

Universidade Tecnológica Federal do Paraná – UTFPR
Departamento Acadêmico de Eletrônica – DAELN
Engenharia de Computação
Disciplina: IF69D – Processamento Digital de Imagens
Semestre: 2013/2
Prof: Gustavo B. Borba

RELATÓRIO

Reconhecimento de cores utilizando Lógica Fuzzy

Alunos:
John Théo Sierpinski de Souza / 938459
Tiago Rafael Volkmann Coelho / 889237

Mar.2014

1 Objetivo

Implementar no MATLAB uma lógica fuzzy para reconhecimento de cores combinações de RGB em imagens.

2 Fundamentação Teórica

Definiu-se que para esse projeto utilizaria-se o mapa de cores RGB e lógica Fuzzy por permitir uma definição mais "maléavel" dos limites das níveis de vermelho, verde e azul da imagem.

2.1 Color Maps - mapas de cores

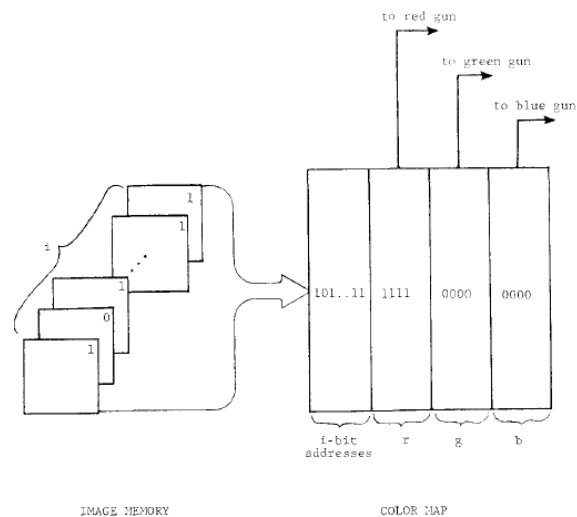


Figura 1: Mapa de cores. Um *i-bit* na memória da imagem é usado como índice no mapa de cores. Neste endereço no mapa está um valor que é interpretado pelo hardware como uma cor ou uma intensidade para ser mostrada.

Fonte: [2]

Segundo [2] O mapa de cores é uma tabela que especifica a transformação dos *bits* de memória da imagem e os *bits* que serão apresentados no *display* 1. A tabela é endereçada pelo valor do pixel na memória da imagem. O conteúdo em cada endereço é um conjunto de *bits* que determina qual sinal será enviado ao *display*.

Se a memória da imagem for do tipo *single-bit plane*, existem apenas dois endereços no mapa de cores sendo um para cada pixel. Os valores do mapa de cores podem ter tamanho variado, podem ser de 24 *bits* e especificar 8 *bits* de intensidade para ser enviado para cada uma das três cores RGB no monitor.

2.2 Lógica Fuzzy

A lógica *Fuzzy* possui o objetivo de modelar situações onde existem incertezas ou imprecisões. Segundo [1] estes casos não podem ser representados através de conjuntos binários os quais possuem apenas dois graus de pertinência, enquanto que nos conjuntos *fuzzy* o elemento pode ter um grau de pertinência que vai de 0 a 1. Isto possibilita o sistema a representar os conhecimentos linguísticos, os quais representamos dizendo por exemplo, que uma fruta está quase madura, ou quando dizemos que devemos pisar no ferio de um automóvel com uma força média. A obtenção de por exemplo, qual seria o valor da força para se pisar no freio do automóvel com uma força média representa uma incerteza pois este valor pode variar de pessoa para pessoa. Para eliminar esta imprecisão foram criados conjuntos *fuzzy* e a lógica *fuzzy*.

Segundo [3] a função triangular é o modelo computacional mais simples e é ilustrado na figura 2. O grau de pertinência, ou seja, o quanto o elemento pertence ao conjunto está representado pelo eixo y e o valor da variável em questão está no eixo x . A variável terá grau de pertinência máxima ao conjunto quando ela assumir o valor 1. Além deste conjunto existem outros, como por exemplo o *Singleton*.

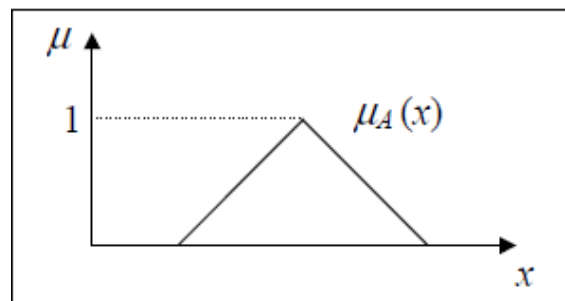


Figura 2: Conjunto Triangular.

Fonte: [3]

Segundo [3] o sistema fuzzy possui um conjunto de regras no formato *SE antecedente ENTÃO conseqüente*. Estas regras são utilizadas para realizar a inferência e informar um valor crisp para as variáveis de saída do sistema, ou seja, obter o valor de uma aceleração a ser aplicada por exemplo. Estas regras podem ser formadas através de um algoritmo de *Wang-Mendel*. Este algoritmo possui 5 passos os quais são descritos abaixo:

1. Divisão dos dados de entrada e de saída de conjuntos *fuzzy*;
2. Geração das regras *fuzzy*;

3. Definição de um grau para cada regra gerada;
4. Composição da base de conhecimento;
5. Defuzzificação (previsão de valores futuros);

Os quatro primeiros passos são responsáveis pela geração da base de conhecimento e compõem a chamada etapa de treinamento. O quinto passo realiza, a partir da base de conhecimento gerada, a previsão dos dados [1].

