

# Geração de Imagens Aleatórias por meio de Algoritmos Computacionais

## Generation of Random Images Through Computational

Juliano Bitencourt Padilha<sup>1</sup> , Lisandra Kittel Ries<sup>2</sup> , Anésio Felipe Zeitune<sup>3</sup> ,  
Ricardo Luiz Alves<sup>4</sup> 

1. Doutor em Engenharia  
Instituto Federal de Educação, Ciência e  
Tecnologia de Santa Catarina (IFSC)  
E-mail: juliano.padilha@ifsc.edu.br

2. Doutora em Engenharia  
Instituto Federal de Educação, Ciência e  
Tecnologia de Santa Catarina (IFSC)  
E-mail: lisandra.ries@ifsc.edu.br

3. Doutor em Engenharia  
Instituto Federal de Educação, Ciência e  
Tecnologia de Santa Catarina (IFSC)  
E-mail: anesio.felipe@ifsc.edu.br

4. Doutor em Engenharia  
Instituto Federal de Educação, Ciência e  
Tecnologia de Santa Catarina (IFSC)  
E-mail: ricardoalves@ifsc.edu.br

### Artigo Original

**Resumo:** Este trabalho explora a geração de imagens aleatórias por meio de algoritmos computacionais. A capacidade de gerar imagens aleatórias é um recurso valioso em campos diversos como arte digital, segurança de dados e reconhecimento de padrões, com implicações que vão desde a expressão criativa até a segurança cibernética. Nesse contexto, este trabalho descreve minuciosamente as etapas envolvidas na criação de imagens aleatórias, detalhando o funcionamento de três algoritmos desenvolvidos especificamente para esse fim. Esses algoritmos incorporam decisões aleatórias em diferentes estágios do processo de geração, proporcionando uma ampla gama de resultados únicos e imprevisíveis. Por fim, os resultados obtidos a partir da execução desses algoritmos são apresentados. Em suma, este estudo exemplifica a capacidade de geração de imagens aleatórias por meio da computação.

**Palavras-chave:** Processo de geração de imagens; Algoritmos de decisão aleatória; Aplicações de imagens aleatórias.

**Abstract:** This work explores the generation of random images through computational algorithms. The ability to generate random images is a valuable resource in various fields such as digital art, data security, and pattern recognition, with implications ranging from creative expression to cybersecurity. In this context, this work meticulously describes the steps involved in creating random images, detailing the operation of three algorithms developed specifically for this purpose. These algorithms incorporate random decisions at different stages of the generation process, providing a wide range of unique and unpredictable results. Finally, the results obtained from the execution of these algorithms are presented. In summary, this study exemplifies the capability of generating random images through computation.

**Keywords:** Image generation process; Random decision algorithms; Random image applications.

## Introdução

A representação de desenhos é uma atividade que acompanha o homem desde os tempos do homem primitivo que habitava as cavernas (Grobel, M. C. B. *et al.*, 2014), (Pereira, T. *et al.*, 2014). Ao decorrer da evolução humana, os desenhos passaram a ser uma ferramenta poderosa para o desenvolvimento social. Exemplos práticos são a linguagem escrita, representações cartográficas, desenhos industriais (projetos) e fotografias (Valle, M. *et al.*, 2009). Nesse cenário, também podemos citar os grandes pintores que registraram em desenhos uma arte atemporal (Rahe, N., 2022).

Nesse contexto, este trabalho busca explorar uma forma de arte única e modesta por meio do uso de algoritmos computacionais para a geração de imagens. Em um mundo onde a criatividade e a tecnologia estão interligadas, este estudo se concentra na criação de imagens que, embora modestas, têm o poder de inspirar e encantar. É importante notar que os algoritmos que empregados têm a capacidade de gerar imagens distintas a cada execução, consequência da introdução da aleatoriedade nos códigos. Esse fato nos faz refletir sobre como a aleatoriedade pode dar origem ao inimaginável.

