Toonify: Cartoon Photo Effect Application with Python and OpenCV

Estevam Albuquerque
Universidade de Brasília
Departamento de Ciência da Computação
Brasília, Brasil
estevamgalvao@aluno.unb.br

Marcos Paulo Diniz
Universidade de Brasília
Departamento de Ciência da Computação
Brasília, Brasil
marcosdiniz@aluno.unb.br

***Resumo*—O filtro Toonify surge com mais uma opção aos usuários de transformar suas fotos em cartoons, fazendo isso de forma rápida e fácil, para se alcaçar isso foi usado o Python 3 e a biblioteca OpenCV.**

I. Introdução

Levando em conta a crescente popularização dos *smartphones* nos últimos anos, em especial no Brasil, cresce concomitantemente os registros fotográficos, tendo em vista que não é mais necessário uma maquina fotográfica para capturar momentos.

Com o aumento do número de registros de momentos, provocou-se uma necessidade de criar novas soluções para edição de fotos de forma simples e rápida. Com isso, o Toonify aparece como um filtro que dá novas possibilidades ao receber uma foto, transformando a imagem original em uma imagem cartoon.

E para a mensuração das imagens geradas, elas foram submetidas em um questonário para ter avaliação do público, sendo o objetivo inicial conseguir uma aceitação média de 7 pontos em um universo de 10 pontos possíveis.

II. Cenário e Trabalhos Relacionados

Com o aumento do uso de aplicativos de celulares que visam dar a possibilidade do usuário aplicar filtros divertidos às suas imagens, para se fugir de um registro normal. O filtro de cartoon se apresenta como uma possibilidade gerar uma imagem em cartoon de um registro que o próprio usuário tenha em seu dispositivo.

A priori, o Toonify se apresenta em uma solução implementada em Python com auxílio da biblioteca OpenCV, para manipular as imagens. Porém há a possibilidade de expandir para o uso web ou para dispositivos móveis, como proposto por Kevin Dade [1].

III. Solução Proposta

O processo para cartoonizar uma imagem é dividido em quatro etapas principais, sendo todas elas importantes para no final gerar uma boa imagem. A seguir será mostrado detalhes de cada uma dessas etapas.

A. Median Filter

O *Median Filter* é o primeiro a ser aplicado na imagem, sendo ele de fundamental importância tendo em vista que com ele consegue-se obter a imagem sem *noise* - isto é, sem ruído - sem , que será usada para a detecção das bordas e, posteriormente, para a quantização das cores. E com essas duas saídas obtém-se a imagem cartoonizada.

Para filtrar a imagem e retirar os ruídos, aplicamos o *Median Filter* com tamanho 7x7, com isso para cada pixel será analisado os 48 pixels que estão ao seu redor e o pixel em questão, totalizando 49 pixels. A partir dos valores dos 49 pixels pega-se o valor da mediana e troca o valor do pixel em questão pelo valor da mediana encontrado, realizando esse procedimento para todos os pixels da imagem.

Como pode ser visto no exemplo abaixo, após a passagem do filtro, a imagem fica suavisada, causando uma perda de detalhes e principalmente de *noise*, para o filtro não causar a perda de informações importantes na imagem e só retirar o noise, deve-se ajustar o tamanho do filtro que se passa na imagem. A imagem [b] é o resultado da imagem [a] após o *Median Filter* com *kernel* de tamanho 7x7.

Figura 1. [a] Imagem original [b] Imagem após o *Median Filter*

B. Detecção de Bordas

A Detecção de Bordas indubitavelmente também é de fundamental importância para se ter um bom efeito cartoon, pois ao salientar as bordas de uma imagem fica mais notável o efeito e com isso se aproxima mais da realidade de um desenho em cartoon, já que nas etapas de desenhar uma imagem em cartoon a primeira etapa é denhar as bordas da imagem em preto, ou seja, para retratar uma imagem normal em cartoon é fundamental adicionar as bordas da imagem.

A borda pode ser considera a fronteira entre duas regiões, ou seja, o que delimita partes da imagem, geralmente marcada pela troca de luminosidade. Sabendo da importância da detecção das bordas para se obter uma imagem cartoonizada, usamos o método de extração de bordas de *Canny*, por ser o mais eficiênte disponível, sendo bem tolerante à ruídos e ao mesmo tempo identifica as bordas com poucos erros.