3 Implementação

A figura 3 mostra como foi implementado o projeto.

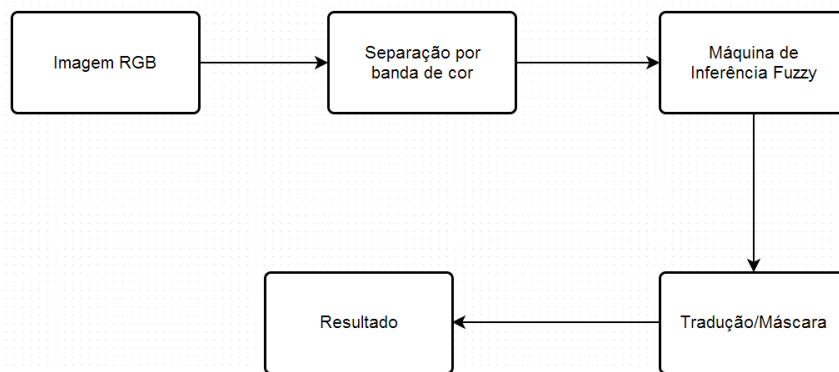


Figura 3: Diagrama de blocos do projeto

A seguir será descrita cada etapa do diagrama em detalhes:

3.1 Imagem RGB

Nesta etapa o sistema recebe uma imagem RGB para ser analisada. A imagem é guardada através das seguintes linhas de código:

```

% Carrega a imagem em um array
[rgbImage, storedColorMap] = imread(fullImageFileName);
[rows, columns, numberOfColorBands] = size(rgbImage);
  
```

Sendo 'fullImageFileName' o caminho para a imagem que deseja-se carregar

3.2 Separação por banda de cor

Nesta etapa separamos a matriz tridimensional da imagem em 3 matrizes separadas, uma representando cada banda de cor: R(vermelho), G(verde) e B(azul). Para tanto, usa-se as seguintes linhas de código:

```

% Extraindo R, G e B separadamente
rImage = rgbImage(:,:,1);
gImage = rgbImage(:,:,2);
bImage = rgbImage(:,:,3);
  
```

A figura 4 ilustra como o processo de separação por bandas de cor age na imagem. A imagem foi separada em bandas de RGB pois o *ColorMap* (mapa de cores) definido era o mapa de cores RGB.



Figura 4: Separação da imagem por banda de cores RGB

Esse processo é importante pois é através dele que serão obtidas as entradas para a máquina de inferência Fuzzy.

3.3 Máquina de Inferência Fuzzy

A Máquina de Inferência Fuzzy foi desenvolvida utilizando a toolbox de Fuzzy do próprio MATLAB. Nela pode-se definir toda a lógica a ser utilizada no projeto, que envolve a criação das variáveis linguísticas, as entradas, o método de inferência e as demais características da máquina de inferência fuzzy.

Em resumo o sistema fuzzy deste projeto é um sistema Mamdani com 3 entradas e 1 saída. Cada entrada representa uma banda de cor RGB e é dotada de 3 variáveis linguísticas: "pouco", "médio" e "muito" que denotam a intensidade da banda de cor em questão na imagem de entrada do sistema, são iguais no seu universo de discurso e são do tipo triangular. A saída é composta por 8 variáveis linguísticas que representam as cores: vermelho, verde, azul, magenta, ciano, amarelo, branco e preto. Adotou-se essa abordagem de modo a dar uma semântica para a saída crisp. Para o processo de defuzzificação foi utilizado o método do centro de gravidade.

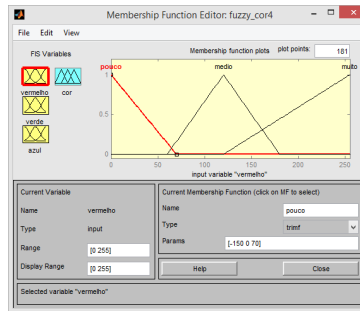
As figura 5 representa como ficaram configuradas as entradas e a saída do sistema fuzzy.

Como existem 3 entradas e 3 variáveis linguísticas, isso resulta em um conjunto de 27 possíveis combinações que serão as regras da Máquina de Inferência Fuzzy. A figura 6 mostra como ficaram configuradas as regras.

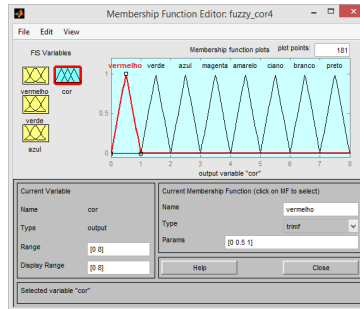
As regras foram configuradas de modo supervisionado, ou seja, com a ajuda de uma ferramenta de paleta de cores (<http://www.colorschemer.com/online.html>) usou-se valores para as bandas correspondentes aos intervalos (variáveis linguísticas) correspondentes a "pouco" (valor em torno de 60), "médio" (valor em torno de 100) e "muito" (valor em torno de 200) e de acordo com as combinações possíveis foi-se verificando qual cor dentre as 8 se encaixava melhor à cor obtida e com isso gerava-se a regra. Neste passo percebeu-se que a discretização em "pouco", "médio" e "muito" não era satisfatória visto que em uma mesma combinação, duas classes de cores distintas (mas próximas) eram possíveis. Isso foi percebido ao fazer o reconhecimento de cores em imagens com varias nuances de cor.