## Revisão da Literatura

Com o crescente uso de técnicas de inteligência artificial, a geração de imagens por meio de algoritmos computacionais vem alcançando destaque nos dias atuais. Como exemplo, tem-se a ferramenta *Bing Image Creator* da empresa Microsoft, que é capaz de gerar imagens a partir de descrições textuais fornecidas pelo usuário.

Assim, para situar esse trabalho em um contexto mais amplo, essa seção de revisão da literatura descreve algumas publicações relacionadas a

algoritmos computacionais para a geração de imagens, buscando contextualizar a importância de nossa abordagem na evolução contínua desse campo de pesquisa.

Gatys *et al.* (2015) descrevem um algoritmo que combina o conteúdo de uma imagem com o estilo de outra por meio de redes neurais convolucionais. Destacam que técnica empregada permite a transferência de estilo de pinturas famosas para fotos, criando imagens artísticas únicas.

Em outro trabalho (Gatys, L. A. *et al.*, 2015), os autores apresentam um novo modelo de texturas naturais baseado em redes neurais convolucionais otimizadas para reconhecimento de objetos. O modelo gera texturas de alta qualidade e destaca como as representações de texturas se desenvolvem nas camadas, tornando informações sobre objetos mais claras. Os autores destacam que o modelo pode ser aplicado em pesquisa de neurociência e pode oferecer *insights* sobre as representações de redes neurais convolucionais.

Elgammal *et al.* (2017) propõem um sistema de geração de arte que aprende estilos a partir de obras de arte e se torna criativo ao desviar desses estilos. Usando redes renerativas adversariais modificadas, o sistema gera arte criativa que os humanos não conseguem distinguir da produzida por artistas contemporâneos, evidenciando sua capacidade de criação artística.

Radford *et al.* (2016) abordam o desequilíbrio entre o sucesso das redes convolucionais em aprendizado supervisionado e não supervisionado. Os autores introduzem redes generativas adversariais convolucionais profundas, que se mostram promissoras no aprendizado não supervisionado. No trabalho, as técnicas apresentadas são aplicadas na geração de imagens.

Dosovitskiy & Brox (2016) destacam que modelos de aprendizado de máquina geradores de imagens são geralmente treinados com funções de perda baseadas na distância no espaço da imagem, o que frequentemente resulta em resultados excessivamente suavizados. Assim, os autores propõem uma classe de funções de perda, denominada de métricas de similaridade perceptual profunda, que atenuam esse problema.

Mordvintsev *et al.* (2015) apresentam uma ferramenta de visualização projetada para ajudar a entender como as redes neurais funcionam e o que cada camada aprendeu. Além de obter algumas informações sobre como essas redes executam tarefas de classificação, os autores destacam como esse processo também gerou algumas belas obras de arte.

## **Método**

O desenvolvimento de algoritmos para geração de imagens aleatórias é um processo que requer criatividade humana. Assim, o objetivo deste artigo é exemplificar esse processo a partir de três algoritmos desenvolvidos. Não há regras pré-definidas que necessitam ser seguidas na elaboração de um algoritmo com essa finalidade, apenas deve ser assegurado que ao longo do código se faça uso da aleatoriedade em uma ou mais etapas. Também não há dados a serem inseridos pelo usuário na execução dos algoritmos, embora seja plenamente possível implementar solicitações ao usuário, como por exemplo, solicitar quais cores o usuário deseja escolher para a geração das imagens.

As descrições dos algoritmos desenvolvidos são apresentadas na próxima seção deste trabalho. Os resultados de todos os algoritmos são imagens que se diferem sempre que os códigos são executados.

Os algoritmos desenvolvidos foram implementados em linguagem Matlab (outras linguagens de programação também podem ser utilizadas), cabendo destacar que a mesma apresenta funções nativas para o uso da aleatoriedade (Chapman, S. J., 2016). Em relação ao hardware utilizado, constitui-se de um Core i7-6700QH com 2,60 GHz e 16,0 GB de memória RAM. Já o tempo de execução dos códigos são da ordem de segundos e, portanto, não se trata de uma variável que requeira preocupação em termos de esforço computacional.

