



UNIVERSIDADE FEDERAL DO CEARÁ  
CAMPUS RUSSAS  
CURSO DE GRADUAÇÃO EM ENGENHARIA DE SOFTWARE

JOÃO DENILSON NERY SOUSA

**DESENVOLVIMENTO DE UM SOFTWARE PARA AUXILIAR NA CONFECÇÃO  
DE *CIRCLE STRING ART***

Russas

2021

JOÃO DENILSON NERY SOUSA

DESENVOLVIMENTO DE UM SOFTWARE PARA AUXILIAR NA  
CONFECÇÃO DE *CIRCLE STRING ART*

Trabalho de Conclusão de Curso  
apresentado ao Curso de Graduação  
em Engenharia de Software do  
Campus Russas da Universidade  
Federal do Ceará, como requisito  
parcial à obtenção do grau de  
bacharel em Engenharia de Software,

Orientador: Prof. Me. Daniel Márcio  
Batista de Siqueira

Russas

2021

Dados Internacionais de Catalogação na Publicação  
Universidade Federal do Ceará  
Biblioteca Universitária  
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

---

S697d Sousa, João Denilson Nery.  
Desenvolvimento de um software para auxiliar na confecção de Circle string art / João Denilson Nery  
Sousa. – 2021.  
35 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Campus de Russas,  
Curso de Engenharia de Software, Russas, 2021.

Orientação: Prof. Me. Daniel Márcio Batista de Siqueira.

1. String Art. 2. Algoritmo. 3. Circle string art. 4. Arte computacional. I. Título.

CDD 005.1

---

JOÃO DENILSON NERY SOUSA

DESENVOLVIMENTO DE UM SOFTWARE PARA AUXILIAR NA  
CONFECÇÃO DE *Circle String Art*

Trabalho de Conclusão de Curso  
apresentado ao Curso de Graduação  
em Engenharia de Software do  
Campus Russas da Universidade  
Federal do Ceará, como requisito  
parcial à obtenção do grau de  
bacharel em Engenharia de Software,

Orientador: Prof. Me. Daniel Márcio  
Batista de Siqueira

Aprovada em:

BANCA EXAMINADORA

---

Prof. Me. Daniel Márcio Batista de Siqueira (Orientador)  
Universidade Federal do Ceará (UFC)

---

Prof. Dr. Rafael Fernandes Ivo  
Universidade Federal do Ceará (UFC)

---

Prof. Me. Rodrigo Valença Cavalcante Frade  
Universidade Federal do Ceará (UFC)

Dedico este trabalho a toda minha família e meus amigos, Ismael Moreira, Maximiliano Pirilo e Francisco Wellington.

## **AGRADECIMENTOS**

Agradeço primeiramente a Deus por ter me feito chegar até onde cheguei, e por tudo de bom que ele me proporcionou.

A toda minha família que sempre me motivou a continuar. Minha mãe Rosélia, meu pai Edilson, minha avó Maria de Fátima, minha tia Ozelia e todas as minhas primas e irmã que estiveram sempre comigo.

Aos meus amigos que tive o prazer de conhecê-los no ensino médio e continuar com a parceria na faculdade, Wellington Freitas, Maximiliano Pirilo e Ismael Moreira. Agradeço demais a parceria de vocês durante todo esse período.

A todos os meus novos amigos que adquiri na faculdade.

Aos professores da faculdade que me ensinaram tantas coisas valiosas para me tornar um profissional bem qualificado.

A minha namorada Renata Rodrigues por sempre ter me motivado a continuar, e estar comigo nos piores e melhores momentos. Mesmo dizendo que minha faculdade é muito difícil. Muito obrigado por tudo.

“No fim tudo dá certo, e se não deu certo é porque ainda não chegou ao fim.”

(Fernando Sabino)

## RESUMO

*String Art* é uma técnica que consiste em confeccionar quadros feitos com linhas e pinos, é uma arte que depende da habilidade e criatividade do artista, porém, pode-se utilizar algoritmos computacionais que podem auxiliar a confeccionar estas artes, mas a grande maioria dos sistemas não possuem uma interface ou uma usabilidade boa para os usuários leigos em tecnologia, assim, dificultando a utilização. Dentro do *String Art* existe uma categoria chamada *Circle String Art*, onde os pinos são dispostos de forma circular, podendo gerar padrões ou até formas mais complexas como, por exemplo, reproduzir fotos de pessoas. Este trabalho visa desenvolver um software para facilitar a utilização desses algoritmos, além de fornecer uma base inicial para arte em *Circle String Art*, onde o artista pode desenvolver seu próprio estilo de arte utilizando a base gerada pelo algoritmo, podendo assim gerar uma nova forma de renda ou satisfação pessoal, tendo em vista as belas obras de arte que este algoritmo pode produzir.

**Palavras-chave:** String Art. Algoritmo. Circle String Art. Arte Computacional.



## ABSTRACT

*String Art* is a technique that consists of making pictures made with lines and pins, it is an art that depends on the skill and creativity of the artist, however, computational algorithms that can help to make these arts can be used, but the vast majority of systems they do not have a good interface or usability for lay technology users, thus making it difficult to use. Within *String Art* there is a category called *Circle String Art*, where the pins are arranged in a circular shape, which can generate patterns or even more complex shapes such as, for example, reproducing photos of people. This work aims to develop software to facilitate the use of these algorithms, in addition to providing an initial basis for art in *Circle String Art*, where the artist can develop his own style of art using the basis generated by the algorithm, thus being able to generate a new form of income or personal satisfaction, considering the beautiful works of art that this algorithm can produce.

**Keywords:** String Art. Algorithm. Circle String Art. Computational Art.

## SUMÁRIO

<b>1 INTRODUÇÃO</b>	<b>11</b>
<b>2 FUNDAMENTAÇÃO TEÓRICA</b>	<b>12</b>
2.1 STRING ART	12
2.2 ALGORITMO GULOSO	13
2.3 PROCESSAMENTO DIGITAL DE IMAGENS	14
<b>3 TRABALHOS RELACIONADOS</b>	<b>17</b>
3.1 STRING ART: TOWARDS COMPUTATIONAL FABRICATION OF STRING IMAGES	17
3.3 VISUAL ATTRIBUTE TRANSFER THROUGH DEEP IMAGE ANALOGY	18
<b>4 PROCEDIMENTOS METODOLÓGICOS</b>	<b>19</b>
4.1 ESTADO DA ARTE	19
4.2 ALGORITMOS RELACIONADOS	19
4.3 DESENVOLVIMENTO DO SISTEMA	20
<b>4.3.1 Receber imagem</b>	<b>20</b>
<b>4.3.2 Processamento da imagem</b>	<b>21</b>
<b>4.3.3 Adicionando pinos nas suas posições</b>	<b>22</b>
<b>4.3.4 Algoritmo de escolha do caminho</b>	<b>23</b>
<b>4.3.5 Retornando lista de ligações</b>	<b>27</b>
<b>4.3.6 Desenvolvimento da interface gráfica</b>	<b>27</b>
<b>5 RESULTADOS</b>	<b>28</b>
<b>6 CONCLUSÃO</b>	<b>32</b>
<b>7 TRABALHOS FUTUROS</b>	<b>33</b>
<b>REFERÊNCIAS</b>	<b>34</b>
<b>APÊNDICES</b>	<b>35</b>

## LISTA DE FIGURAS

Figura 1: Jesus em <i>Circle String Art</i>	12
Figura 2: Mapa	13
Figura 3: Mapa Marcado	14
Figura 4: Matriz de valores da imagem	15
Figura 5: Resultados do artigo Relacionado	18
Figura 6: Albert Einstein	20
Figura 7: Processamento da imagem	21
Figura 8: Imagem com pontos	23
Figura 9: Imagem e canvas	24
Figura 10: Escolha da linha	25
Figura 11: Processo de criação do algoritmo	26
Figura 12: Resultado	28
Figura 13: Trabalho confeccionado	29

## 1 INTRODUÇÃO

A arte é o modo mais antigo dos seres humanos expressarem sentimentos: músicas, pinturas, danças, esculturas são exemplos de artes que estão entre os humanos há muito tempo. A pintura, por exemplo, consiste em pegar substâncias líquidas ou qualquer material e aplicar sobre alguma superfície a fim de marcá-la ou representar algo. Existem diversas formas de realizar uma pintura, para tal, pode-se utilizar: pincéis, tintas, lápis de cera, carvão, pintura digital, entre outras formas.