Como citado antteriormente, um ponto fundamental da cartoonização de imagens é salientar as bordas das imagens em cor preta, porém ao usar a detecção de bordas de *Canny*, a imagem resultante será uma de fundo preto com as bordas em branco. Com isso, uma etapa importante da detecção das bordas é a inversão das cores da imagem gerada pela função *Canny*, resultando uma imagem como o exemplo abaixo.

Figura 2. [a] Imagem original [b] Bordas da imagem original

C. Bilateral Filter

O *Bilateral Filter* é um filtro de suavização, que preserva as bordas e borra as demais regiões, ou seja, causa a perda de detalhes na imagem, porém sem perder ou danificar as fronteiras da imagem.

Para se conseguir isso, cada pixel da imagem é substituído pela média ponderada, bem similar ao filtro de Gauss, se diferenciando que o de Gauss leva em consideração apenas a distência euclidiana dos pixels, enquanto o *Bilateral Filter* também leva em consideração outros aspectos. Como, entre outros, intensidade de cor e distância de profundiade.

A preservação das bordas e a suavização de detalher causadas pelo *Bilateral Filter* podem ser notadas no exemplo abaixo.

Figura 3. [a] Imagem original [b] Imagem após o *Bilateral Filter*

D. Quantização de Cores

Quantização de Cores é o processo que reduz o número de cores usadas em uma imagem, porém preservando a informação contida na imagem, ou seja, as cores de tons parecidos se reduzem a um tom similar. O que é fundamental para a representação de uma imagem cartoonizada, já que uma imagem de cartoon é marcada pela presença de poucas cores e pouco gradiente nas trasnsições de cores.

O efeito da quantização se limita e pagar tons similares e unficálos em um só, até um limite k tons, considerando a distância euclidiana e a intensidade de cor do pixel. O efeito gerado pela quantização pode ser visto no exemplo abaixo.

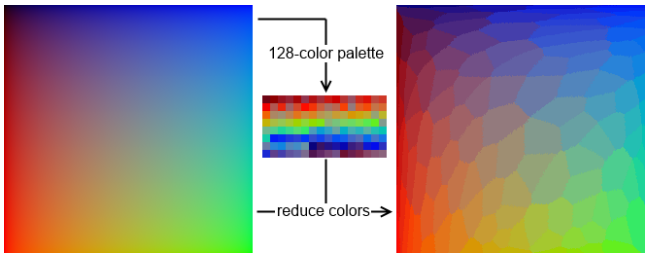

Figura 4. [a] Imagem original [b] Imagem após a Quantização de Cores

IV. Resultados

Após ter o algoritmo pronto, conseguindo realizar de forma satisfatória o efeito cartoon nas imagens, salientando as bordas, diminuindo os detalhes, deixando as cores mais sólidas, as regiões mais homogêneas e, por fim, a paleta de cores reduzida.

Para mensurar a aceitação das imagens geradas pelo o algoritmo pegamos uma amostra de 115 imagens e colocamos para avaliação pública. Alcançamos 25 respostas para cada uma das 115 imagens amostrais, totalizando 2875 avaliações.

Após analizar as respostas do público, a imagem abaixo foi a que alcançou a maior nota das 115, chegando à pontuação de média de 9.43 pontos. Segundo alguns *feedbacks* recebidos, houve uma detecção das bordas muito satisfatória, bem como as cores fiéis a realidade, sem a perda dos efeitos de profundidade, perda de sombras e de diferença de contraste nos balões.

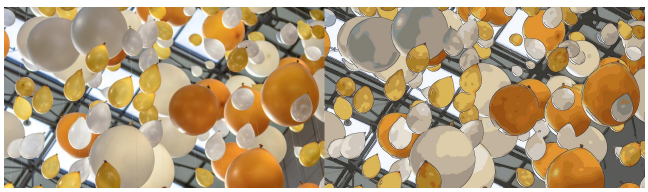

Figura 5. [a] Imagem original [b] Imagem Cartoonizada

Usando critérios mais técnicos, tivemos as duas imagens abaixos com os melhores resultados, sendo observado a conformidade com a ideia de cartoon e a fidelidade à imagem original, ou seja, sem perda informações e sem marcas que distoam do efeito cartoon.

Na imagem abaixo, é perceptível o sucesso ao detectar as bordas da imagem e a fidelidade nas cores ao aplicar o cartoon, o único ponto que foge em parte da cartoonização foram as sombras superior e inferior, que num cartoon seriam uniforme, sem variação de cores.

