Trabalho final - Controle de qualidade em limões

Visão por Computador

Elaborado por:

Augusto Luvisa Dessanti - a61492

Gabriel Oliveira Pimentel - a61413

Bragança

09 de Junho, 2024

Conteúdo

Capítulo 1

Introdução

De maneira geral, este trabalho tem como objetivo realizar o controle de qualidade de limões utilizando dois conjuntos de dados distintos. Para isso, são identificadas e categorizadas em imagens de limões em boas e más condições, além de reconhecer imagens que não contêm limões. Neste contexto, o trabalho visa reconhecer a presença e condição dos limões utilizando técnicas de visão computacional, classificar limões através de redes neurais e, a partir desses resultados, avaliar a precisão e eficiência do sistema.

O trabalho está organizado da seguinte forma: o Capítulo 1 apresenta a introdução; o Capítulo 2 descreve a metodologia utilizada; o Capítulo 3 aborda a análise dos resultados e discussões; e, por fim, o Capítulo 4 trata das conclusões deste estudo.

Capítulo 2

Metodologia

Neste capítulo, é apresentada a metodologia aplicada neste trabalho. Para a realização deste trabalho, como previamente mencionado, foram empregados dois conjuntos de imagens de limões. Um desses conjuntos é composto por 2690 imagens de limões em boas condições e com mofo, todas com fundo preto, conforme exemplificado pela Figura 2.1.

Figura 2.1: Imagem de limão do primeiro conjunto de imagens.

O segundo conjunto de imagens, contendo 2528 imagens de limões, consistiu de imagens com fundo cinza e é dividido em três categorias distintas: 1125 imagens de limão de boa qualidade, 951 imagens de limão de péssima qualidade e 452 imagens sem limão. Um exemplo representativo deste conjunto é apresentado na Figura 2.2.

Figura 2.2: Imagem de limão do segundo conjunto de imagens.

A partir disso, realizou-se uma filtragem manual das imagens para reduzir o número de imagens a serem trabalhadas nesta aplicação e para remover imagens que apresentavam defeitos que poderiam dificultar as operações desenvolvidas. Dessa forma, decidiu-se utilizar 413 imagens do primeiro conjunto e 344 imagens do segundo conjunto.

A etapa subsequente deste trabalho envolveu a criação da interface gráfica utilizando a ferramenta do MATLAB. Essa interface foi projetada com uma lista das imagens disponíveis na pasta, dois gráficos para ilustrar a imagem original e o resultado da aplicação, e uma caixa de texto para exibir o resultado obtido. Além disso, tem-se quatro botões para executar as operações de verificação da qualidade do limão.

Dos quatro botões disponíveis, dois são destinados à realização de operações manuais: um para executar a verificação apenas uma imagem e exibir o resultado visualmente na interface, e o outro para repetir a verificação em todas as imagens e gerar o resultado em um arquivo `.xlsx`. De maneira semelhante, há dois botões para realização da verificação por rede neural. A interface gráfica desenvolvida pode ser observada na Figura 2.3.