*String Art* é uma técnica onde o artista faz pinturas por meio de linhas e pinos, visto que, muitas dessas artes são feitas manualmente onde a criatividade do artista é totalmente necessária. Uma categoria existente no *String Art* é o *Circle String Art*, onde os pinos são dispostos de forma circular, podendo, assim, gerar padrões ou replicas de fotos.

Para confeccionar *Circle String Art* pode-se criar algoritmos que nos ajude a fazer essas artes, tendo como parâmetro de entrada uma imagem qualquer e como saída uma lista contendo as ligações necessárias para gerar aquela imagem em *Circle String Art*, como por exemplo, o Weaving Algorithm, que é um algoritmo guloso que nos retorna uma sequência de ligações entre pontos para formar uma cópia de uma dada imagem em *Circle String Art*. Dessa forma, este trabalho irá fornecer um sistema de apoio a pessoas que desejam construir essas artes, porém não tem criatividade ou a técnica necessária para fazê-lo, disponibilizando um software fácil e intuitivo para qualquer usuário.

Tendo em vista que muitas pessoas vivem de arte, este software é uma forma de abrir mais portas a esses artistas, também é uma forma de incentivar a criar um novo estilo de arte. *Circle String Art* é uma técnica que segue em constante evolução, porém muitas pessoas não conhecem este tipo de trabalho, o qual, além de ajudar os artesãos, oferece um novo estilo de trabalho, além de ser uma forma de gerar renda, pois a procura deste tipo de trabalho cresce a cada dia, assim, abrir um novo mercado e oferecer novas oportunidades de trabalho a pessoas que já trabalham com arte, como também a novos artistas.

O objetivo deste trabalho visa desenvolver um sistema com boa usabilidade que gere um caminho entre pontos, de modo que transforme uma dada imagem passada pelo usuário em uma cópia da imagem em *Circle String Art*

Este trabalho está dividido em 5 sessões, onde, na primeira sessão é uma breve introdução ao assunto abordado, na segunda sessão é falado sobre a fundamentação teórica, na terceira sessão os trabalhos relacionados, na quarta sessão os procedimentos metodológicos, onde é mostrado como o projeto foi desenvolvido e como o algoritmo funciona, na quinta e última sessão o resultado do trabalho, onde é mostrado algumas imagens geradas pelo algoritmo.

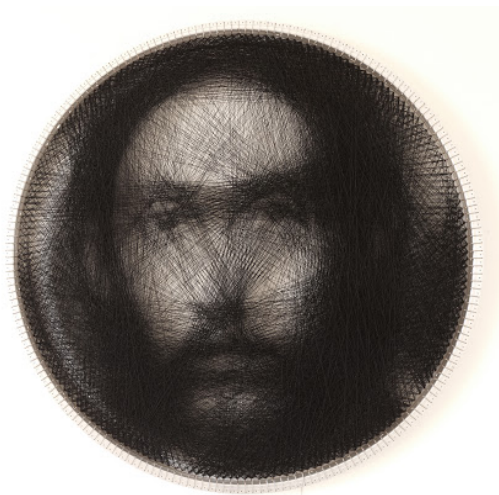
## 2 FUNDAMENTAÇÃO TEÓRICA

Nesta seção é mostrado todos os termos necessários para o desenvolvimento deste trabalho, algumas metodologias e ferramentas que foram utilizadas no desenvolvimento do presente trabalho.

### 2.1 *STRING ART*

*String Art* é uma técnica criada pela matemática Mary Everest, no século XIX, como forma de ensinar matemática para crianças. Esta técnica consiste em criar telas usando-se de linhas e pinos onde os pinos são fixos em uma tábua para que através de uma linha ligando nesses pinos possa formar padrões ou desenhos complexos (Figura 1). Essa técnica passou a ser bastante conhecida nos anos 60 com os hippies. A importância deste trabalho, visa deixar mais acessível os algoritmos para fabricar *Circle String Art* e assim trazer novas formas de trabalhos para artistas plásticos autônomos, podendo oferecer novos produtos que possam gerar algum tipo de renda ou satisfação pessoal.

Figura 1: Jesus em *Circle String Art*



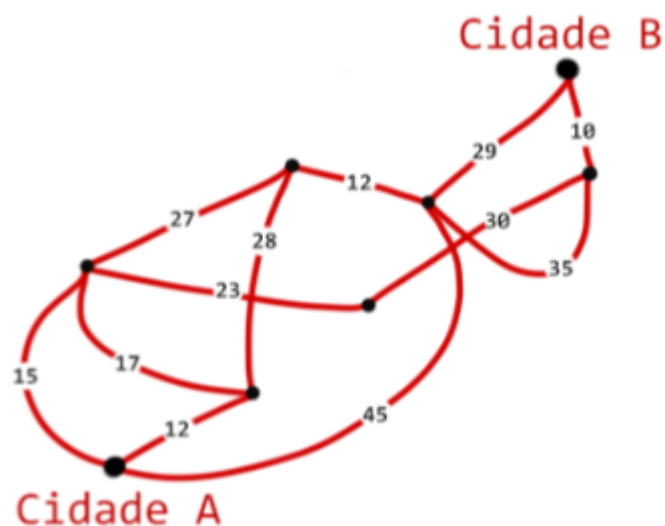
Fonte: petros vrellis (2016)

## 2.2 ALGORITMO GULOSO

Segundo (ROCHA; DORINI, 2004) os algoritmos gulosos são algoritmos que tomam decisões com base apenas na informação disponível no momento, ou seja, são algoritmos que não se preocupam se sua decisão pode afetar ou não os próximos passos, como no exemplo a seguir.

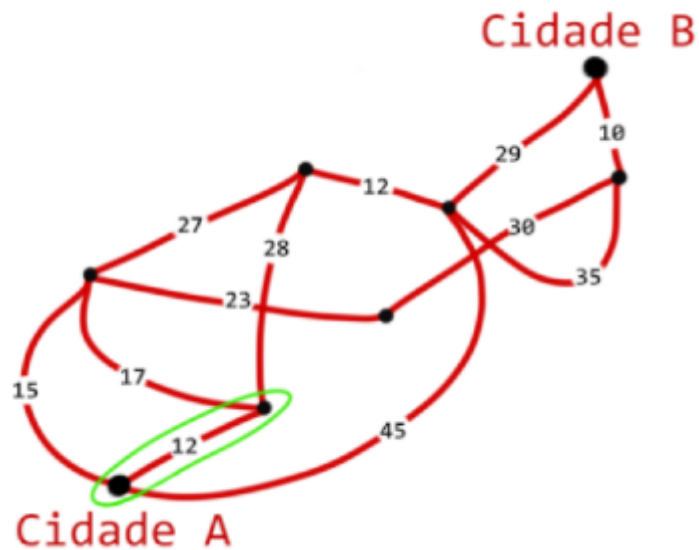
Suponha que um indivíduo deseja ir de uma cidade A para uma cidade B no menor caminho possível (Figura 2), e tudo que esse indivíduo tem é um mapa mostrando os caminhos e distâncias possíveis, e que ele não conhece o menor caminho entre estas cidades. Dessa forma, ele deverá seguir uma sequência de estradas até chegar à cidade B. Como tal indivíduo não sabe qual o menor caminho, sua única alternativa é escolher o caminho com valor mais acessível no momento, essa escolha pode ser considerada uma escolha gulosa (Figura 3).