3.4 Tradução/Máscara

Com base nas saídas defuzzificadas, foram sendo preenchidas 8 matrizes de imagens correspondetes a cada com os valores 1 caso fosse a cor desejda ou com 0 quando não fosse a cor desejada, na posição do pixel referente a imagem orignal. Dessa forma ter-se-ia uma máscara que ao ser multiplicada à imagem original, deixaria apenas passar os pixels



(a) Configuração da entrada



(b) Configuração da saída

Figura 5: Configuração sistema fuzzy

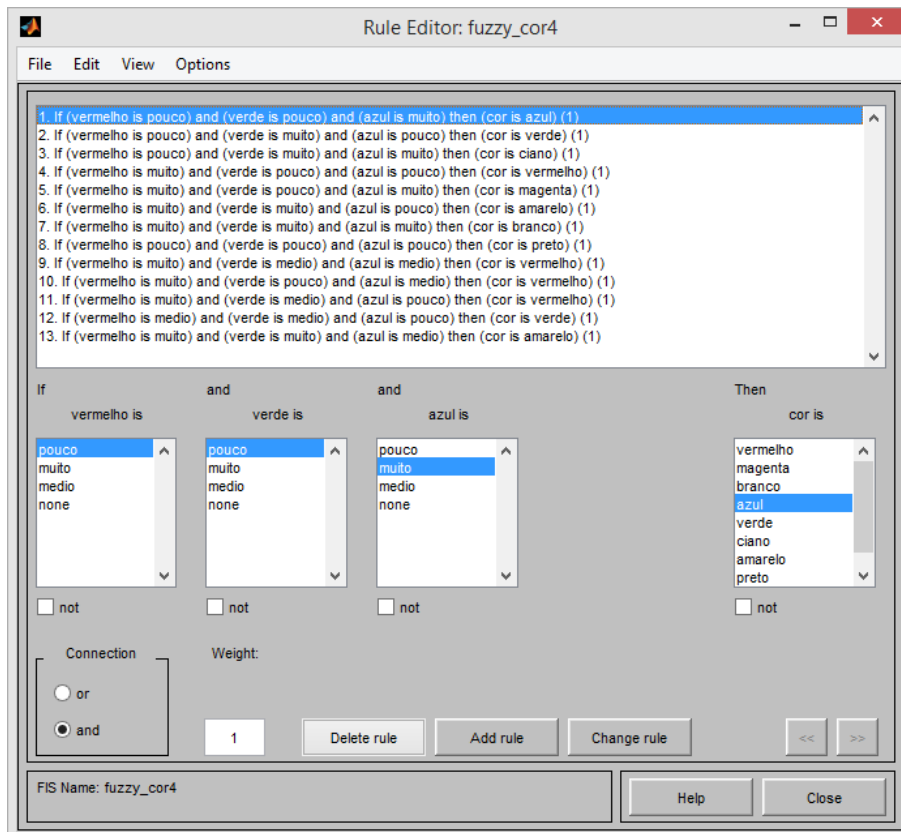


Figura 6: Regras fuzzy para reconhecimento de cores

referentes à cor desejada. Isso foi feito com a seguinte linha de código(para o caso da máscara de cor vermelha):

```
subplot(4,3,4), imshow(rgbImage.*repmat(rImage,[1,1,3])), title('Vermelho');
```

4 Resultados e conclusões

A primeira abordagem adotada foi utilizar apenas duas variáveis linguísticas para cada entrada ("pouco" e "muito"). Essa estratégia tinha o objetivo de reduzir o número de regras fuzzy e por consequência o erro por parte de treinamento humano das mesmas. Essa abordagem se apresentou muito ruim pois havia muitas possibilidades de cores dentro de um mesmo cluster fuzzy. A figura 7 mostra o resultado onde é possível perceber que há uma falta de classificação de cores principalmente no verde e no amarelo.

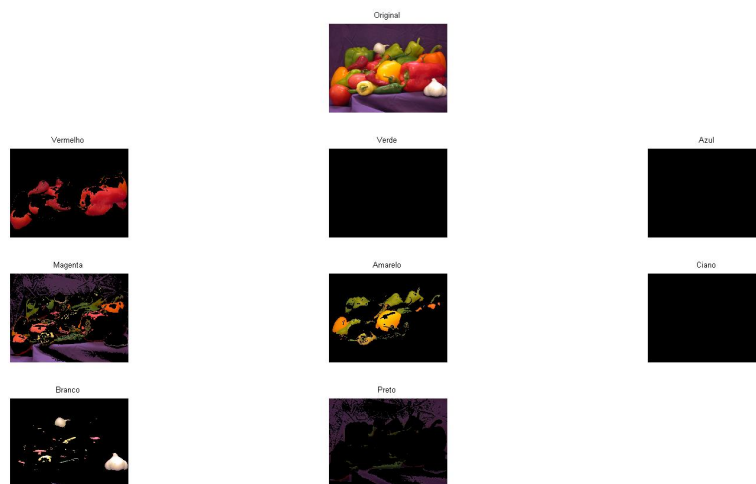


Figura 7: Resultado da execução da máquina de inferência fuzzy para reconhecimento de cores utilizando apenas duas variáveis linguísticas na entrada ("pouco" e "muito")

