

Esse modulo define uma classe [Imagem](#)

## Modules

[PIL.Image](#)

## Classes

[builtins.object](#)

[Imagem](#)

class **Imagem**([builtins.object](#))

Esta classe representa uma imagem

Methods defined here:

**\_\_init\_\_**(self, caminho, nome, formato)

Esse metodo cria um objeto da classe [Imagem](#)

:param caminho: caminho da imagem

:param nome: nome da imagem

:param formato: formato da imagem

**abrir**(self)

Esse metodo abre uma imagem

**carregar**(self)

Esse metodo carrega os pixels da imagem

**inicializar**(self)

Esse metodo abre e carrega uma imagem, alem de salvar a largura e altura

**salvar**(self, largura, altura, imagem\_rotulada, tabela)

Esse metodo salva uma imagem

:param largura: largura da imagem inicial

:param altura: altura da imagem inicial

:param imagem\_rotulada: lista que contem ids

:param tabela: objeto da classe Tabela

Data descriptors defined here:

**\_\_dict\_\_**

dictionary for instance variables (if defined)

**\_\_weakref\_\_**

list of weak references to the object (if defined)

Esse modulo define uma classe [Mapa](#)

## Classes

[builtins.object](#)

[Mapa](#)

```
class Mapa(builtins.object)
```

Esta classe representa um mapa

Methods defined here:

```
__init__(self, tamanho)
```

Esse metodo cria um objeto da classe [Mapa](#)

:param tamanho: as dimensoes do mapa (x,y)

```
arredondar(self)
```

Esse metodo arredonda todos os elementos da lista

```
atualizar(self, posicao, referencia, epoca)
```

Esse metodo atualiza o mapa

:param posicao: a posicao do melhor neuronio

:param referencia: um pixel da imagem original

:param epoca: a epoca do treinamento

```
classificar(self, referencia)
```

Esse metodo encontra o melhor neuronio no mapa

:param referencia: um pixel da imagem original

:return: a posicao do melhor neuronio

```
gerar_tabela(self)
```

Esse metodo gera uma tabela, da seguinte forma: [id, R, G, B]

:return: um objeto da classe Tabela

---

Data descriptors defined here:

```
__dict__
```

dictionary for instance variables (if defined)

```
__weakref__
```

list of weak references to the object (if defined)

## Functions

```
exp(...)
```

[exp](#)(x)

Return e raised to the power of x.

Esse modulo define uma classe [Tabela](#)

## Classes

[builtins.object](#)

[Tabela](#)

class **Tabela**([builtins.object](#))

Esta classe representa uma tabela

Methods defined here:

**\_\_init\_\_**(self, lista)

Esse metodo cria um objeto da classe [Tabela](#)

:param lista: a lista que contem os identificadores e o respectivo RGB

**classificar**(self, referencia)

Esse metodo encontra o melhor neuronio na tabela

:param referencia: um pixel da imagem original

:return: o identificador do melhor neuronio

---

Data descriptors defined here:

**\_\_dict\_\_**

dictionary for instance variables (if defined)

**\_\_weakref\_\_**

list of weak references to the object (if defined)

## Functions

**exp(...)**

[exp](#)(x)

Return e raised to the power of x.

## bitwise

Esse modulo define uma classe [Bitwise](#)

## Classes

[builtins.object](#)

[Bitwise](#)

class **Bitwise**([builtins.object](#))

Esta classe representa o bitwise

Methods defined here:

**\_\_init\_\_**(self, quant)

Esse metodo cria um objeto da classe [Bitwise](#)

:param quant: a quantidade de bits

**ampliar**(self, bytes\_entrada, tamanho)

Esse metodo amplia uma lista de quant bits para uma lista de 8 bits

:param bytes\_entrada: uma lista de bytes reduzida

:param tamanho: o tamanho da lista original

:return: uma lista de bytes ampliada

**reduzir**(self, bytes\_entrada)

Esse metodo reduz uma lista de 8 bits para uma lista de quant bits

:param bytes\_entrada: uma lista de bytes

:return: uma lista de bytes reduzida

---

Data descriptors defined here:

**\_\_dict\_\_**

dictionary for instance variables (if defined)

**\_\_weakref\_\_**

list of weak references to the object (if defined)

## Functions

**ceil(...)**

[ceil](#)(x)

Return the ceiling of x as an int.  
This is the smallest integral value  $\geq x$ .

## arquivo

[index](#)

Esse modulo define uma classe [Arquivo](#)

## Classes

[builtins.object](#)

[Arquivo](#)

class **Arquivo**([builtins.object](#))

Esta classe representa um arquivo

Methods defined here:

**\_\_init\_\_**(self, caminho, nome, formato)

Esse metodo cria um objeto da classe [Arquivo](#)

:param caminho: caminho do arquivo

:param nome: nome do arquivo

:param formato: formato do arquivo

**escrever**(self, cabecalho)

Esse metodo escreve um conjunto de bytes no arquivo

:param cabecalho: o conteudo que sera escrito no arquivo

**ler**(self)

Esse metodo le o arquivo

:return: o conteudo do arquivo, em bytes

---

Data descriptors defined here:

**\_\_dict\_\_**

dictionary for instance variables (if defined)

**\_\_weakref\_\_**

list of weak references to the object (if defined)

## compactar

[index](#)

Esse modulo e responsavel pelo processo de compactacao

### Modules

[gzip](#)

### Functions

**compactar**(caminho, nome, formato)

Essa funcao compcta uma imagem e salva num arquivo .SOM

:param caminho: o caminho da imagem

:param nome: o nome da imagem

:param formato: o formato da imagem

## descompactar

[index](#)

Esse modulo e responsavel pelo processo de descompactacao

### Modules

[gzip](#)

### Functions

**descompactar**(caminho, nome, formato)

Essa funcao descompacta o arquivo e salva uma imagem no formato .TIFF

:param caminho: o caminho do arquivo compactado  
:param nome: o nome do arquivo compactado  
:param formato: o formato do arquivo compactado

## matematica

[index](#)

Esse modulo contem funcoes matematicas

## Functions

**aprendizado**(epoca)

Essa funcao calcula o aprendizado

:param epoca: numerador  
:param decaimento: denominador  
:return: o resultado da funcao exponencial

**distancia\_euclidiana**(x, y, n)

Essa funcao calcula a distancia euclidiana entre n pontos

:param x: um numero ou uma lista de numeros  
:param y: um numero ou uma lista de numeros  
:param n: quantidade de pontos  
:return: a distancia euclidiana

**exp**(...)

[exp](#)(x)

Return e raised to the power of x.

**gaussiana**(distancia, abertura)

Essa funcao calcula uma funcao de gauss

:param 0.1: altura do pico da curva  
:param distancia: posicao do centro do pico  
:param abertura: largura do sino  
:return: o resultado da funcao exponencial