Figura 2: Mapa



Fonte: Criada pelo autor (2019)

Figura 3: Mapa Marcado



Fonte: Criado pelo autor (2019)

A técnica gulosa é um método que não se preocupa com o futuro, para ela, a melhor escolha é aquela que é boa no momento. Voltando ao exemplo das cidades, escolher a estrada que tem a menor distância, não necessariamente vai retornar ao indivíduo o menor caminho da cidade A para a cidade B, já que o restante do caminho pode haver estradas longas. Embora seja um algoritmo fácil de implementar, é um algoritmo que não garante a melhor solução.

Utilizaremos um algoritmo guloso neste trabalho explicaremos de forma mais aprofundada o funcionamento no tópico 5.3.4 (Algoritmo de escolha dos caminhos), a escolha de algoritmos gulosos se faz bastante útil devido à sua facilidade de implementação, além de que a grande maioria dos algoritmos já feitos para esse problema são feitos com a lógica gulosa.

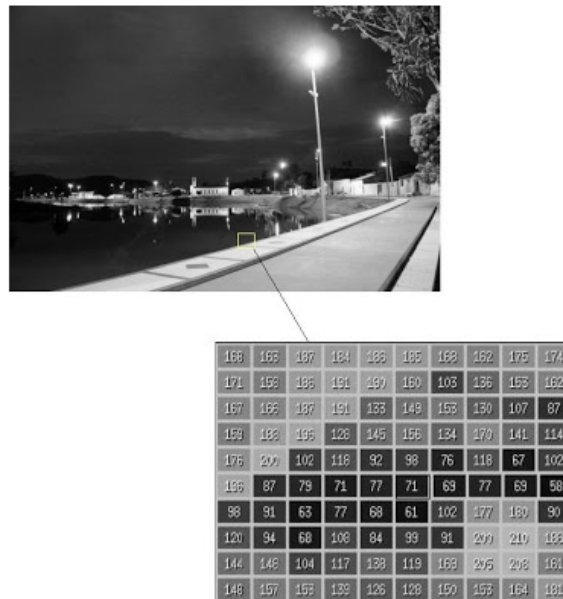
### 2.3 PROCESSAMENTO DIGITAL DE IMAGENS

Processamento digital de imagens (PDI) é uma sub-área da visão computacional, com ela, os sistemas podem aplicar filtros em uma imagem para que possa identificar objetos, melhorar a resolução de vídeos, reconhecimento facial entre outros. O uso de PDI teve início no Jet propulsion Laboratory (Pasadena, Califórnia - EUA) em 1964, quando imagens transmitidas pela câmera de TV acoplada a sonda Ranger 2 eram processadas por

computadores para corrigir vários tipos de distorções que viessem a acontecer. Existem algumas etapas genéricas para que uma imagem possa ser processada, que são as etapas de aquisição, armazenamento, processamento e exibição (MARQUES FILHO; VIEIRA NETO, 1999).

**Aquisição:** tem como principal objetivo converter uma imagem em uma representação numérica adequada para o entendimento do sistema, como por exemplo, transformar uma imagem de 100x100 pixels em uma matriz de inteiros com 100 linhas e 100 colunas, e que em cada posição da matriz é um valor no intervalo  $[0 \dots 255]$  onde 0 seria totalmente preto e 255 seria totalmente branco ( no caso de imagens em preto e branco) (Figura 4). O algoritmo utilizado neste trabalho, centra-se nesta informação para escolher qual a melhor ligação dentre todas ofertadas, que será visto no tópico 5.3.4 (Algoritmo de escolha dos caminhos).

Figura 4: Matriz de valores da imagem



Fonte: vinicius neves sdbmcc(2009)

**Armazenamento:** após convertida a imagem, deve ser feito o armazenamento dessa estrutura. Esse é um dos grandes desafios voltados ao PDI, já que, para se obter bons resultados em análises de imagens, elas devem possuir uma boa qualidade. Imagens de boa qualidade, nos dias atuais, podem chegar a (8.000 x 8.000 pixels) no caso de imagens em 8k de resolução, isso nos dá uma matriz com 64.000.000 milhões de posições para imagens em preto e branco, se considerar uma imagem colorida deve multiplicar esse valor por 3 já que terá de ser



analisado os 3 canais de cores RGB totalizando 192.000.000 milhões de dados para analisar, ou como por exemplo, os vídeos em 4K de resolução que podem chegar a 60 imagens de 4k por segundo de vídeo, informações essas que atualmente podem ser salvas em Hard Disks (HDs), Random Access Memory (RAMs), ou Solid-state Drive (SSDs) todos com espaços limitados.

**Processamento:** está diretamente relacionado aos algoritmos, bibliotecas ou softwares que serão usados para tirar as informações desejadas das imagens. Nesta etapa, podemos citar a biblioteca OpenCV, que é uma biblioteca com funções próprias para o manuseio de imagens digitais, desde funções para alterar a cor dos pixels como funções prontas para reconhecimento facial. Além disso, a biblioteca OpenCV já implementa todas essas etapas de aquisição até a exibição de imagens. O algoritmo proposto neste trabalho também é uma forma de processamento de imagem.

**Exibição:** é como essas imagens vão ser mostradas para o usuário, seja por televisores, monitores, celulares, tablets ou outros dispositivos. Disponibilizamos hoje várias tecnologias para a exibição de imagens, dentre elas, temos o Liquid Crystal Display (LCD), Light Emitting Diode(LED), Organic Light-Emitting diode (OLED) e os televisores de Plasma, tecnologias que se diferem na maneira como as imagens são exibidas, onde cada tipo de tecnologia é boa para uma situação ou ambiente em específico. Também temos as resoluções suportadas, que são a quantidade de pixels dos televisores, se um televisor é HD (1280x720) isso quer dizer que ele possui 1280 pixels de largura e 720 pixels de altura, podendo chegar hoje em dia até televisores com 4k ou como podem ser chamados de Ultra HD (3840x2160) pixels.

### 3 TRABALHOS RELACIONADOS

Para o desenvolvimento da pesquisa, foi feita uma busca sobre os trabalhos relacionados a este tema. Como é um assunto novo e os algoritmos para esse tipo problema são relativamente novos, foram encontrados poucos resultados, porém o projeto *String Art: Towards Computational Fabrication of String Images*, foi de grande ajuda para este projeto além dos materiais encontrados na internet, a maioria dos algoritmos encontrados não possuem uma interface gráfica fácil de utilizar, alguns nem possuem interface. Este trabalho fornece uma interface de fácil utilização, para que até pessoas leigas em tecnologia possa utilizar.

#### 3.1 *STRING ART: TOWARDS COMPUTATIONAL FABRICATION OF STRING IMAGES*

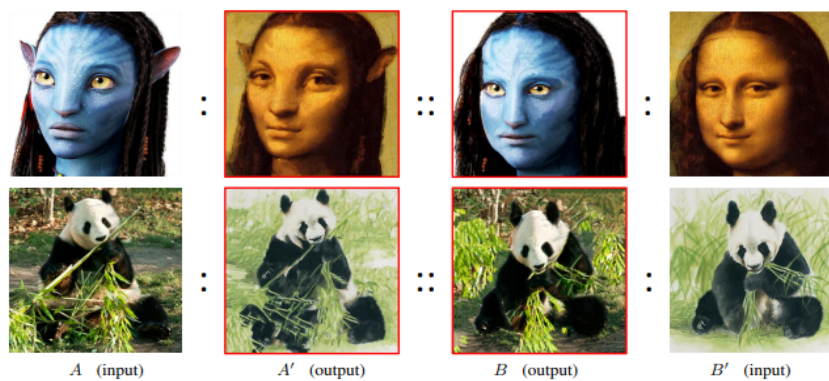
Este é um trabalho feito por *Michael Birsak, Rist Florian, Peter Wonka e Przemyslaw Musialski*, no qual o objetivo principal é desenvolver um algoritmo e fabricar uma máquina para a construção automática de *Circle String Art*, eles propõem um algoritmo de otimização que leva como entrada uma imagem em preto e cinza e converte essa imagem em um grafo conexo de linhas. Para a confecção dessas artes, eles propõem um Robô industrial que faz às conexões das linhas nos pinos de forma automática, já que para realizar este trabalho de forma manual pode ser bastante cansativo, levando em consideração a quantidade de ligações que devem ser feitas para gerar um bom trabalho (*BIRSAK; FLORIAN, 2018*).