Posteriormente aumentou-se o número de variáveis linguísticas para 3 ("pouco", "médio" e "muito") e, portanto, a quantidade de regras subiu de 8 para 27. A figura 7 permite fazer um comparativo com a execução anterior. Neste caso já percebe-se uma boa melhora quanto à classificação/reconhecimento das cores (principalmente no verde no amarelo), mas ainda há confusões em algumas tonalidades próximas como vermelho e magenta, verde e amarelo. A classificação do azul apresenta um comportamento bem fora do esperado, mas com o tempo dedica a este projeto, não foi possível concluir a causa desse fenômeno

Como sugestões de melhorias pode-se levantar as seguintes:

- Usar um espaço de cores diferente de RGB, pois este não responde bem a variações de luminosidade.
- Usar uma abordagem Neuro-Fuzzy para treinar os clusters de cores das variáveis de entrada.
- Melhorar a modelagem fuzzy para obter uma defuzzificação mais performática bem como uma semântica melhor da utilizada para esse projeto

O projeto está disponível em: <http://johntheo.github.io/PDI/>

Referências

- [1] Claudio Vasconcelos Ribeiro, Ronaldo Ribeiro Goldschmidt, and Ricardo Choren. Métodos para previsão de séries temporais e suas tendências de desenvolvimento. Série Monografias em Sistemas e Computação - Instituto Militar de Engenharia, Setembro 2009. Artigo. ISSN: 1982-9035.
- [2] Kenneth R. Sloan, Jr. and Christopher M. Brown. Color map techniques. 10(4):297–317, August 1979.
- [3] Ricardo Tanscheit. Sistemas fuzzy. Departamento de Engenharia Elétrica - Pontifícia Universidade Católica do Rio de Janeiro, 2011. Notas de Aula.

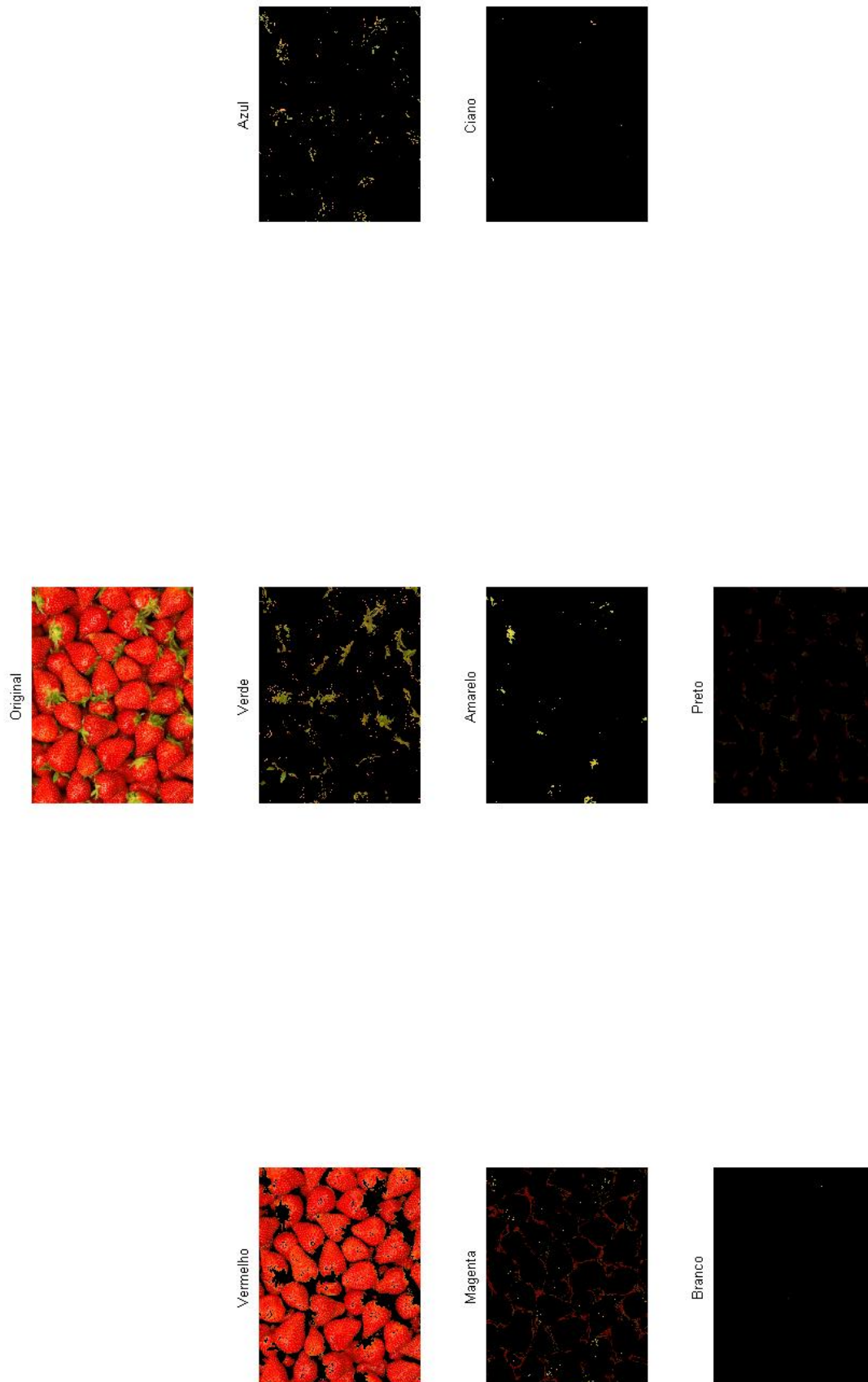


Figura 8: Resultado da execução da máquina de inferência fuzzy para reconhecimento de cores utilizando três variáveis linguísticas na entrada ("pouco", "médio" e "muito")

