

Instruções

Execute `main.py` para executar o programa (`./main.py`). Você pode enviar o nome da imagem a ser processada pela linha de comando (`./main.py simple.png`), caso não seja passado nenhum argumento, será assumido como argumento a “`./double.png`”.

Assim que o programa inicia você tem a opção de decidir qual algoritmo usar. Após processada a imagem de treino, será mostrado o número de classificações corretas (hit) e o número de classificações incorretas (miss), e então será gerada uma imagem com 3000 pontos aleatórios. Sem seguida é perguntado se você quer que o programa tente fazer uma imagem com pixels em ordem sequencial, pintando assim todos os pixels da imagem, formando uma imagem de alta qualidade, no entanto este processo pode ser lento (cerca de 1 minuto).

Resultados e Conclusões

Os algoritmos IBL 1 e 2 mostraram uma capacidade surpreendente visto sua simplicidade de implementação, com poucas linhas de código foi possível de se facilmente classificar uma espiral, e com um desempenho também bastante bom.

Entre o IBL 1 e 2 só existe uma única linha de código diferente, e esta pequena modificação já faz mudanças profundas na maneira como é armazenado o Descritor Conceitual, tornando-o substancialmente menor, fazendo com que ele ocupe menos memória e também execute mais rápido (já que na classificação, cada ponto processado é comparado com todos do descritor conceitual). No entanto ele dá como output uma imagem um pouco mais distorcida do que o IBL 1, mas ainda gera uma imagem suficientemente boa.

Já o IBL 3 mostrou-se desafiante na sua implementação, sendo ele bem maior que suas versões anteriores. No entanto seu Descritor Conceitual pode vir a ser ainda menor do que o do IBL 2, e ainda sim ser mais consistente que o IBL 2. Isto acontece por que o IBL 3 tem um melhor critério de avaliação sobre quais pontos realmente são importantes para ser usados como

classificadores e quais são ruins (tanto por já existirem vizinhos melhores tanto quando é o caso deste determinado ponto ser ruído).

Apesar do algoritmo de treino ser mais complexo, tanto na implementação, tanto no desempenho, o algoritmo de classificação ainda é o mesmo, portanto o tempo extra despendido com o treino pode compensar bastante na hora da classificação.

Fiz com que um ponto não pudesse ser eliminado do descritor conceitual sem antes ter sido usado para avaliação pelo menos 30 vezes. Isto se deve ao fato de que no começo do treinamento, a maior parte dos pontos aparenta ser um "mal classificador", e pontos que poderiam ser bons futuramente acabariam sendo excluídos do DC prematuramente.

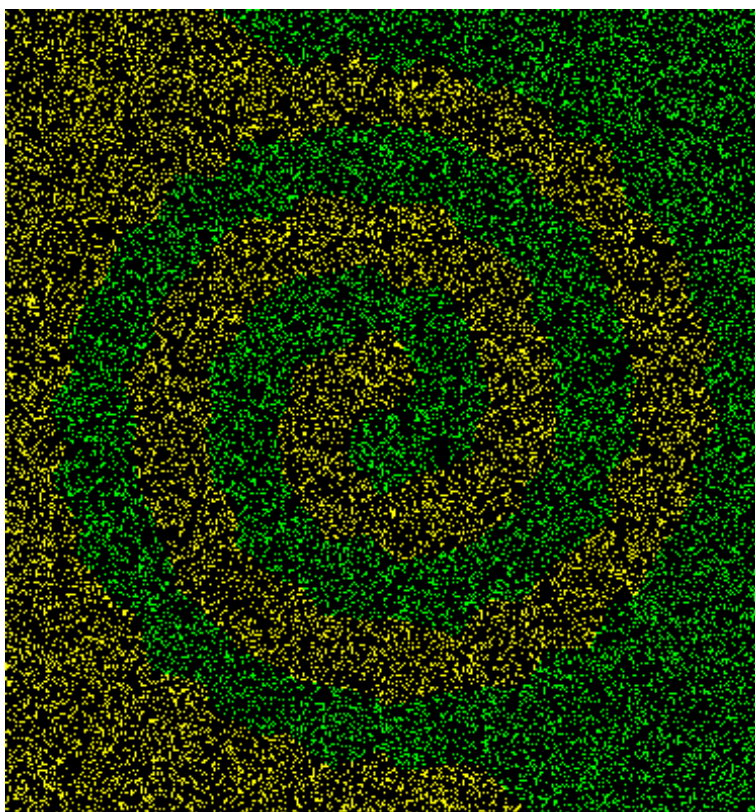


Figura 1: Imagem gerada com 3000 pontos aleatórios utilizando o IBL 1.