No trabalho, é argumentado sobre a dificuldade que é encontrar o caminho para transformar uma imagem em *String Art*, pois no algoritmo, vários parâmetros devem ser avaliados para se ter um bom resultado no final do processo, como por exemplo, a quantidade de pinos, a quantidade de iterações do algoritmo, a espessura da linha entre outros. Prosseguindo, é mostrado os testes feitos e quais materiais serão usados para a confecção da imagem em *String Art*. Para a linha foi utilizado um fio de poliéster overlock, que para essa premissa, atende muito bem, e para a base eles utilizam de uma lona circular, com uma espessura específica e resistente à resistência da linha.

### 3.3 VISUAL ATTRIBUTE TRANSFER THROUGH DEEP IMAGE ANALOGY

Seguindo a mesma ideia de criar arte com algoritmos, o *Visual Attribute transfer through image analogy*, fala sobre a criação de novas artes usando artes já existentes por meio de Inteligência artificial. Neste trabalho, os autores propõem o uso de uma rede neural convolucional profunda para que possa captar atributos de imagens e mesclá-los para gerar variações dessa imagem, como mostrado na Figura 5. Neste trabalho é buscado capturar principalmente o estilo do artista, como também as texturas e cores de uma imagem e retratar em outra imagem, fundindo as duas (JING; YAO; YUAN).

Figura 05 - Resultados do artigo Relacionado



Fonte: Visual Attribute Transfer through Deep Image Analogy

## 4 PROCEDIMENTOS METODOLÓGICOS

Essa pesquisa tem como objetivo desenvolver um sistema para auxiliar na confecção de *Circle String Art*, sendo assim, em termos científicos será uma Pesquisa Aplicada, onde foi feita uma revisão, primeiramente sobre as técnicas existentes para se confeccionar *String Arts*, algoritmos propostos para o problema e sistemas possivelmente já criados para o problema e com boa usabilidade. Após a pesquisa, ocorreu o desenvolvimento do sistema, onde foi usado os artefatos encontrados na revisão para auxiliar no desenvolvimento.

### 4.1 ESTADO DA ARTE

Primeiramente, foi realizada uma pesquisa sobre *String Art*, buscando sua origem e conceitos importantes. Para isso, utilizamos uma String de busca no Google ([www.google.com](http://www.google.com)) “String AND art AND história” retornando 40.800.000 resultados. Após isso, foi feita uma pesquisa sobre algoritmos gulosos utilizando a string de busca no Google acadêmico “Algoritmos AND Gulosos” nos retornando 4.380 resultados. Dentre esses, um artigo escrito por (ROCHA; DORINI, 2004) que fala sobre definições e aplicações de algoritmos gulosos. Feito isto, foi realizada uma pesquisa sobre processamento digital de imagens (PDI) no Google Acadêmico utilizando-se da String de busca “processamento AND digital AND imagens” nos retornando 57.500 resultados dentre esse o livro “Processamento digital de imagens” dos autores (MARQUES FILHO; VIEIRA NETO, 1999).

### 4.2 ALGORITMOS RELACIONADOS

Realizou-se, primeiramente, uma busca sobre algoritmos existentes para facilitar na implementação da solução, dentre os algoritmos encontrados vamos utilizar a ideia do vídeo do canal “Universo programado” intitulado “Fiz um Quadro do Homem de Ferro usando um ALGORITMO” (link para o vídeo nos apêndices) e o artigo *String Art: Towards Computational Fabrication of String Images*. Foi utilizado este vídeo, porque ele ensina passo a passo o funcionamento deste tipo de algoritmo, além de dar um exemplo prático de como confeccionar a *Circle String Art* do modelo gerado pelo algoritmo. Vamos também utilizar o código do desenvolvedor Júlio Bandeira encontrado no github (link para o github nos apêndices).

Este algoritmo servirá como base para o nosso sistema, porque além de ser desenvolvido em python, o código em si está bem didático e de fácil interpretação, porém o

algoritmo pode ser desenvolvido em qualquer linguagem que tenha suporte a utilização de bibliotecas para edição de imagens, sendo escolhido python para o projeto devido a facilidade e afinidade do autor com a linguagem.

### 4.3 DESENVOLVIMENTO DO SISTEMA

O desenvolvimento do sistema foi dividido em etapas, onde cada etapa foi uma parte do processo de transformação em *String Art*. O desenvolvimento seguiu a seguinte ordem:

- 1) Receber imagem.
- 2) Processamento da imagem.
- 3) Adicionar pinos.
- 4) Algoritmo de escolha do caminho.
- 5) Retornar lista de ligação entre os pinos.
- 6) Desenvolvimento da interface gráfica

#### 4.3.1 Receber imagem

Nesta etapa, o usuário irá importar a imagem que ele deseja que seja feito o *Circle String Art* (Figura 6), por meio da interface do usuário interface gráfica (Figura 14)

A interface gráfica possui os seguintes parâmetros:

**Load image:** Onde a imagem vai ser carregada no App

**W - H:** Largura e Altura da imagem

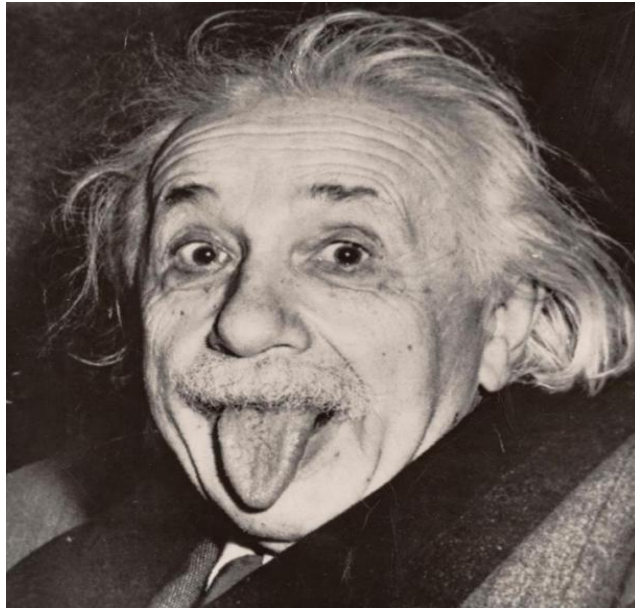
**Pinos:** Quantidade de pinos

**Iterações:** Quantidade de ligações

**Limiar:** Ajustar o Limiar da imagem

**Adjust:** Posicionamento vertical e horizontal da imagem

Figura 6: Albert Einstein



Fonte: Arthur Sasse/France Presse

#### 4.3.2 Processamento da imagem

Após o recebimento da imagem, ocorre o processamento desta aplicando um filtro de limiarização, nele a figura é transformada em uma imagem binária, ou seja, apenas tons de preto e branco e partir de um parâmetro podemos controlar a quantidade preto e branco desta imagem. Logo após, é aplicado redimensionamento na imagem para que se torne uma imagem quadrada (mesma altura e mesma largura). Aplicando esses filtros, é possível dar mais fidelidade à imagem final gerada em *Circle String Art*. A Figura 7 mostra o processamento da imagem de entrada e como deve ficar a imagem final, apenas em tons de preto e branco com contornos bem definidos.

