464, 1978. DOI: 10.2307/2958889. Disponível em: <http://www.citeulike.org/user/abhishekseth/article/90008>. Acesso em: 15 out. 2024.

VAN AKEN, Joan Ernst. Management research as a design science: Articulating the research products of mode 2 knowledge production in management. **British Journal of Management**, [S. l.], v. 16, n. 1, p. 19–36, 2005. DOI: 10.1111/J.1467-8551.2005.00437.X. Acesso em: 7 out. 2024.

XU, Rui; WUNSCH, Donald. Survey of clustering algorithms. **IEEE Transactions on Neural Networks**, [S. l.], v. 16, n. 3, p. 645–678, 2005. DOI: 10.1109/TNN.2005.845141. Disponível em: [https://www.researchgate.net/publication/3303538\\_Survey\\_of\\_Clustering\\_Algorithms](https://www.researchgate.net/publication/3303538_Survey_of_Clustering_Algorithms). Acesso em: 7 out. 2024.

ZHANG, Ji; TAN, Leonard; TAO, Xiaohui; PHAM, Thuan; CHEN, Bing. Relational intelligence recognition in online social networks — A survey. **Computer Science Review**, [S. l.], v. 35, p. 100221, 2020. DOI: 10.1016/J.COSREV.2019.100221. Acesso em: 9 out. 2024.

Daniel R. A. B. Vianna  
[danielrabv@gmail.com](mailto:danielrabv@gmail.com)

Claudio F. de Magalhães  
[claudio-design@puc-rio.br](mailto:claudio-design@puc-rio.br)

Érico F. Mineiro  
[erico.acad@gmail.com](mailto:erico.acad@gmail.com)