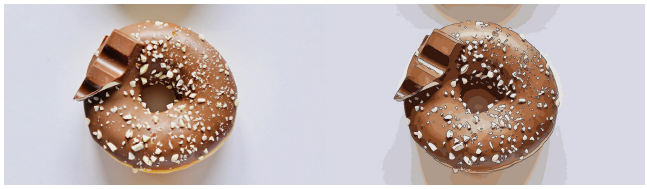

Figura 6. [a] Imagem original [b] Imagem Cartoonizada

Já na imagem abaixo, é notável o sucesso do cartoonização como um todo, deixando o céu todo com uma só cor e uniforme, as nuvens tendo uma pequena variação de cor, porém essa é notável na imagem original e, por fim, uma efetiva cartoonização da paisagem da imagem.

Figura 7. [a] Imagem original [b] Imagem Cartoonizada

Já como a imagem com pior avaliação do público, temos a imagem que se segue, acançando pouco mais de 4 pontos. Tal fato se deve da imagem ter um contraste muito grande entre as cores, causando um problema no hora da quantização das cores, e perdendo a fidelidade com a imagem original. Como foi percebido, a cor das peles mais claras foram mapeadas com a cor de concreto do fundo, que certamente acarretou uma avaliação mais baixa pelo público. Também vale ressaltar que a imagem represente uma pessoa, com isso a avaliação tende a ser mais rigorosa, em vista da sensibilidade ao analizar rostos, que no exemplo ficaram bem distorcidos e com bastante perda de detalhe.

Figura 8. [a] Imagem original [b] Imagem Cartoonizada

E outra imagem classificada como ruim, dessa vez usando critérios técnicos, é a imagem abaixo. Por sua vez, o motivo se deve a não fidelidade à cartoonização, já que a imagem apresenta vários intensidades de cor para representar o rosto. Outro ponto que se pode salientar é o fundo que ficou muito homogêneo, mas a sombra dela que não ficou homogêna, mas sim em dois tons, contrariando o que seria esperado de um efeito cartoon. E, também, a camisa dela que foi mapeada com duas cores diferentes.

Figura 9. [a] Imagem original [b] Imagem Cartoonizada

Além disso, vale ressaltar o problema que ocorreu quando a imagem tinha uma resolução muito alta. Por conta da resolução muito alta, imagens 4k sofrem uma o que chamamos de cartoonização interna, isto é, são cartoonizadas, mas não são visivéis se não ter um zoom local na imagem.

Para solucionar esse problema, usamos a função *resize* com interpolação cúbica, com isso reduzimos a resolução da imagem, em seguida aplica-se o algoritmo de cartoonização e, por fim, redefinimos a imagem para a resolução original.

A diferença entre as imagens pode ser vista nas figuras 10 e 11, sendo a 10 sem usar o algoritmo de redimensionamento e a 11 usando o esse algoritmo.

Figura 10. [a] Imagem original [b] Imagem cartoonizada sem redimensionamento

Figura 11. [a] Imagem original [b] Imagem cartoonizada com redimensionamento

V. Conclusão

Após a análise de todos as 2875 respostas e alcançando uma nota média de 7,19 pontos, ultrapassando a meta inicial.

Com uma análise mais minunciosa, foi possível notar a aceitação do público quando a imagem original representa uma passagem, animais ou quando o fundo tem um contraste forte com primeiro plano.

Já os piores resultados, em sumas, foram os que haviam pessoas, fundo de cor parecida com o primeiro plano, imagens com muitos detalhes.

Após ver as melhores e as piores imagens segundo as avaliações, é percepitivel que as melhores são mais uniformes (facilitando a detecção das bordas), cores mais homogêneas (deixando a quantização de cores mais simples). E as piores imagens,apresentam bastantes detalhes (causando excesso de bordas, o que distoa do cartoon), uma ampla paleta de cores (prejudicando o resultado da quantização das cores). Também vale ressaltar as imagens que representam pessoas, que em sua maioria não tiveram uma boa aceitação, principalmente por terem diferentes tonalidades de cores nos rostos.

AGRADECIMENTOS

À seleção brasileira de futebol, que irá trazer o Hexa para o Brasil da 2018 FIFA World Cup .

REFERÊNCIAS

[1] K. Dade *Toonify: Cartoon Photo Effect Application*, 3rd ed. Stanford, CA.