Esta parte, é uma das mais importantes do algoritmo, pois o resultado desta etapa irá impactar no resultado final da arte. Para essa parte, foi usada a biblioteca OpenCV do python, pois é uma biblioteca completa e fácil de ser manipulada, dentro desta biblioteca existe uma função chamada “threshold” que pode ser chamada para aplicar o filtro de Limiarização mencionado no texto.

Figura 7: Processamento da imagem



Fonte: Autor deste trabalho (2019)

#### 4.3.3 Adicionando pinos nas suas posições

Depois da imagem estar preparada, é adicionado os pinos. Os pinos são dispostos de maneira circular pegando as extremidades da imagem, já que será uma imagem quadrada. Como queremos um círculo que pegue a maior quantidade possível da imagem, os pinos devem ficar na extremidade da imagem, para isso o raio do círculo será a metade da largura da imagem (como é uma imagem quadrada, pode-se usar a metade da altura da imagem como raio também), (Figura 8), com o raio, basta achar a posição do primeiro pino. A coordenada X do primeiro pino é a soma do (raio + centro) multiplicado pelo cosseno do ângulo, como é o primeiro pino o cosseno do ângulo é 0.

O usuário irá escolher a quantidade de pinos, com este valor informado, é calculado o cosseno do ângulo, bastando dividir a quantidade de pinos informado por 360, assim retornando o valor do ângulo. A mesma ideia é aplicada a coordenada Y do pino, porém é trocado o cosseno(cos) por Seno(sen).

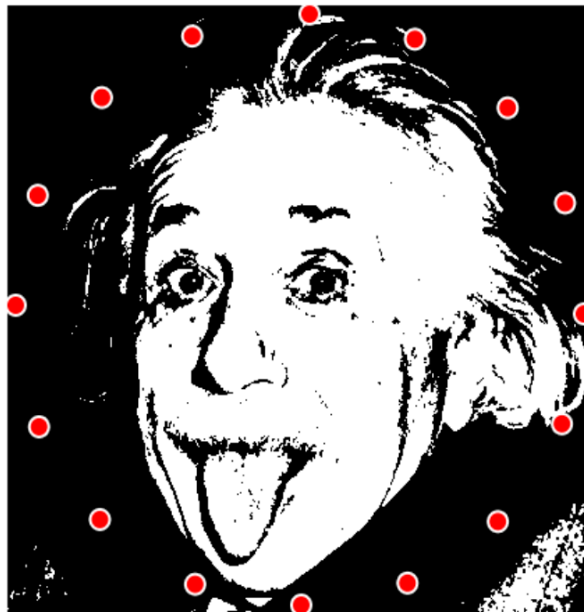
Coordenada (X,Y) do pino

$$X = (\text{centro} + \text{diâmetro}) * \cos(\hat{\text{ângulo}})$$

$$Y = (\text{centro} + \text{diâmetro}) * \text{sen}(\hat{\text{ângulo}})$$

Em seguida o sistema retorna uma imagem com os pinos posicionados. Também é criado um rótulo que será usado para o usuário saber em qual pino a linha deverá ser ligada. Por final criar uma lista contendo a posição (x, y) de todos os pinos gerados, essa lista será usada posteriormente no algoritmo, o rótulo pode ser o índice da lista criada.

Figura 8: Imagem com pontos



Fonte: Denilson Sousa (2019)

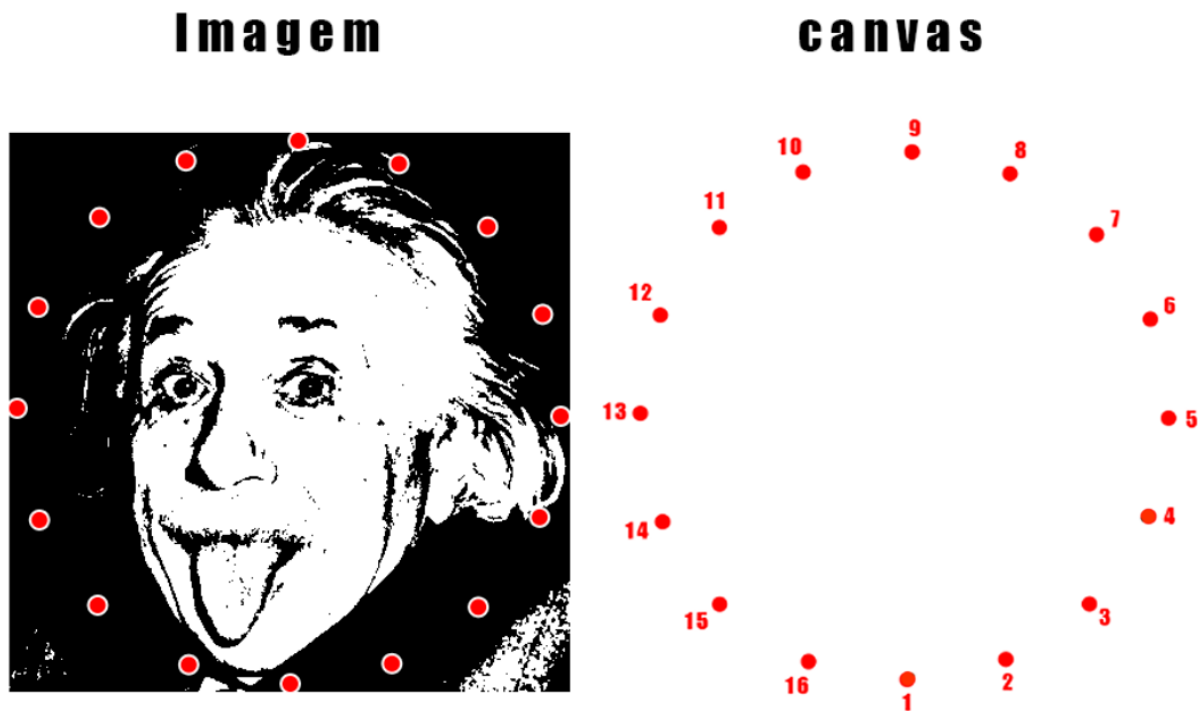
#### 4.3.4 Algoritmo de escolha do caminho

Logo depois dos pinos estarem nas suas posições, o sistema calcula o caminho que o usuário deverá ligar a linha nos pinos para que a imagem seja gerada. Esse algoritmo será feito com base nos algoritmos relacionados citados na seção 4.2. O passo a passo do algoritmo está descrito abaixo.

1. Criar uma imagem completamente branca, contendo as mesmas dimensões da imagem que será transformada em *Circle String Art*, que será chamado de Canvas. A Figura 9 demonstra como é o canvas e qual o estágio atual da imagem, lembrando que a figura é apenas um exemplo e que na realidade a quantidade de pinos é maior. Para o exemplo utilizamos 16 pinos.