## Resultados

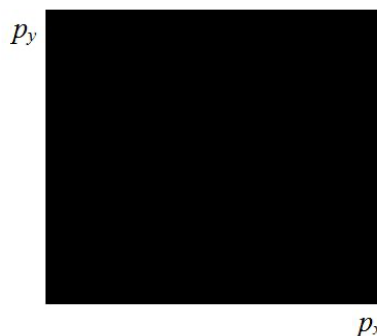
Nesta seção serão apresentadas as etapas de cada um dos algoritmos elaborados. A descrição detalhada das etapas visa a exemplificar a criatividade e pensamento humano, servindo de base para o leitor que deseja elaborar códigos para a geração de imagens aleatórias.

### Algoritmo 1

#### a) Geração de um retângulo

Inicialmente é gerado um retângulo na cor preta com as dimensões  $p_x$  por  $p_y$  (ver Figura 1), onde  $p_x$  e  $p_y$  são, respectivamente, o total de pixels nos eixos das abscissas e coordenadas.

**Figura 1.** Retângulo com  $p_x$  por  $p_y$  pixels.

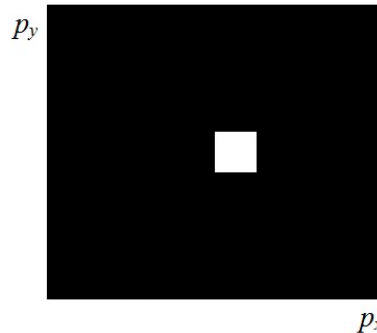


Fonte: Elaborado pelo autores (2023).

b) Início da geração da imagem aleatória

O ponto de partida para a geração da imagem aleatória é um pixel localizado ao centro do retângulo apresentado na Figura 1. A cor branca será atribuída a esse pixel central (ver Figura 2).

**Figura 2.** Ponto de partida localizado ao centro do retângulo.

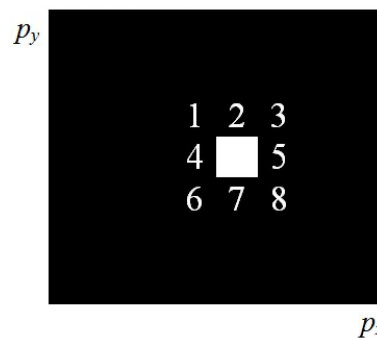


Fonte: Elaborado pelos autores (2023).

c) Definindo os demais pixels que serão brancos

Excluindo os pixels das bordas, ao redor de cada pixel há outros 8 pixels adjacentes (ver Figura 3). Por meio de uma função, de forma aleatória é selecionado um número entre 1 e 8, de modo que o próximo pixel a assumir a cor branca está relacionado ao número selecionado. E o novo pixel na cor branca também apresenta 8 pixels adjacentes, e assim, o processo de seleção aleatória se repete até que seja ultrapassado o limite de uma das 4 bordas do retângulo (ver Figura 4).

**Figura 3.** Seleção do próximo pixel a assumir a cor branca.



Fonte: Elaborado pelos autores (2023).

**Figura 4.** Fim do processo de definição dos pixels brancos. No exemplo, os pixels com números 1, 4 e 6 estão além de uma borda, portanto, se alguns desses números for o selecionado, a etapa aqui descrita é encerrada e a imagem final é mostrada.

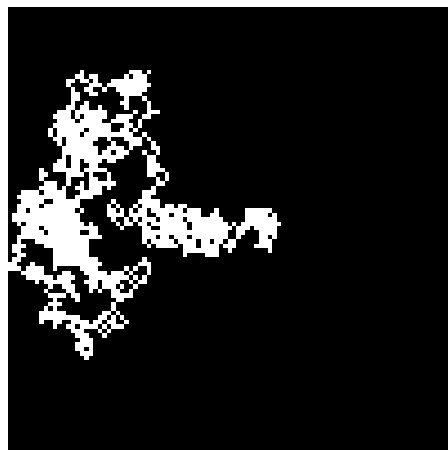


Fonte: Elaborado pelos autores (2023).