Figura 2: Imagem gerada pintando todos os pixels da imagem utilizando o IBL 1.

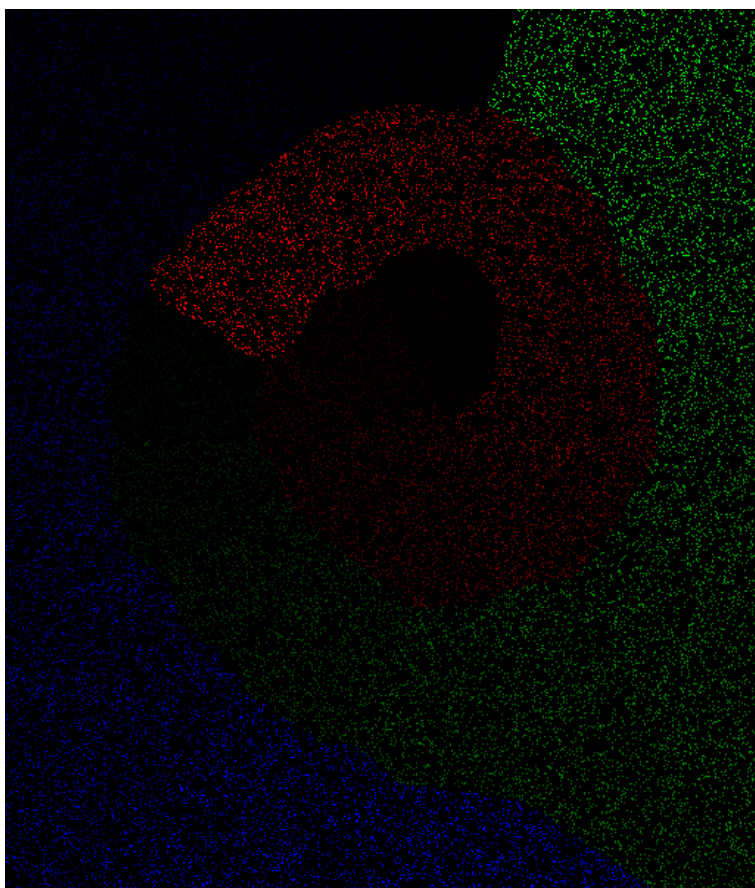


Figura 3: Imagem gerada com 3000 pontos aleatórios utilizando o IBL 2.

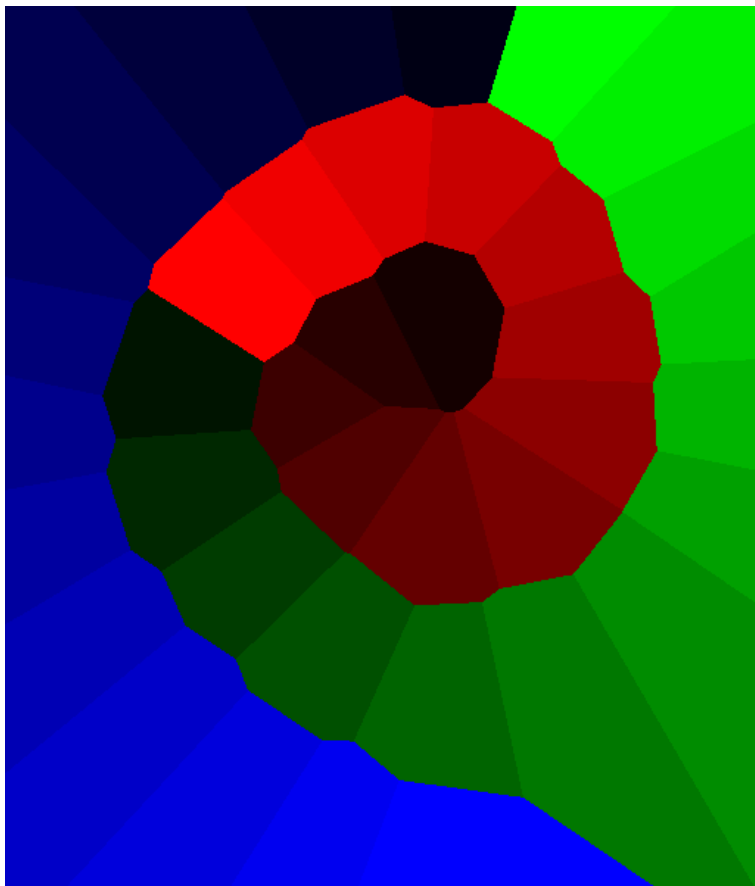


Figura 4: Imagem gerada pintando todos os pixels da imagem utilizando o IBL 2.