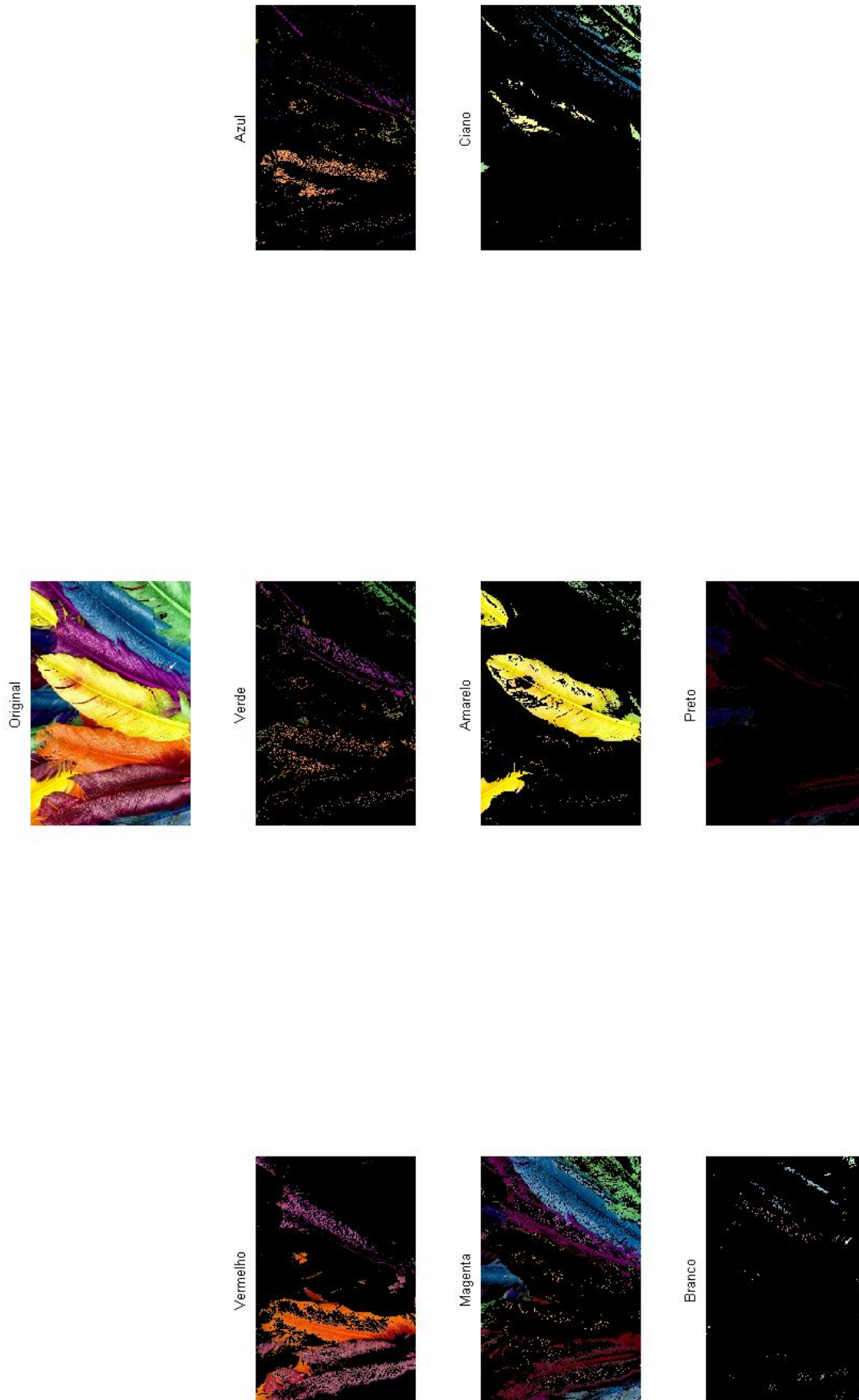


Figura 9: Resultado da execução da máquina de inferência fuzzy para reconhecimento de cores utilizando três variáveis linguísticas na entrada ("pouco", "médio" e "muito")