#### d) Imagem final gerada

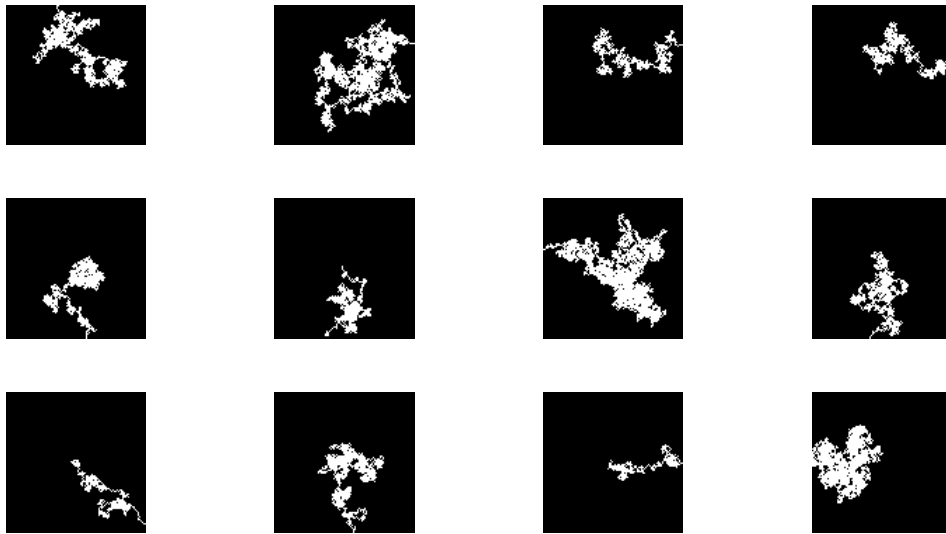
Nesta última etapa, é mostrado ao usuário a imagem gerada. Cada vez que o programa é executado, uma imagem diferente é gerada. A quantidade de imagens possíveis depende de  $p_x$  e  $p_y$ . Na Figura 5 é apresentado um dos resultados obtidos. Já na Figura 6 são apresentados outros exemplos, em menor escala, de imagens obtidas.

**Figura 5.** Exemplo de imagem obtida com o algoritmo 1.



Fonte: Elaborado pelos autores (2023).

**Figura 6.** Outros exemplos de imagens obtidas com o algoritmo 1.



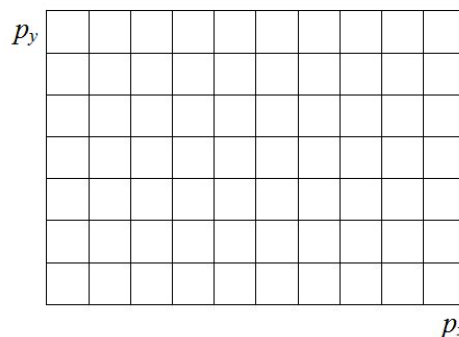
Fonte: Elaborado pelos autores (2023).

## Algoritmo 2

### a) Geração de um retângulo

Inicialmente é gerado um retângulo na cor branca com as dimensões  $p_x$  por  $p_y$ , onde  $p_x$  e  $p_y$  são, respectivamente, o total de pixels nos eixos das abscissas e coordenadas. Ainda, esse retângulo é dividido em quadrados menores que apresentam uma certa quantidade de pixels para a dimensão do lado (ver Figura 7).