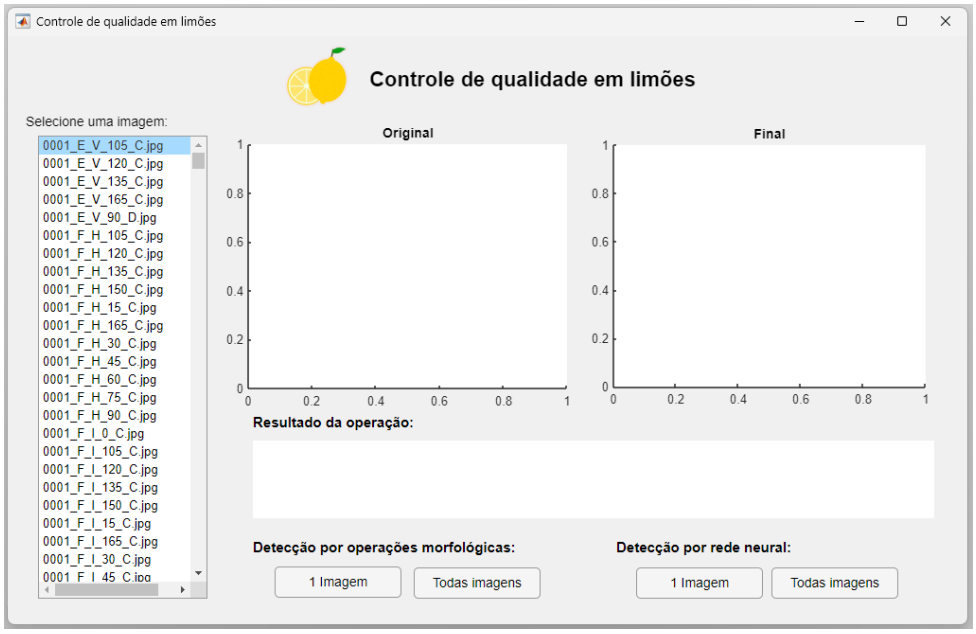

Figura 2.3: Interface gráfica da aplicação.

A construção dessa interface é realizada através da manipulação de elementos gráficos disponíveis no MATLAB, possibilitando a elaboração de funções a serem executadas através de *callbacks* dos elementos gráficos.

Com isso, o trabalho foi elaborado para identificar a condição do limão através de duas abordagens: por meio de operações morfológicas, utilizando operações morfológicas, técnicas de visão computacional, análise de histograma e segmentações; e por meio de rede neural, para classificar a imagem com base em uma rede treinada.

2.1 Detecção por operações morfológicas

Como essa aplicação consistiu em utilizar dois datasets, a primeira funcionalidade da aplicação foi padronizar todas as imagens, redimensionando-as para um tamanho de 300x300 pixels. Isso permitiu que todas as imagens fossem tratadas de maneira uniforme nas operações subsequentes.

Para detectar e classificar as imagens, a aplicação começa com a verificação do fundo da imagem. Ao selecionar uma imagem da lista de imagens na interface e executar a

funcionalidade, a primeira etapa é contar o número de pixels pretos presentes na imagem. Se o número de pixels pretos ultrapassasse 10.000, a imagem era identificada como pertencente ao primeiro conjunto de dados, que contém imagens com fundo preto.

2.1.1 Imagens de fundo preto

Para as imagens com fundo preto, o tratamento envolveu a conversão da imagem para escala de cinza. Em seguida, foi calculado um valor de limiar (*threshold*) utilizando o método de Otsu, que minimiza a variância dentro das classes. A imagem é binarizada usando a função `imbinarize`. Com a imagem binarizada, as bordas dos contornos do limão são obtidas através da função `bwboundaries`, permitindo o desenho do contorno do limão. Nesta etapa, a imagem do limão com seu respectivo contorno já estava disponível, pronta para a verificação da condição do limão, conforme exemplificado pela Figura 2.4.

Figura 2.4: Imagem de limão de fundo preto com o contorno.

2.1.2 Imagens de fundo cinza

Se a imagem tivesse um número de pixels pretos abaixo do limite de 10.000, ela era identificada como pertencente ao segundo conjunto de dados, que contém imagens com fundo cinza.

Para estas imagens, uma binarização foi realizada e uma função foi desenvolvida para verificar a presença de um limão. Essa função contava o número de pixels brancos na imagem; se o número de pixels brancos fosse superior a 150.000, a imagem era considerada como não contendo um limão, e a interface exibiria essa informação.

Para imagens com menos de 150.000 pixels brancos, uma função foi elaborada para remover o fundo e segmentar apenas o limão. Primeiramente, a imagem é convertida para o espaço de cores HSV usando a função `rgb2hsv`, como ilustrado pela Figura 2.5, o que facilitou a segmentação baseada na cor.

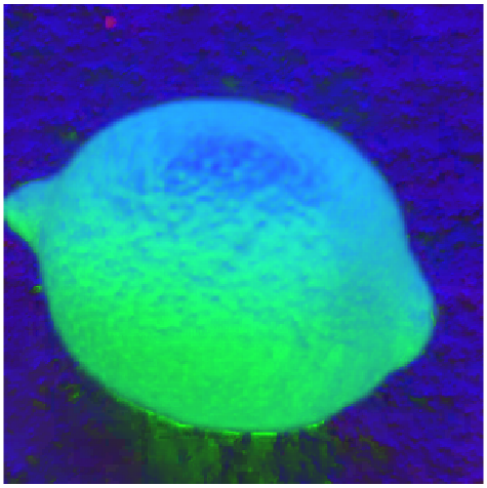

Figura 2.5: Imagem no espaço de cores HSV.