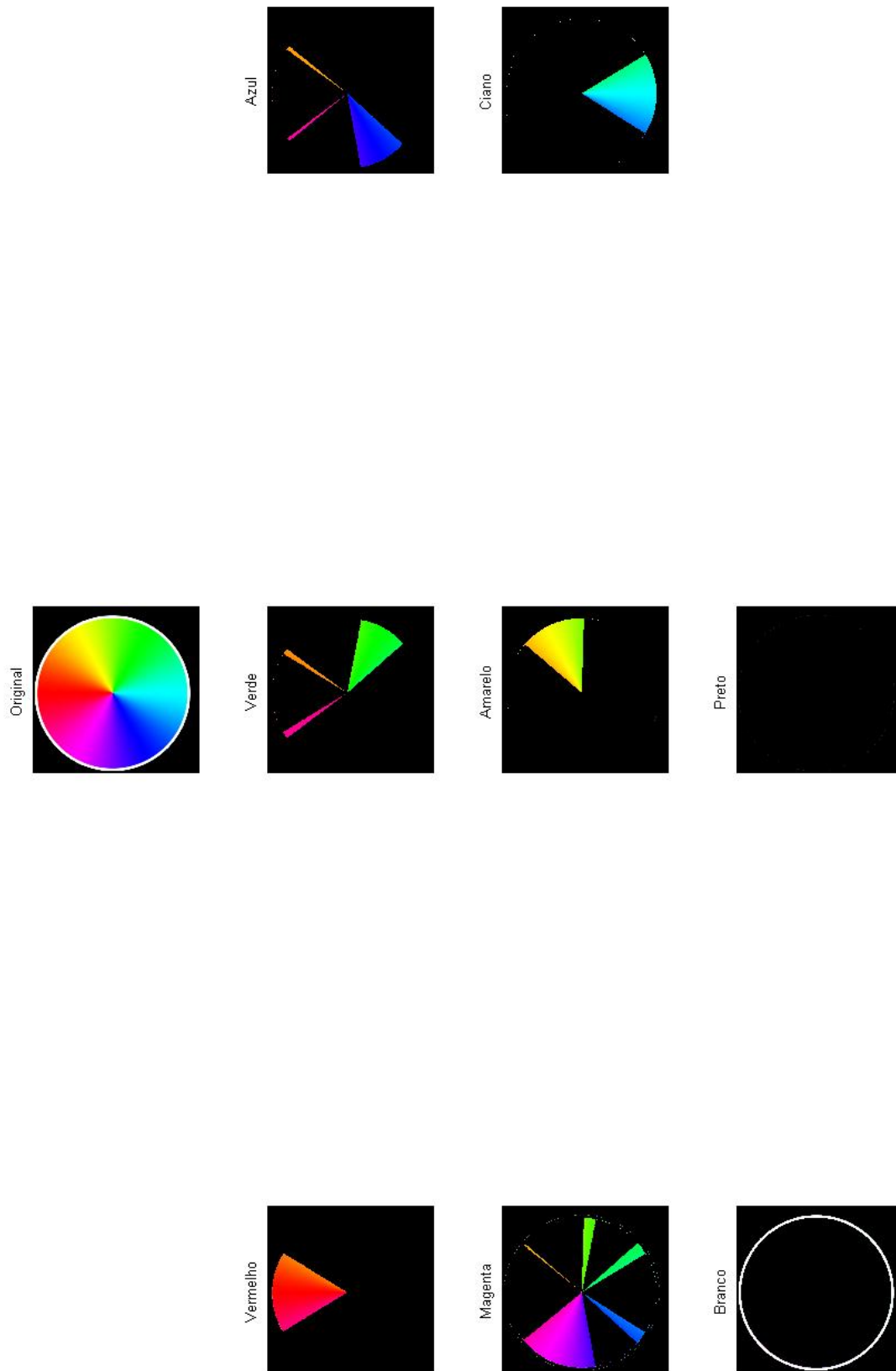


Figura 10: Resultado da execução da máquina de inferência fuzzy para reconhecimento de cores utilizando três variáveis linguísticas na entrada ("pouco", "médio" e "muito")