**Figura 7.** Retângulo com  $p_x$  por  $p_y$  pixels subdividido em quadrados.



Fonte: Elaborado pelos autores (2023).



b) Preenchendo cada um dos quadrados

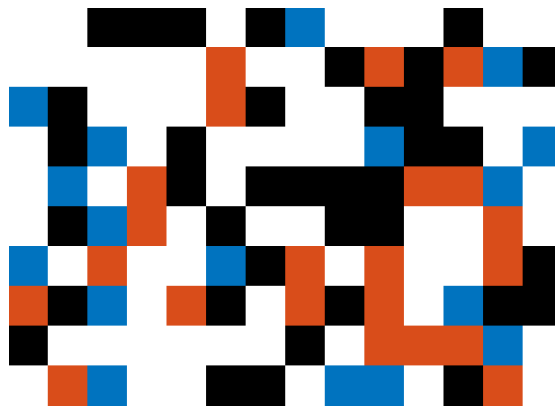
Nesta etapa, para o primeiro quadrado presente na Figura 7, determina-se se o mesmo será ou não preenchido (se não, permanecerá na cor branca). Essa escolha é determinada de forma aleatória. Se opção for sim, determina-se de qual cor será o preenchimento dentre três possíveis. Essa escolha também é determinada de forma aleatória.

c) Imagem final gerada

Nesta última etapa, é mostrado ao usuário a imagem gerada. Cada vez que o programa é executado, uma imagem diferente é gerada.

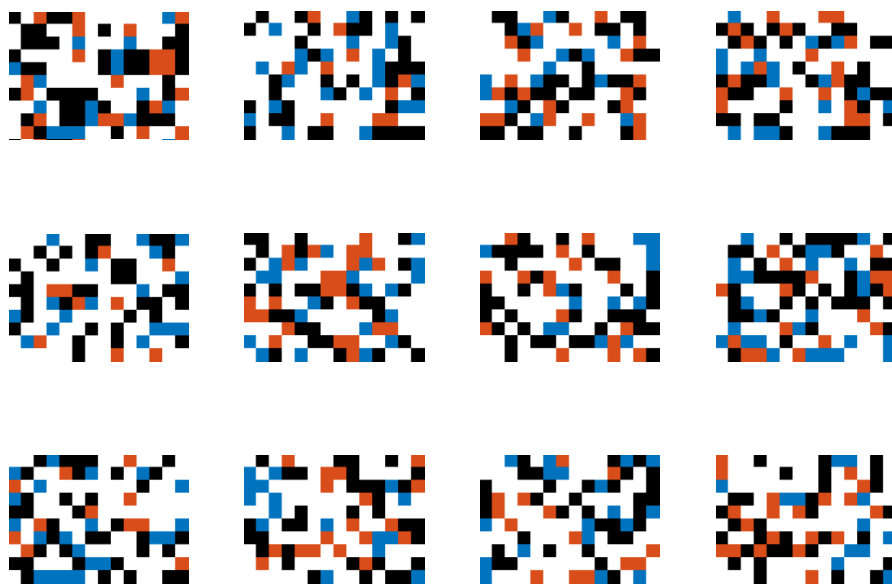
A Figura 8 apresenta um dos resultados obtidos com o algoritmo 2. Já na Figura 9 são apresentados outros exemplos, em menor escala, de imagens obtidas.

**Figura 8.** Exemplo de imagem obtida com o algoritmo 2.



Fonte: Elaborado pelos autores (2023).

**Figura 9.** Outros exemplos de imagens obtidas com o algoritmo 2.



Fonte: Elaborado pelos autores (2023).

### **Algoritmo 3**

#### a) Geração de um retângulo

Assim como no algoritmo 2, inicialmente um retângulo é definido.

#### b) Quantidade de retas

Nesta etapa, é definida uma quantidade de retas que serão desenhadas. Essa quantidade será determinada de forma aleatória, mas deverá estar dentro de um intervalo pré-definido (assumido entre 30 e 70).