A segmentação é realizada com base na cor amarela característica do limão, definindo limiares para os componentes de Matiz (*Hue*), Saturação (*Saturation*) e Brilho (*Value*). A partir disso, uma máscara foi criada combinando esses limiares, selecionando pixels com matiz entre 0.0 e 0.15, saturação entre 0.3 e 1, e brilho entre 0.2 e 1.

Após a criação da máscara, a imagem passa por um processamento adicional por meio de operações como `imfill` e a função `bwareaopen` para remover pequenas áreas ruidosas. A partir disso, a máscara é refinada com operações morfológicas, como `imclose` e `imopen`, para remover imperfeições, resultado em uma máscara conforme mostrado pela Figura 2.6.

Figura 2.6: Máscara do limão criada para segmentar o limão.

Com a máscara refinada, ela é aplicada à imagem original, definindo os pixels fora da máscara como preto (0), resultando na imagem segmentada. A função `bwboundaries` identifica os contornos das regiões conectadas e a função `regionprops` extrai propriedades das regiões, como a área.

Finalmente, a imagem segmentada é exibida com contornos das regiões maiores que o limiar definido, destacando o limão, conforme mostrado pela Figura 2.7.

Figura 2.7: Imagem de limão de fundo cinza com o contorno.

2.1.3 Verificação da condição do limão

Para a etapa da verificação da condição do limão, uma função foi desenvolvida para determinar se o limão estava em boa qualidade ou com mofo. Nesta função, a imagem tratada contém apenas o limão segmentado com fundo preto para ser analisada da maneira igual para ambos conjuntos de imagens.

Com isso, os canais de cor vermelho, verde e azul são extraídos da imagem. Uma máscara é aplicada para isolar os pixels do limão, ignorando os pixels pretos. Em seguida, as médias dos valores dos pixels para cada canal de cor são calculadas. A partir disso, a condição do limão é determinada comparando as médias dos canais de cor com limites estabelecidos: 128 para o canal vermelho, 105 para o canal verde e 60 para o canal azul. Se a média dos canais satisfizesse essas condições, o limão é considerado de boa qualidade; caso contrário, é considerado com mofo.

2.2 Detecção por rede neural

Para realizar a detecção utilizando a rede neural, o primeiro passo foi organizar as imagens selecionadas em três categorias: limões de boa qualidade, limões com mofo e imagens sem limões, cada uma armazenada em uma pasta específica.

Em seguida, as imagens de ambos os conjuntos de dados foram carregadas usando a função `imageDatastore`, com os rótulos (*labels*) baseados nos nomes das pastas. Essa etapa foi essencial para preparar as imagens para o treinamento da rede neural.

O conjunto de dados foi então dividido aleatoriamente em três subconjuntos: 70% para treinamento, 15% para validação e 15% para teste. Então, as imagens foram redimensionadas para um tamanho padrão de 300x300 pixels, garantindo uniformidade.

Com as imagens separadas, a rede neural foi estruturada com várias camadas: uma camada de entrada para as imagens; camadas de convolução para extrair características; camada de normalização para estabilizar o treinamento; função de ativação $ReLU$ para introduzir não-linearidade; camada conectada para classificação; e finalmente camadas *softmax* e de classificação para gerar probabilidades de classificação e calcular a perda.

Após definir a arquitetura da rede e configuração do treinamento, como o algoritmo de otimização. O treinamento foi realizado, de modo que é possível acompanhar o progresso do treinamento em tela, conforme mostrado pela Figura 2.8.