Figura 9: Imagem e canvas



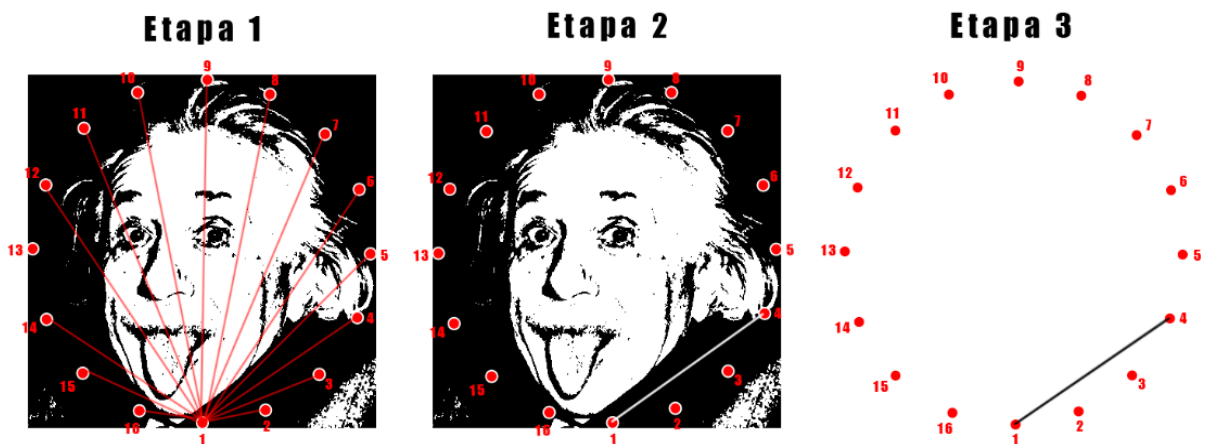
Fonte: Denilson Sousa (2019)

2. Dentre a lista de pinos da imagem original escolher um pino inicial, por exemplo o pino com rótulo 1.

3. Traçar uma reta do ponto escolhido até seu vizinho, e contar a quantidade de pixels pretos que existem nessa reta. Fazer esse processo para todos os vizinhos desse ponto. A reta que tiver a maior quantidade de pixels pretos será retirada da imagem original e enviada para o canvas (Figura 10). Repare-se, que na imagem original a reta escolhida teve todos os pixels pintados de branco, isso se faz necessário, para que essa reta não seja verificada novamente, pois caso contrário o algoritmo nunca terminaria.

O algoritmo utilizado para a análise da linha, ou seja, a forma como a quantidade de pixels pretos foi calculada foi utilizado o *Digital Differential Analyser* ou DDA, com ele, a partir de uma laço de repetição é possível desenhar linhas em uma imagem, como também, capturar os pixels de uma linha para análise, que no caso do trabalho, esse algoritmo é utilizado para a contagem dos pixels pretos da linha.

Figura 10: Escolha da linha



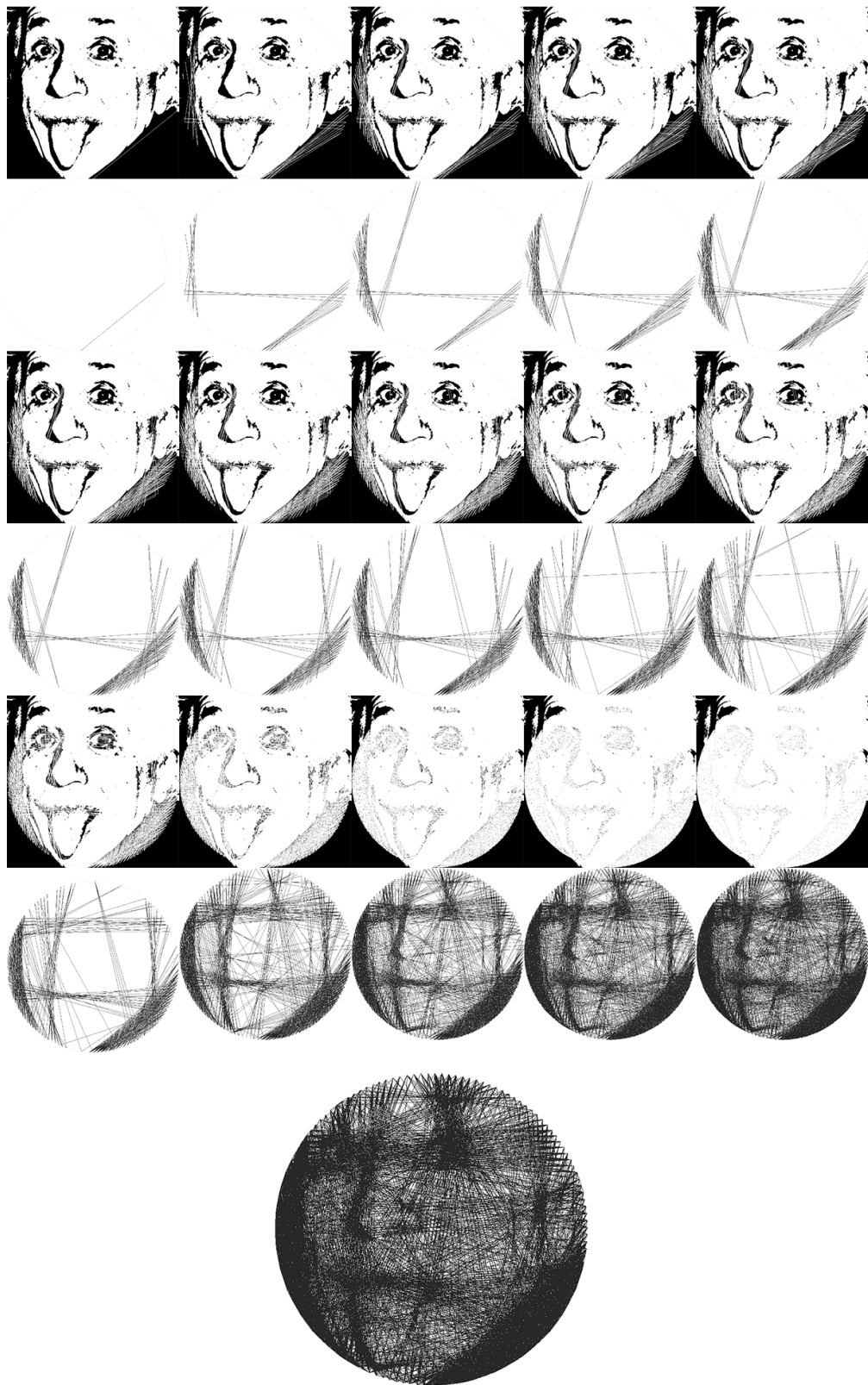
Fonte: Autor deste trabalho (2019)

4. O próximo ponto a ser escolhido será o ponto final da reta escolhida anteriormente, no nosso caso o ponto 4, voltamos ao passo 3 e continuamos o algoritmo. É interessante fazer desta forma, porque na hora de confeccionar manualmente não haverá muitas quebras na linha.

5. O algoritmo continuará se repetindo até que o usuário esteja satisfeito com o resultado, ou atingir o número máximo de ligações dado pelo usuário (Figura 11). Na figura é mostrado o processo de criação, nesse exemplo foi utilizado como parâmetro uma imagem de Albert Einstein, com 139 pinos, e 1500 ligações, e com o valor do limiar em 80.

A cada novo pino descoberto ele é armazenado em uma lista, para que o final possa ser retornado para o usuário.

Figura 11: Processo de criação do algoritmo



Fonte: Denilson Sousa (2019)

#### 4.3.5 Retornando lista de ligações

Após terminar o cálculo dos caminhos, o sistema enviará um “arquivo.txt” para o usuário contendo dicas para a confecção daquela imagem em *Circle String Art*, e por fim, fornecer as ligações entre os pinos para a criação da peça.

Por exemplo:

Pino 1 ao Pino 4.

Pino 4 ao Pino 3.

Pino 3 ao Pino 14.

...