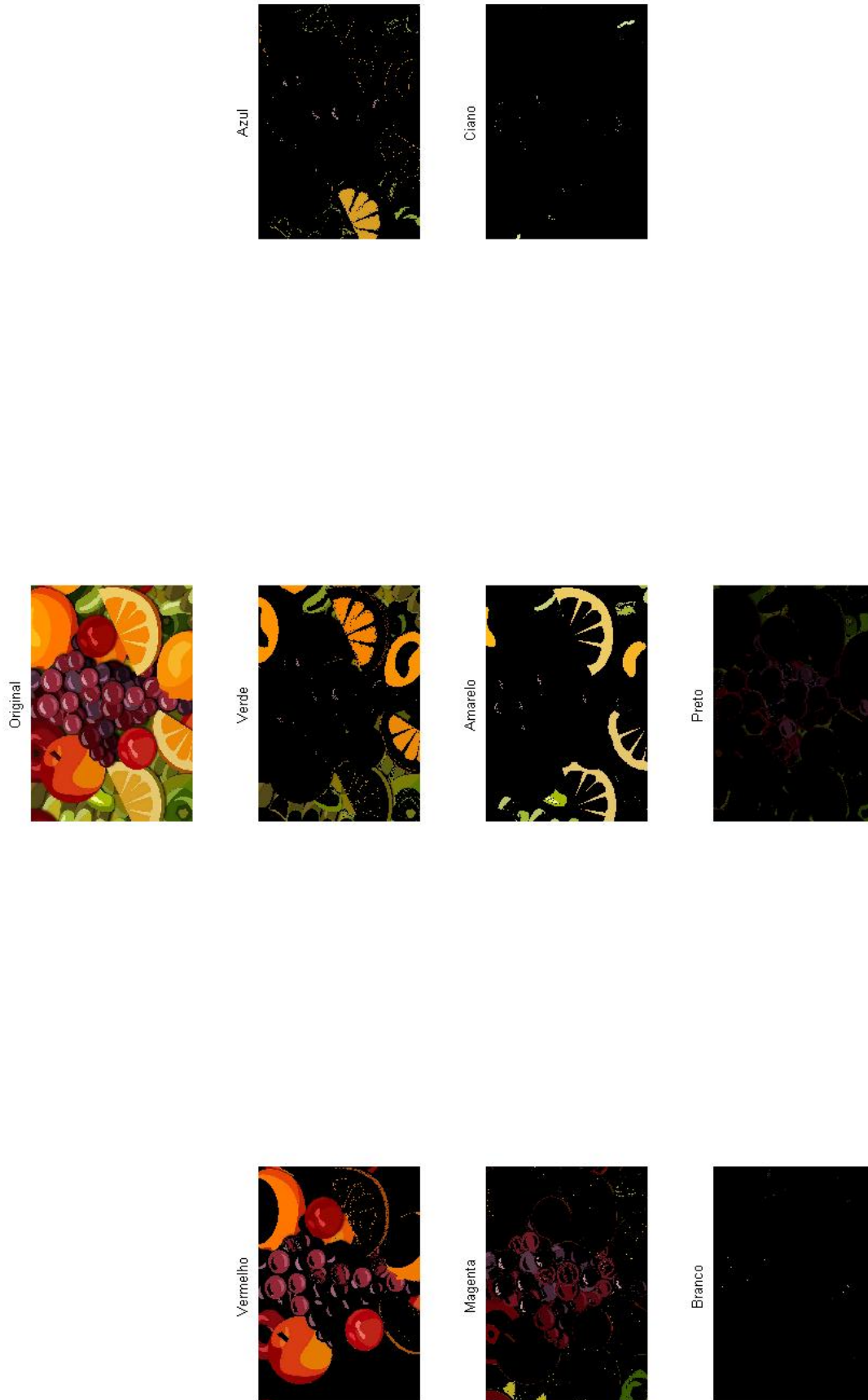


Figura 11: Resultado da execução da máquina de inferência fuzzy para reconhecimento de cores utilizando três variáveis linguísticas na entrada ("pouco", "médio" e "muito")



Figura 12: Resultado da execução da máquina de inferência fuzzy para reconhecimento de cores utilizando três variáveis linguísticas na entrada ("pouco", "médio" e "muito")