#### c) Pontos, espessura e cor de uma reta

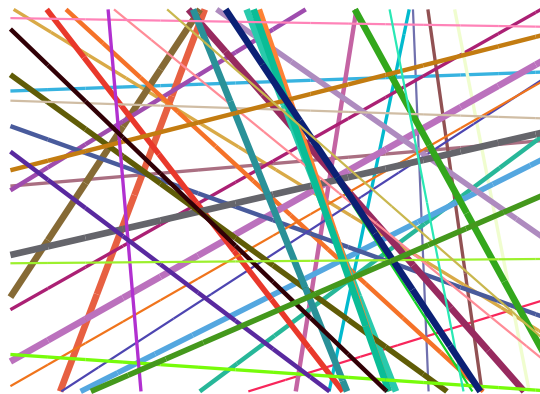
Dois pontos aleatórios são definidos, pelos quais passará a reta, dentro do retângulo. Após, também de forma aleatória, são determinadas a espessura (dentre 5 possíveis) e a cor da reta. Com todos os dados definidos para a reta, a mesma é traçada e o processo se repete até que todas as retas sejam traçadas.

d) Imagem final gerada

Nesta última etapa, é mostrada ao usuário a imagem gerada. Cada vez que o programa é executado, uma imagem diferente é gerada.

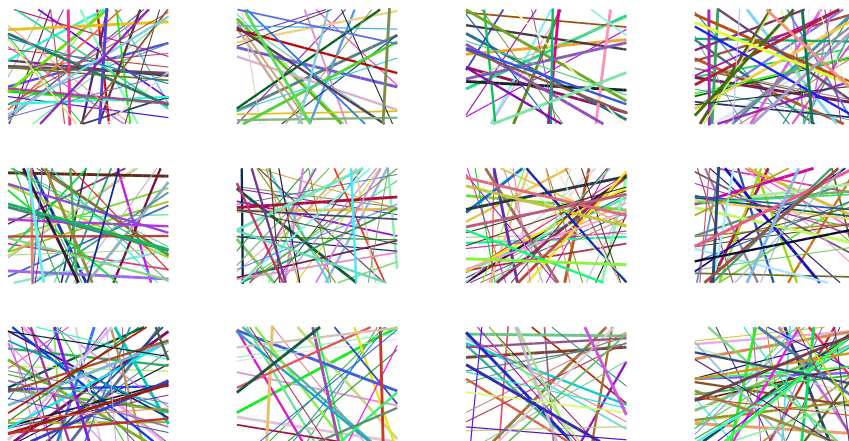
A Figura 10 apresenta um dos resultados obtidos com o algoritmo 3. Já na Figura 11 são apresentados outros exemplos, em menor escala, de imagens obtidas.

**Figura 10.** Exemplo de imagem obtida com o algoritmo 2.



Fonte: Elaborado pelos autores (2023).

**Figura 11.** Outros exemplos de imagens obtidas com o algoritmo 2.



Fonte: Elaborado pelos autores (2023).

## Considerações Finais

Neste artigo foram apresentados três algoritmos computacionais para a geração de imagens aleatórias. Ao detalhar os processos de geração passo a passo, o trabalho fornece insights sobre o potencial desses algoritmos.

Um dos aspectos fundamentais a ser enfatizado é a versatilidade inerente à utilização da aleatoriedade em algum estágio do processo de geração das imagens. Os resultados aqui apresentados representam apenas um vislumbre do vasto espectro de possibilidades que essa abordagem pode oferecer. Cada execução desses algoritmos resulta em imagens únicas e diversas, evidenciando o poder criativo da aleatoriedade na geração de conteúdo visual.

É importante ressaltar que as imagens geradas pelos algoritmos 1 e 3 são tão diversas que é desafiador estimar o número total de imagens distintas que podem ser produzidas. Já para o caso do algoritmo 2, para uma imagem contendo 140 quadrados (conforme os resultados apresentados), tal número é da ordem de  $2^{280}$  (algo em torno de  $10^{84}$ ). Esse número supera significativamente o estimado para o total de átomos no universo observável, que é da ordem de  $10^{80}$ .

