Aluno: Halyson Ricardo Rabuski Junior Ano: 2021

MÉTODO MULTIPROCESSADO PARA FILTRO DE BORDAS ATRAVÉS DE CONVOLUÇÃO DE ARRAYS 2D

METODOLOGIA

Para aplicar um filtro em uma imagem através do método de convolução, é necessário obter um array 2d com os dados da imagem. Com o array 2d obtido, é possível através do processo de convolução de matrizes com outro array 2d de valores específicos, aplicar um filtro sobre a imagem. A figura 1 representa a matriz com os valores aplicados na pesquisa, responsável por aplicar um filtro de bordas na imagem escolhida, esse filtro é também chamado de kernel.

Figura 1 - Kernel de filtro de bordas

```
kernel = [[-1,-1,-1],
          [-1,8,-1],    #Bordas 2
          [-1,-1,-1]]
```

O filtro demonstrado acima, é um filtro para detecção de bordas, que é capaz de determinar os pontos em uma imagem onde a intensidade luminosa muda de forma repentina. É um filtro utilizado na área de extração de características, pois ao ser aplicado, diminui a quantidade de dados a serem processados, de modo que os elementos da imagem continuam sendo reconhecidos, mas as propriedades menos relevantes são descartadas.

Existem dois tipos de processo presentes na pesquisa, o master e os workers. O master é responsável por inicializar os valores, preparar o ambiente para o multiprocessamento e obter os resultados dos workers, transformando em uma imagem novamente, que dessa vez estará com filtro de bordas aplicado. Neste trabalho, foi utilizada a linguagem de programação python com a biblioteca nativa para multiprocessamento, o multiprocessing.

Após fazer a configuração inicial é separado uma carga de trabalho para cada worker. Isso é feito analisando a quantidade total de iterações necessárias para aplicar totalmente o filtro na imagem e esse valor é dividido por 8, pois nesse caso foram utilizadas 8 threads diferentes, que é o limite máximo do hardware nesse caso.

Os workers são responsáveis por obter os valores vizinhos a um determinado valor dentro da matriz da imagem e aplicar a expressão de convolução, retornando um resultado para a master. O processo é balanceado pelo método comentado acima, e a carga é semelhante entre os workers, de modo que todos realizam a mesma quantidade de tarefas iguais.

A figura 2 representa uma imagem antes da mesma passar pelo sistema.

Figura 2 - Imagem antes de passar pelo filtro de bordas

A figura 3, representa a imagem da figura 2 após passar pelo filtro de bordas aplicado nesta pesquisa. É possível verificar que os elementos presentes na mesma continuam sendo reconhecidos, mas a quantidade de informações é menor.

Figura 3 - Imagem após passar pelo filtro de bordas