#### 4.3.6 Desenvolvimento da interface gráfica

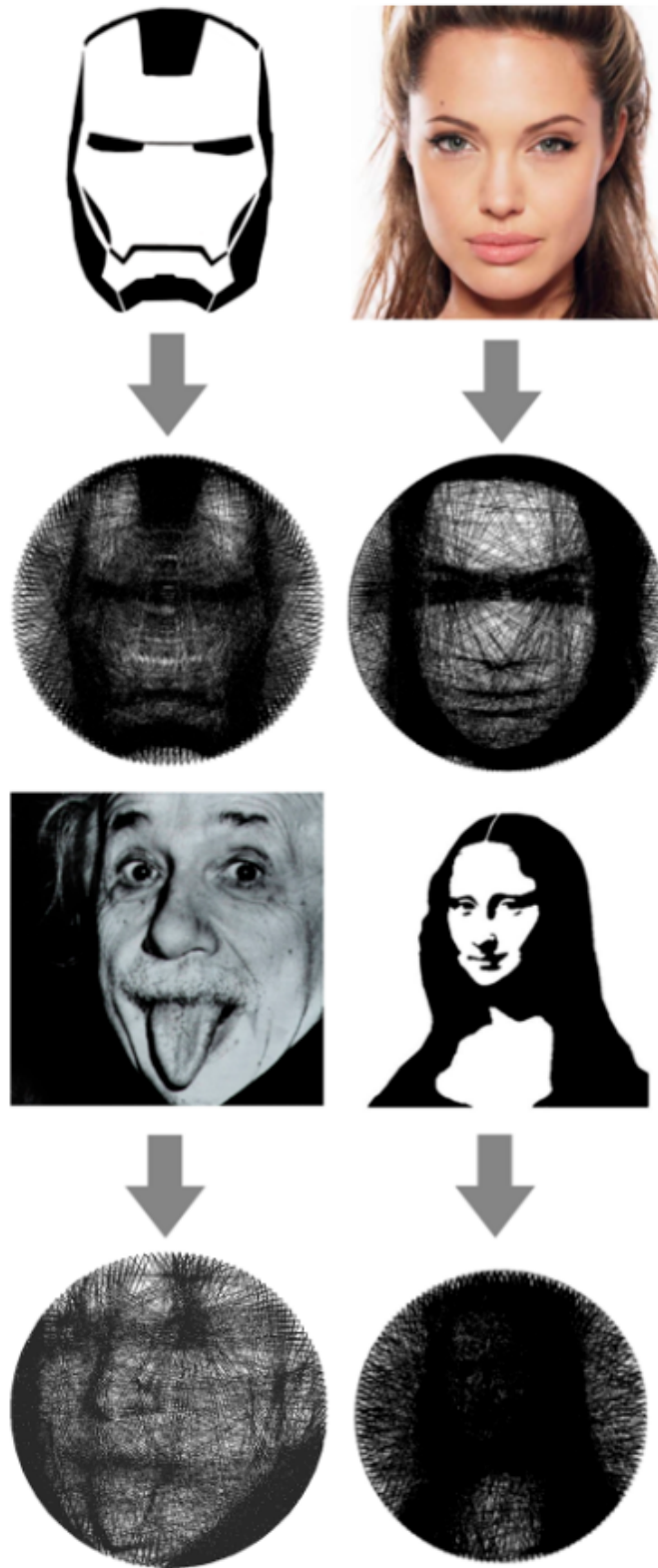
Para o desenvolvimento da interface do usuário foi utilizado a biblioteca PySimpleGui que é uma biblioteca completa para o desenvolvimento de telas simples. É possível criar janelas GUI usando um "layout" que contém widgets (eles são chamados de "Elementos"). O layout é usado para criar uma janela usando uma das 4 estruturas suportadas para exibir e interagir com ela. O código nesta biblioteca é mais simples e mais curto do que escrever diretamente usando a estrutura subjacente como por exemplo, o tkinter. Além disso, as interfaces são simplificadas para exigir o mínimo de código possível para obter o resultado desejado. Como os parâmetros que são desejados no projeto são poucos, a biblioteca trás bastante benefício pois é possível deixar o código mais acessível com pouco código.

## 5 RESULTADOS

Com o algoritmo encontrado feito pelo desenvolvedor Júlio Bandeira conseguimos gerar as seguintes imagens (Figura 12). A maioria dos algoritmos existentes se baseiam nessa premissa de escolher os pixels pretos de uma imagem, o que vai definir se a imagem vai ficar boa ou não no final é o fator de escolha das melhores linhas, a maioria dos algoritmos existentes são privados ou não tem lógica compartilhada, muitos deles utilizam de Inteligência artificial para a escolha das melhores linhas, outros utilizam de pontuação, alguns algoritmos funcionam na imagem em preto e cinza, outros utilizam na imagem limiarizada, infelizmente o autor deste trabalho não pode ter acesso a lógica desenvolvida dos melhores algoritmos, por isso a escolha do método mais básico.

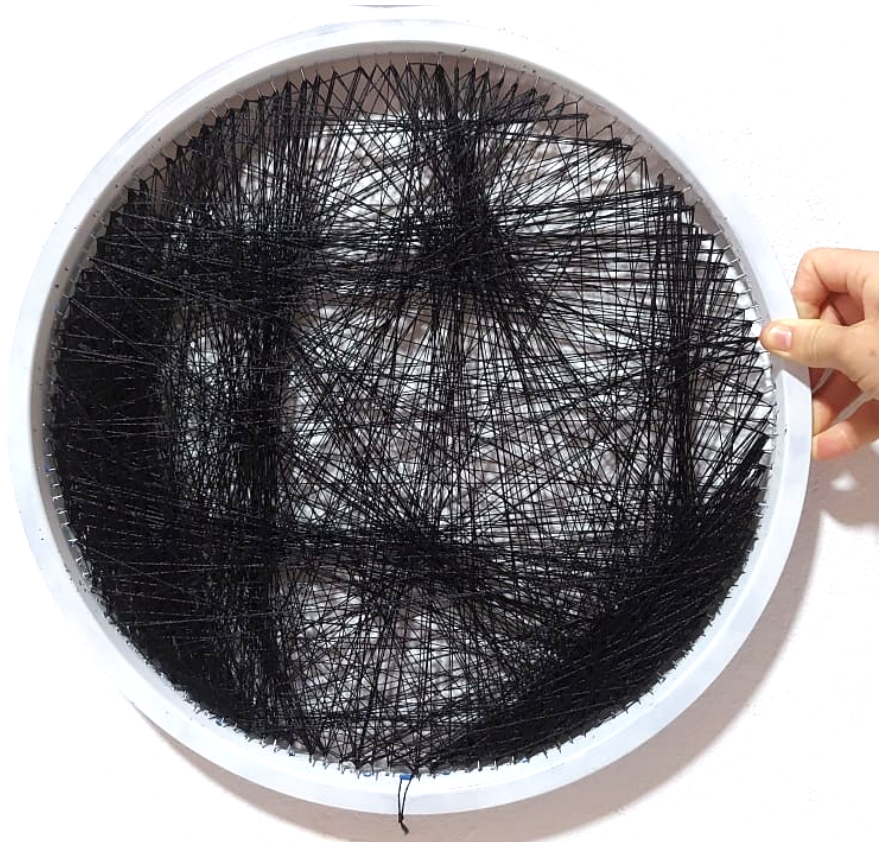
Para gerar as imagens, utilizamos dos valores padrões dos parâmetros do algoritmo (quantidade de pinos = 144, quantidade de ligações = 1500, limiar = 80), sendo que para se obter melhores resultados, os parâmetros devem ser alterados e ajustados para cada imagem. Lembrando que quanto mais pinos, mais resolução a imagem pode ganhar, porém o algoritmo levará mais tempo para calcular, assim como a quantidade de ligações. Quanto mais ligações, mais escura sua imagem pode ficar, porém com esse algoritmo vai chegar um momento limite, ou seja, quando a imagem original não terá mais pixels pretos, não é recomendado rodar o algoritmo até esse ponto, sendo que o fator visual do artista pode estabelecer quando o algoritmo deve parar. O link no Github para código do projeto deste trabalho se encontra nos apêndices.

