Processamento de imagem

|  |  |
| --- | --- |
| Nomes: | RGMs: |
| Eduarda Fernandes | 29204356 |
| Davi Santos de Andrade | 31075550 |
| Johnatan Caetano | 30087155 |
| Everman | 30333717 |
| Daniel Medeiros | 29381169 |

* Objetivos:
  + Implementação de filtros que retirem um ou mais canais de cor de uma imagem dependendo do sistema de cores que esteja sendo aplicado.
  + Implementação do filtro Blur, mais qualquer outro filtro convencional.
  + Implementação de uma interface para execução das etapas em qualquer ordem podendo aproveitar o resultado de uma operação anterior para executar um novo processamento de imagem.
* Equipamentos/ferramentas utilizados:

Para desenvolver esta aplicação conforme descrito anteriormente, utilizamos um notebook Samsung Core i5 de décima geração como plataforma principal. Optamos pela IDE chamada Visual Studio Code devido à sua interface intuitiva e fácil manipulação. A linguagem de programação escolhida foi Javascript, devido à sua versatilidade e eficiência. Para garantir uma organização adequada do projeto, decidimos utilizar o GitHub para armazenar e colaborar no código, permitindo que todos os membros do grupo pudessem contribuir de forma eficiente.

* Procedimento experimental:

Para cumprir o primeiro objetivo descrito anteriormente, começamos selecionando elementos do DOM (Document Object Model) e definindo algumas variáveis e arrays para manipular esses elementos posteriormente.

Texto

Descrição gerada automaticamente

Na segunda parte do código, desenvolvemos um objeto CORES, que mapeia identificadores de cores (números de 1 a 20) para suas respectivas informações, incluindo o nome da cor e seu valor RGB (Red, Green, Blue). O objeto CORES pode ser utilizado para diversas finalidades, como atribuir cores específicas a elementos de uma interface, aplicar filtros de cor, ou realizar outras operações relacionadas a cores em um contexto de manipulação de imagens.

Texto

Descrição gerada automaticamente

No terceiro trecho do código, elaboramos duas funções: enableButtons e disableButtons, que são responsáveis por habilitar e desabilitar um conjunto de botões na interface do usuário.

Texto

Descrição gerada automaticamente

Logo após no quarto trecho do código adicionamos ouvintes de eventos (event listeners) para manipular o comportamento de arrastar e soltar (drag and drop) uma imagem sobre um elemento da interface do usuário identificado pela classe picture.

Texto

Descrição gerada automaticamente

O quinto trecho do código, trata o evento de soltar um arquivo de imagem no elemento picture. Quando uma imagem é solta, ela é lida e exibida na página. Durante o processo, o código fornece feedback visual ao usuário (adicionando e removendo a classe dragover), manipula o conteúdo do elemento pictureImage, e habilita botões de ação que permitem modificar a imagem carregada. Se nenhum arquivo for solto, o texto padrão é exibido na área da imagem:

Texto

Descrição gerada automaticamente

Logo após, o código trata o evento de mudança no campo de entrada de arquivo inputFile, que ocorre quando o usuário seleciona um arquivo para carregar. Quando um arquivo é selecionado, ele é lido e exibido na página. Durante o processo, o código verifica se há um arquivo selecionado, manipula o conteúdo do elemento pictureImage, e habilita botões de ação que permitem modificar a imagem carregada. Se nenhum arquivo for selecionado, o texto padrão é exibido na área da imagem:

Texto

Descrição gerada automaticamente

Na próxima etapa, o código permite que o usuário converta uma imagem carregada na página em um arquivo Excel (.xlsx). Quando o botão convertToExcelBtn é clicado, a imagem é desenhada em um canvas, os dados da imagem são extraídos e convertidos em uma matriz, essa matriz é transformada em uma planilha do Excel usando a biblioteca SheetJS (XLSX), e finalmente um arquivo Excel é gerado e disponibilizado para download. Se nenhuma imagem for carregada, um alerta é exibido pedindo ao usuário para carregar uma imagem antes de continuar:  
  
Texto

Descrição gerada automaticamente

Essa próxima função é útil ao trabalhar com operações que envolvem manipulação de arquivos binários em JavaScript, como a criação de arquivos Excel, manipulação de imagens, entre outros. Ao receber uma string como entrada, ela converte essa string em um ArrayBuffer contendo os bytes correspondentes aos caracteres da string. Isso é frequentemente usado em conjunto com APIs que manipulam dados binários, onde é necessário representar os dados de uma forma específica para serem processados corretamente.