Assim, este trabalho não apenas demonstra a viabilidade e eficácia desses algoritmos, mas também ressalta sua relevância no contexto mais amplo da criatividade na geração de imagens. A aplicação prática desses algoritmos pode ser encontrada em campos diversos, desde a produção de arte digital até a resolução de problemas complexos em áreas como reconhecimento de padrões e visão computacional.

Em suma, esta pesquisa abre novas perspectivas para a exploração da aleatoriedade como uma ferramenta criativa na geração de imagens. À

medida que avançamos, é fundamental considerar como esses algoritmos podem ser aplicados em diversos contextos, inspirando futuros trabalhos de pesquisa e aplicações práticas que se beneficiem do tema abordado.

## Referências

CHAPMAN, S. J. **Programação em Matlab para Engenheiros**. Tradução da 5ª edição norte-americana. Tradução: Noveritis do Brasil. Revisão técnica: Flávio Soares Corrêa da Silva. São Paulo, SP: Cengage Learning, 2016.

DOSOVITSKIY, A.; BROX, T., **Generating Images with Perceptual Similarity Metrics based on Deep Networks**. 2016. Disponível em: <<https://doi.org/10.48550/arXiv.1602.02644>>. Acesso em: 01 nov. 2023.

ELGAMMAL, A. *et al.*, **CAN: Creative Adversarial Networks, Generating "Art" by Learning About Styles and Deviating from Style Norms**. 2017. Disponível em: <<https://doi.org/10.48550/arXiv.1706.07068>>. Acesso em: 31 out. 2023.

GATYS, L. A. *et al.*, **A Neural Algorithm of Artistic Style**. 2015. Disponível em: <<https://doi.org/10.48550/arXiv.1508.06576>>. Acesso em: 31 out. 2023.

GATYS, L. A. *et al.*, **Texture Synthesis Using Convolutional Neural Networks**. 2015. Disponível em: <<https://doi.org/10.48550/arXiv.1505.07376>>. Acesso em: 31 out. 2023.

GROBEL, M. C. B. *et al.*, **Da comunicação visual pré-histórica ao desenvolvimento da linguagem escrita, e, a evolução da autenticidade documentoscópica**. Revista Acadêmica Oswaldo Cruz, ano 1, n. 1, 2014.

MORDVINTSEV A., **DeepDream - a code example for visualizing Neural Networks**. 2015. Disponível em: <<https://blog.research.google/2015/07/deepdream-code-example-for-visualizing.html?m=1>>. Acesso em: 01 nov. 2023.

PEREIRA, T. *et al.*, **Um bestiário pré-histórico? A pré-história através das pinturas rupestres**. Revista de História da Arte e Arqueologia, n. 21: 28-51, 2014.

RADFORD, A. *et al.*, **Unsupervised Representation Learning with Deep Convolutional Generative Adversarial Networks**. 2016. Disponível em: <<https://doi.org/10.48550/arXiv.1511.06434>>. Acesso em: 01 nov. 2023.

RAHE, N., **Coleção Folha Grandes Pintores reúne obras essenciais da arte mundial**. Jornal Folha de São Paulo, 6 mai. 2022.

VALLE, M. *et al.*, **A transformação do mundo pela escrita**. Revista ComCiência, n. 113, 2009.

---

**Revista Interdisciplinar Encontro das Ciências – ISSN: 2595-0959, V. 7, N. 1, 2024**

**Conflito de interesses**

Os autores declaram não haver conflito de interesses.

**Contribuição dos autores**

Concepção e conceitualização: JBP, LKR, AFZ, RLA

Redação do manuscrito original: JBP, LKR, AFZ, RLA

Curadoria de dados: JBP, LKR, AFZ, RLA

Análise de dados: JBP, LKR, AFZ, RLA

Redação textual: JBP, LKR, AFZ, RLA

Supervisão: JBP, LKR, AFZ, RLA

**Financiamento**

Não houve financiamento.

**Consentimento de uso de imagem**

Não se aplica.

**Aprovação, ética e consentimento**

Não se aplica.

---