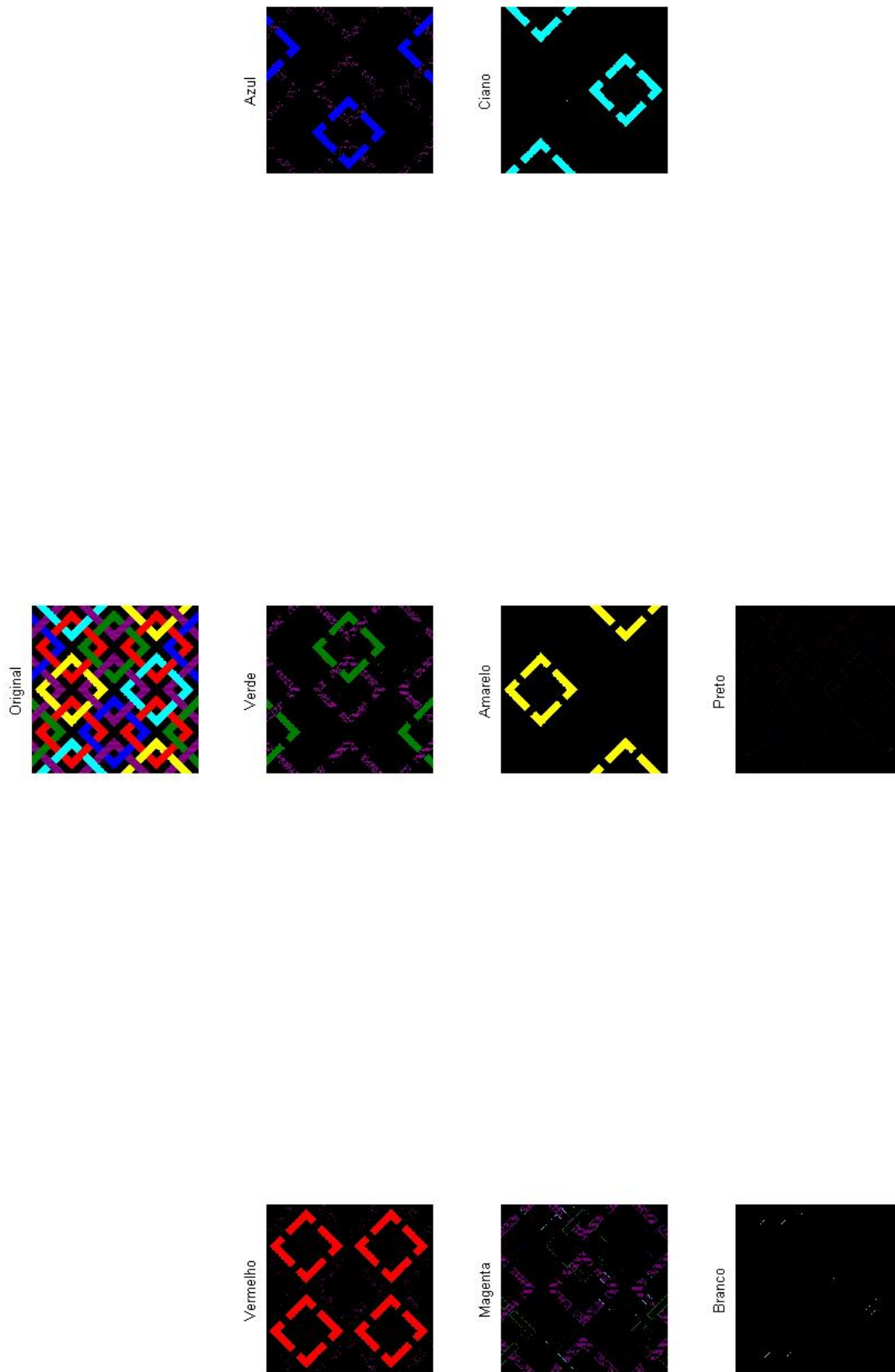


Figura 13: Resultado da execução da máquina de inferência fuzzy para reconhecimento de cores utilizando três variáveis linguísticas na entrada ("pouco", "médio" e "muito")

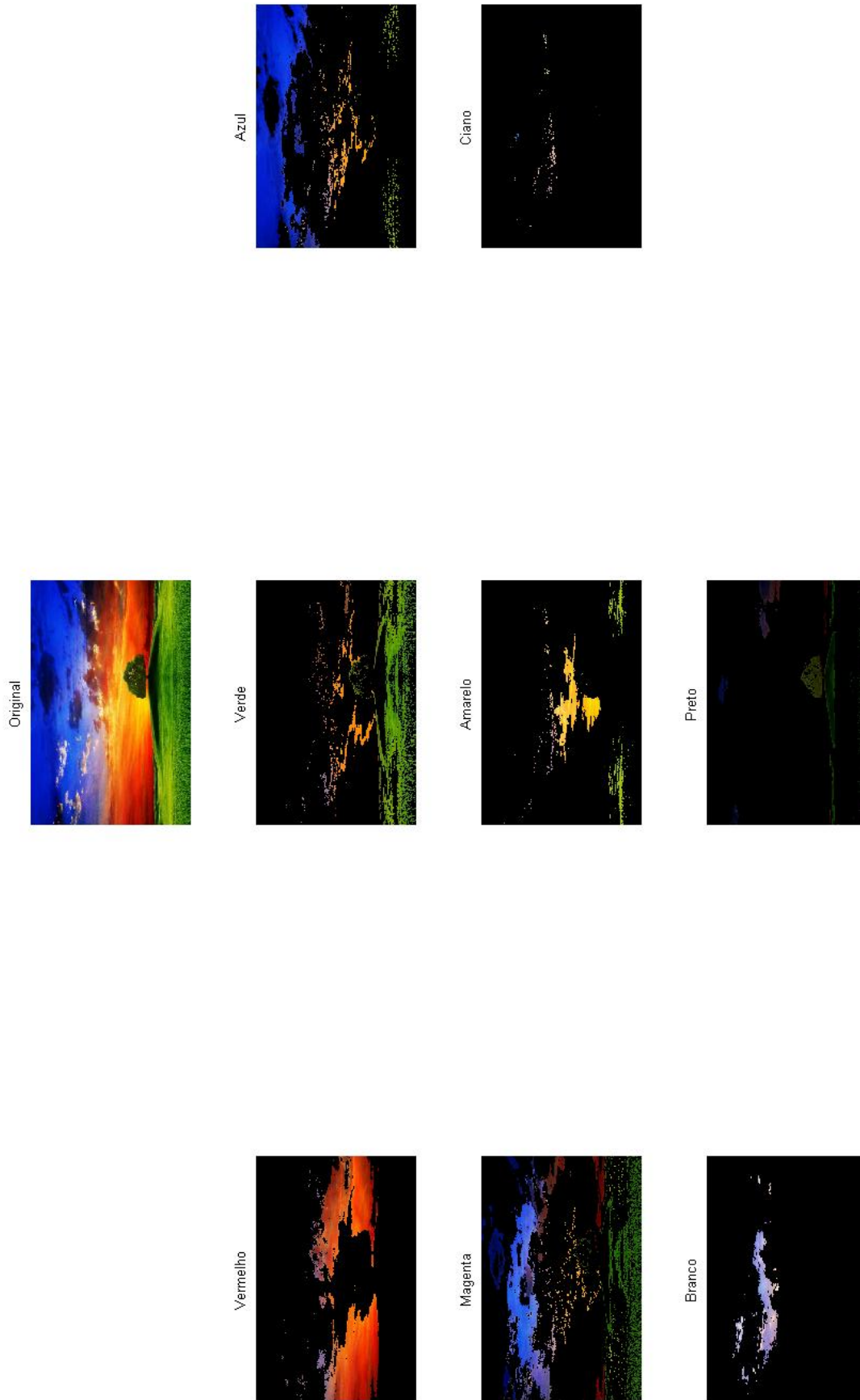


Figura 14: Resultado da execução da máquina de inferência fuzzy para reconhecimento de cores utilizando três variáveis linguísticas na entrada ("pouco", "médio" e "muito")