Texto

Descrição gerada automaticamente

Na próxima etapa, criamos um código que permite que o usuário modifique as cores de uma imagem carregada na página. Quando o botão modifyColorsBtn é clicado, o usuário pode escolher uma nova cor e todas as áreas da imagem que têm a mesma cor que um pixel aleatório selecionado anteriormente serão substituídas pela nova cor selecionada. Este processo é realizado no canvas, e a imagem modificada é exibida ao usuário. Se nenhuma imagem estiver carregada, um alerta é exibido pedindo ao usuário para carregar uma imagem antes de continuar.

Texto

Descrição gerada automaticamente

Texto

Descrição gerada automaticamente

Logo após elaboramos uma função para pegar a cor de cada pixel da imagem e converter de RGB para CMYK.  
  
Texto

Descrição gerada automaticamente

Esse próximo código permite que o usuário converta uma imagem carregada para o formato CMYK e faça o download da imagem resultante. Quando o botão convertToCMYKBtn é clicado, a imagem é convertida de RGB para CMYK usando a função rgbToCmyk, e a imagem CMYK resultante é disponibilizada para download como um arquivo PNG. Se nenhuma imagem estiver carregada, um alerta é exibido pedindo ao usuário para carregar uma imagem antes de continuar.

Texto

Descrição gerada automaticamente

Nas próximas sessões, definimos as funções que convertem a imagem em escala de cinza, seguindo os conceitos apresentados em sala:

* Luminosidade:

Texto

Descrição gerada automaticamente

* Média ponderada:

Texto

Descrição gerada automaticamente

* Dessaturação:

Texto

Descrição gerada automaticamente

* Valor máximo:

Texto

Descrição gerada automaticamente

* Valor mínimo:

Texto

Descrição gerada automaticamente

Após essa definição, foi desenvolvido um método para o usuário selecionar qual o tipo de conversão deseja:

Texto

Descrição gerada automaticamente

Texto

Descrição gerada automaticamente

* Análise de resultados:

A próxima função é útil quando é necessário aumentar o contraste das cores em uma imagem. Ela opera diretamente nos dados de imagem (na forma de um objeto ImageData) e aumenta o contraste aplicando uma transformação aos valores RGB de cada pixel. O fator de contraste determina a intensidade do aumento do contraste. O resultado é uma imagem com cores mais vívidas e distintas.

Tela de computador com texto preto sobre fundo branco

Descrição gerada automaticamente

Este próximo trecho de código permite ao usuário aumentar o contraste das cores em uma imagem carregada, solicitando um fator de contraste e aplicando-o à imagem. A imagem resultante, com contraste aumentado, é disponibilizada para download como um arquivo PNG. Se nenhuma imagem estiver carregada, um alerta é exibido pedindo ao usuário para carregar uma imagem antes de continuar.

Texto

Descrição gerada automaticamente

Na próxima etapa foi definido um método que aumenta o contraste nos tons de cinza:

Texto

Descrição gerada automaticamente

Na próxima etapa foi definido um método que aplica um filtro de remoção de cor na imagem:

Texto

Descrição gerada automaticamente

Logo em seguida foi definido as ações dos botões que referenciam as cores que serão retiradas:

Texto

Descrição gerada automaticamente

Texto

Descrição gerada automaticamenteLogo após, a lógica que traz o filtro Blur a imagem que traz o efeito de desfoque:

Por último foi definido a lógica que armazena a imagem original e carrega a imagem:

Texto

Descrição gerada automaticamente

* Análise de resultados:

A análise do resultado mostra que existem diversas formas de se manipular uma imagem, seja alterar a cor, o contraste, ou adicionar algum filtro, e através deste trabalho foi possível identificar toda a lógica e ver na prática todo esse processo acontecendo através da programação, e após a implementação foi possível identificar como a imagem é manipulada, e mudada desde a cor, até mesmo a forma de se enxergar a mesma.

* Considerações finais:

Uma nova etapa foi integrada ao projeto. Agora, conseguimos reunir todo o conhecimento adquirido ao longo do curso, além de implementar uma interface para facilitar o uso pelo usuário do sistema. É uma etapa interessante, pois todo o aprendizado foi colocado a prova, e com certeza todo o grupo absorveu bastante informação nova e todos poderão ver as imagens com outros olhos.