Figura 12 - Resultados



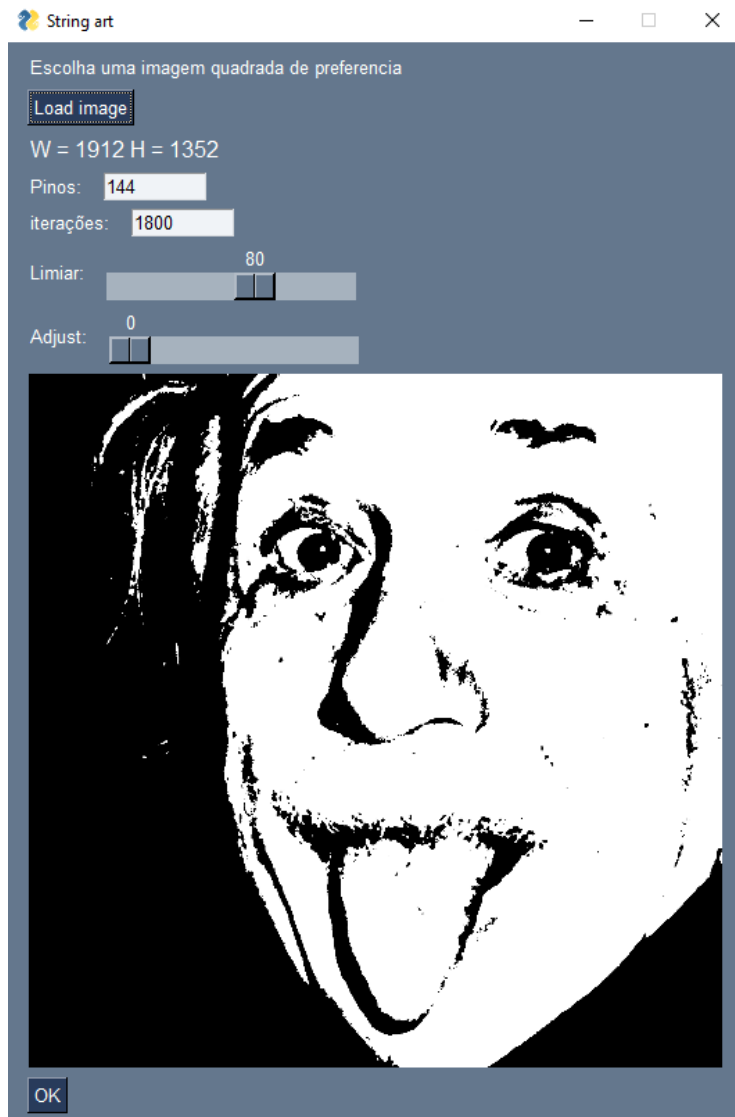
Fonte: Denilson Sousa (2019)

Figura 13: Trabalho confeccionado



Fonte: autor deste trabalho

Figura 14: Interface gráfica do App



Fonte: Autor deste trabalho



## 6 CONCLUSÃO

Durante a confecção do modelo gerado pelo algoritmo notou-se um interesse por parte das pessoas que viram o desenvolvimento, mostrando assim uma certa curiosidade na arte. O algoritmo pode retornar belos trabalhos dependendo do material e parâmetros informados na interface gráfica, porém, é limitado a certas imagens escuras pois muitas linhas são desperdiçadas, pois muitas ligações vão ser direcionadas a estas áreas escuras.

Com base nas pessoas que viram a arte feita (Figura 13) aceitaram totalmente a arte, e foi notado um certo desejo, mostrando que é uma arte muito promissora e que pode, sem dúvidas, aumentar a renda de muitos artistas locais e também de qualquer lugar do mundo que utilize este tipo de algoritmo para desenvolver seus trabalhos.

Na figura 13 é visto como a escolha dos materiais é de extrema importância para se ter um bom resultado, neste trabalho foi utilizado um aro de bicicleta e uma linha de seda, porém a linha de seda disponível não era da espessura correta para o tamanho do círculo, da pra notar que a imagem ficou escura com poucas ligações, para gerar este trabalho usado, 900 ligações e limiar 74, sendo a quantidade de ligações interrompidas devido que ligações acima de 900 a imagem estava ficando muito escura devido a espessura da linha. Este algoritmo abre muitas oportunidades não só para artesãos autônomos como também para grandes empresas, como, por exemplo, a indústria da moda e decoração, podendo estampar estas artes em camisas, calças, cortinas, entre outros, oferecendo mais valor para estas peças.

## **7 TRABALHOS FUTUROS**

Como trabalhos futuros é muito importante melhorar o algoritmo, para que possa gerar imagens cada vez mais realistas, buscando melhores formas de escolher as ligações; edição de imagens para preparar a imagem antes de rodar o algoritmo; como também expandir o nicho, criando formas de gerar padrões complexos onde apenas a criatividade do artista que vai ser o limitador destes trabalhos. Apresentar o algoritmo aos artistas locais e difundir esta arte cada vez mais. Por fim, ver os frutos gerados por meio deste algoritmo, melhores condições de vida aos artistas e artesãos locais e não locais, reconhecimento dos mesmos e estimular a criação de novas artes.

## REFERÊNCIAS

BIRSAK et al. *String Art: Towards Computational Fabrication of String Images*. Computer Graphics Forum (Proc. EUROGRAPHICS 2018), Russas, v. 37, n. 2, p. 1 – 12, apr 2018. Disponível em: <https://www.cg.tuwien.ac.at/research/publications/2018/Birsak2018-SA/>. Acesso em: 18/11/2019.

Jing; Yao; Yuan. *String Art: Visual Attribute Transfer through Deep Image Analogy*. Disponível em: <https://arxiv.org/abs/1705.01088>. Acesso em: 09/03/2021.

MARQUES FILHO; VIEIRA NETO. *Processamento Digital de Imagens*. Rio de Janeiro: Brasport, 1999. 307 p. Disponível em: <http://www.ogemarques.com/wpcontent/uploads/2014/11/pdi99.pdf>. Acesso em: 15/10/2019.

ROCHA; DORINI. *Algoritmos gulosos: definições e aplicações*. 2004. Artigo. Disponível em: <http://www.ic.unicamp.br/~rocha/msc/complex/algoritmosGulososFinal.pdf>. Acesso em: 15/10/2019.

SOUTO. *O olho humano e a visão*. 2018. Disponível em: <https://pt.khanacademy.org/science/6-ano/vida-e-evolucao-6-ano/a-visao-e-seus-defeitos/a/o-olho-humano-e-avisao>. Acesso em: 15/10/2019.

SDBMCC. *Imagem Digital - Teoria*. 2009. Disponível em: <http://sdbmcc.blogspot.com/2009/06/imagem-digital-teoria.html> Acesso em: 27/03/2021.

CORDS. *Por que Albert Einstein mostrou a língua para o mundo?* 2009. Disponível em: <https://noticias.uol.com.br/ultimas-noticias/deutschewelle/2021/03/14/por-que-albert-einstein-mostrou-a-lingua-para-o-mundo.html> Acesso em: 27/03/2021.

VRELLIS. *A new way to knit*. 2016. Disponível em: <http://artof01.com/vrellis/works/knit.html>. Acesso em: 28/03/2021.

## APÊNDICES

Universo programado - vídeo - Fiz um Quadro do Homem de Ferro usando um ALGORITMO. 29 ago 2019. Disponível em : < <https://www.youtube.com/watch?v=YZtx4jNNbx8&t=47s> >

Julio Bandeira - GitHub - Forme imagens por meio de linhas - 15 set 2019 Disponível em: < [https://github.com/julio-bandeira/pyhton\\_weaver](https://github.com/julio-bandeira/pyhton_weaver) >

*Circle String Art* Generator – projeto deste trabalho – 13 mar 2021 Disponível em: <<https://github.com/DenilsonLunna/Circle-String-Art-Genererator> >