

# INE5416 - Paradigmas da Programação (2015/2)

*Projeto T3B: Aprendizado de máquina*  
Caique Rodrigues Marques 13204303  
Gustavo José Carpeggiani 13103524  
Vinícius Couto Biermann 13100778

## Regras

As seguintes regras foram implementadas na linguagem de programação Prolog, as funções de cada uma são executadas em imagens em formato ascii PGM. No banco de dados, as imagens estão referenciadas por um rótulo (nome) e sete momentos de HU, isto são representações da imagem, a partir disto que a máquina se baseia para classificar qual a silhueta presente na imagem. A cada novo aprendizado, a máquina pergunta ao usuário qual a real imagem e armazena ao seu banco de dados ou "base de conhecimento". A cada vez que o programa é executado, é esperado que a máquina reconheça os diversos tipos de silhueta.

## Regras básicas

Operações básicas foram implementadas para a administração do banco de dados, de forma em que possa verificar e adicionar elementos, por exemplo, a seguir, as operações implementadas:

```
load :-
    retractall(img(-, -, -, -, -, -, -, -)),
    open('imgdatabase.pl', read, Stream),
    repeat,
        read(Stream, Data),
        (Data == end_of_file -> true ; assert(Data), fail),
        !,
    close(Stream).

commit :-
    open('imgdatabase.pl', write, Stream),
    telling(Screen),
    tell(Stream),
    listing(img/8),
    tell(Screen),
    close(Stream).

new(FileName, Id) :-
    readPGM(FileName, I),
    coord(I, Iout),
    hu(Iout, I1, I2, I3, I4, I5, I6, I7),
    assertz(img(Id, I1, I2, I3, I4, I5, I6, I7)),
    !.

search_all(Id) :-
    listing(img(Id, -, -, -, -, -, -, -)).
```

- *load/0*: o usuário utiliza esta regra para carregar o banco de dados de imagens do sistema.
- *commit/0*: esta regra, quando executada salva as modificações feitas pelo usuário no banco de dados.
- *new/2*: usada para carregar um arquivo de imagem externo e guardá-lo no banco de dados, o diretório do arquivo é especificado no parâmetro *FileName*. O sistema irá associar o valor do parâmetro *Id* com o arquivo de imagem no banco de dados.
- *search\_all/1*: lista todos os elementos do banco de dados. O parâmetro *Id* corresponde a um nome da imagem registrada que será entendido como um nome genérico, logo, é mostrado todos os elementos presentes no banco de dados.

## euclidean\_dist

Realiza o cálculo da distância euclidiana entre a imagem de entrada e uma outra imagem, de uma lista com o conteúdo do banco de dados. Esta regra é usada pela máquina para fazer as aproximações das imagens recebidas, de forma que ele consiga deduzir algo partindo dos seus "conhecimentos" (conteúdo do banco de dados) e comparando com a imagem recebida.

## compare\_images

Compara as imagens do banco de dados com os sete momentos de HU recebidos, onde, através da regra que calcula a distância euclidiana, a máquina coleta os menores momentos de HU de todas essas imagens. A regra armazena os resultados na lista de saída *Output*, isto serve, posteriormente ao coletar o menor elemento desta lista, fazer uma sugestão de qual seria a imagem que foi sugerida pelo usuário de início.

## scan\_image

Nesta regra é que a máquina começará a verificar a imagem e citar uma sugestão do que seria baseando em seus conhecimentos, no caso, o conteúdo armazenado em seu banco de dados (no arquivo *imgdatabase.pl*). A regra começa com recebimento de uma imagem como parâmetro, a regra *readPGM/2* transforma a imagem PGM em uma matriz de coordenadas, depois, é convertido em uma lista de coordenadas, que está na variável *FileCoord*. A regra *hu/8* coleta os sete momentos de HU de uma lista de coordenadas, em seguida, a regra nativa *findall/3* coleta o conteúdo do banco de dados e armazena na variável *Data\_List*.

*compare\_images* verifica os momentos de HU da imagem com os momentos das imagens no banco de dados, a resposta está armazenada numa lista chamada de *Compare\_Out*, o menor elemento desta é armazenado em *Minimal* e sua posição em *Index*. A mesma posição do elemento com o menor valor de HU, *Minimal*, tem a posição correspondente com a lista *Data\_List* que possui o nome e os momentos dos elementos no banco de dados, portanto, as informações são armazenadas em *Image*, uma tupla com o nome e os sete momentos e será a conclusão que a máquina chegou. O último uso da regra *nth0/3* serve para coletar apenas o primeiro elemento de *Image*, ou seja, o nome.

Por fim a regra lê a entrada do usuário após perguntar se a imagem que ele concluiu é a mesma que o foi especificada de entrada. Se sim ("y" ou "yes"), a máquina verifica se a imagem já estava no banco de dados, senão ele adiciona a imagem com uma nova perspectiva (por exemplo, a máquina já tinha "bat-6" em seu banco de dados, a imagem de entrada foi "bat-2", logo ele adiciona banco de dados esta nova perspectiva). Se não ("n" ou "no"), a máquina pergunta qual foi a imagem de entrada e adiciona a informação ao seu banco de dados, junto com os seus sete momentos de HU.

A seguir, alguns exemplos de reconhecimento das imagens, de início, a máquina só possui uma imagem em seu banco de dados, chamada de "Pockety", que é a silhueta de um relógio de bolso. Depois do usuário instruir à máquina a imagem de um morcego (*bat image*), ela passa a reconhecer quando o usuário referencia a um morcego.

```
?- scan_image('pgm/bat-6.pgm').
pgm/bat-6.pgm
Note: Don't forget to end yours answers with a dot (.)
```

```
Minimal value found: 0.12113930633134963
Image found: Pockety
Position: 0
```

```
This is your image, young padawan? [y./n.]
|: n.
Name thy image: 'Batima'.
All men and machine by nature desire knowledge.
Wow, perplexity is the beginning of knowledge!
Very good!
```

```
?- scan_image('pgm/bat-2.pgm').
pgm/bat-2.pgm
Note: Don't forget to end yours answers with a dot (.)
```

Minimal value found: 0.05774930530830736  
Image found: Batima  
Position: 1

This is your image, young padawan? [y./n.]  
|: y.  
The same in a new perspective!  
Good!

---

?- scan\_image('pgm/bat-2.pgm').  
pgm/bat-2.pgm  
Note: Don't forget to end yours answers with a dot (.)

Minimal value found: 0.0  
Image found: Batima  
Position: 2

This is your image, young padawan? [y./n.]  
|: y.  
Thy image is already in the database.

---

?- scan\_image('pgm/pocket-20.pgm').  
pgm/pocket-20.pgm  
Note: Don't forget to end yours answers with a dot (.)

Minimal value found: 0.0  
Image found: Pockety  
Position: 0

This is your image, young padawan? [y./n.]  
|: y.  
Thy image is already in the database.