2️⃣

Missão OpenCV

Caio Cesar Vieira Cavalcanti (Ciência da Computação)

→ Visão Computacional

⇒ Objetivos:

- Aprender o básico sobre uma biblioteca de visão computacional. Entenda sobre os conceitos básico para trabalhar com imagens e como realizar manipulações de pixel.

⇒ Matérias necessários:

- Python3
- Biblioteca Numpy
- Biblioteca OpenCV:
- Windows 10 ou Linux

⇒ Atividade:

- Ler e executar os exemplos dos capítulos 1 e 2 da apostila *"Introdução a Visão Computacional com Python e OpenCV";*
- Enviar os códigos(.py) comentados;
- Escrever um breve relatório sobre a missão.

⇒ Relatório

A visão computacional (uma das áreas que pretendo atuar), como dito na apostila *"Introdução a Visão Computacional com Python e OpenCV"*, é como a máquina e a tecnologia enxergam o mundo real, desenvolvendo teorias e construção para a criação de sistemas artificiais para obter informação necessária de imagens ou quaisquer dados multidimensionais. Um fator importante é que essa área está em constante evolução, sempre inovando e descobrindo novos algoritmos para o reconhecimento de padrões de dados, mas não necessariamente de processamento de imagens. Um exemplo de empresas que utilizam a visão computacional, é o Facebook, com a função de analisar uma foto, e por ela, ser reconhecido objetos automaticamente, até mesmo os rostos de pessoas (se houver na foto) para serem marcadas posteriormente. Outro detalhe abordado na apostila, pela opinião do autor, é a diferença entre filtro de imagem, e o reconhecimento de objetos, dado uma imagem, uma se trata de um processamento de filtragem casual, e o outro da própria visão computacional.

Agora comentar acerca dos exemplos de códigos envolvendo a visão computacional, é uma experiência nova para mim, trabalhar com as bibliotecas de Python mais elaboradas, como NumPy, e OpenCV, contudo amei ter esse contato, poder trabalhar realmente com esses tipos de manipulações, que acontecem no meio profissional, e adquirir mais conhecimento desses assuntos, ou de pelo menos despertar a curiosidade de ir mais afundo. Consegui compreender o funcionamento de algumas funções do OpenCV, como de leitura de imagens (imread()), de escrita (imwrite()), dentre outras, além disso, aumentar o entendimento do funcionamento das cores, e dos pixels, no qual uma imagem é formada por uma matriz de 3 dimensões ou canais, que formam as 3 cores primárias RGB (red - vermelho, green - verde, blue - azul), em que cada célula da matriz, uma linha e coluna, corresponde a um pixel, que nos exemplos podem conter um inteiro de 8 bits sem sinal (unsigned int), ou seja, um valor entre 0 e 255 inclusive. Dessa forma, um pixel pode ser formado por um tupla de 3 inteiros sem sinais de 8 bits (RGB), resultando através de uma composição entre os pixels, e manipulação desses valores da tupla, numa imagem colorida como conhecemos e enxergamos. Dentre as combinações possíveis desses valores, temos um número exorbitante de 16,7 milhões de possibilidades de cores.

Para finalizar, é abordado no capítulo 2 o sistema de coordenadas e exemplos de manipulações diretas dos valores de cada pixel que constitui uma imagem. É apresentado que o pixel mais a esquerda e acima, possui a sua posição (0, 0), em uma matriz com linha e coluna de uma dada imagem, e o pixel mais a direita e abaixo com sua posição sendo o tamanho da linha e coluna, subtraídos por 1 (pois esses valores indicam os índices, começando do 0 e seu máximo sempre sendo tamanho - 1). Há diferentes formas de acessar esses pixels, sendo individualmente, de maneira bem eficaz pois o acesso é com base O(1) (complexidade de algoritmo, de acordo com as listas de Python, ou tende a O(1)), ou realizando uma "varredura" entre os pixels de uma matriz, passando para cada linha, as colunas, contudo por haver um laço interno ao outro, dependendo do tamanho dessa matriz, o processo se torna cada vez mais lento e não performático (uma vez que posso no pior caso, obter uma complexidade ou eficiência de O(n^2) com "n" o tamanho da matriz). Por fim, é analisado aos arquivos .py os exemplos passados pelo livro, com manipulações de diferentes formas ou estratégias de modificar um pixel em específico, seja utilizando módulo (resto da divisão entre dois valores) ou saltando a cada valor de linha e coluna, e preenchendo a imagem original com alguma forma ou objeto de outra cor por exemplo, como mostra o último caso do capítulo.