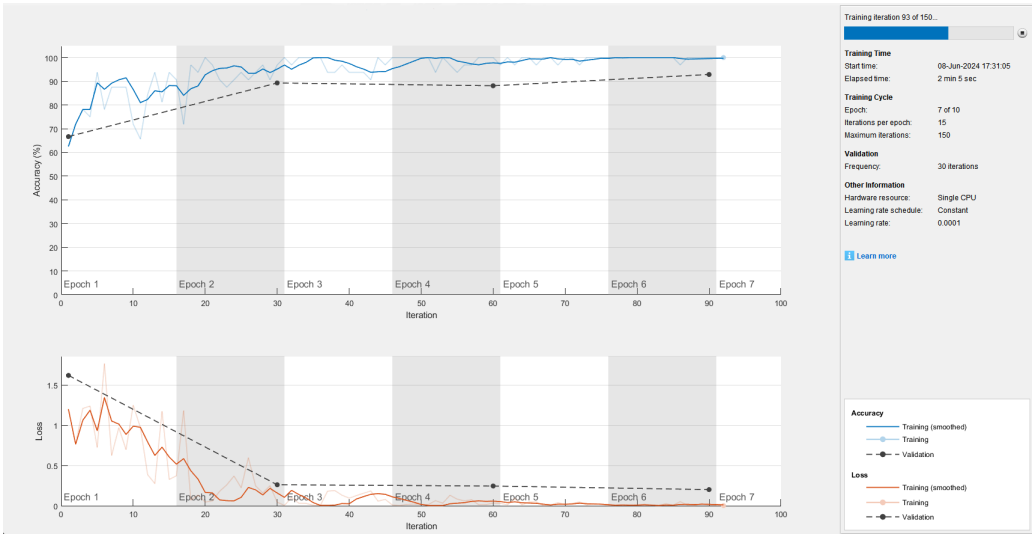

Figura 2.8: Toolbox de treinamento da rede neural.

Uma vez treinada, a rede neural é avaliada com o conjunto de validação para calcular a acurácia, comparando as previsões da rede com os rótulos reais das imagens de validação.

Com o modelo treinado, a rede neural foi salva em um arquivo `.mat` para ser utilizada na interface do trabalho. Na interface, ao selecionar uma ou várias imagens para detecção por rede neural, a função `classify` é utilizada para classificar a(s) imagem(ns) com a rede treinada. O resultado é interpretado e exibido na tela, indicando se a imagem continha um limão de boa qualidade, um limão com mofo, ou se não continha limão.

Capítulo 3

Resultados

Neste capítulo, são apresentados os resultados obtidos durante a realização deste trabalho, os quais são fundamentais para avaliar a eficácia da metodologia proposta. A partir das funcionalidades implementadas, a Figura 3.1 mostra o resultado da aplicação para uma detecção de limão em boa qualidade com a operação morfológica.

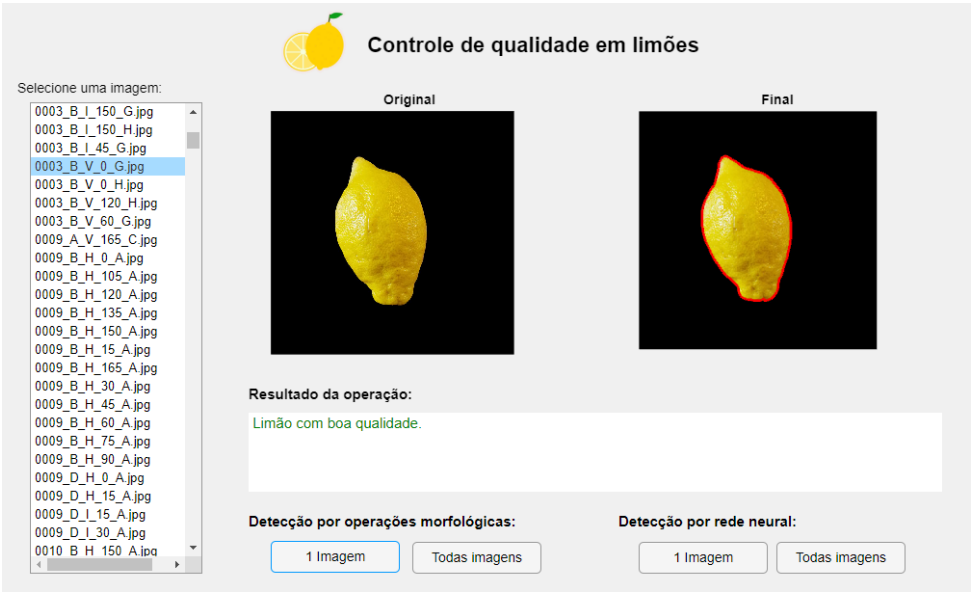

Figura 3.1: Resultado de imagem de limão com boa qualidade.

De maneira semelhante, a Figura 3.2 mostra o resultado da aplicação para uma imagem selecionada de limão em péssima qualidade com a operação morfológica.

Figura 3.2: Resultado de imagem de limão com mofo.

Além disso, a Figura 3.3 mostra o resultado da aplicação para uma detecção de imagem sem limão com a operação morfológica.

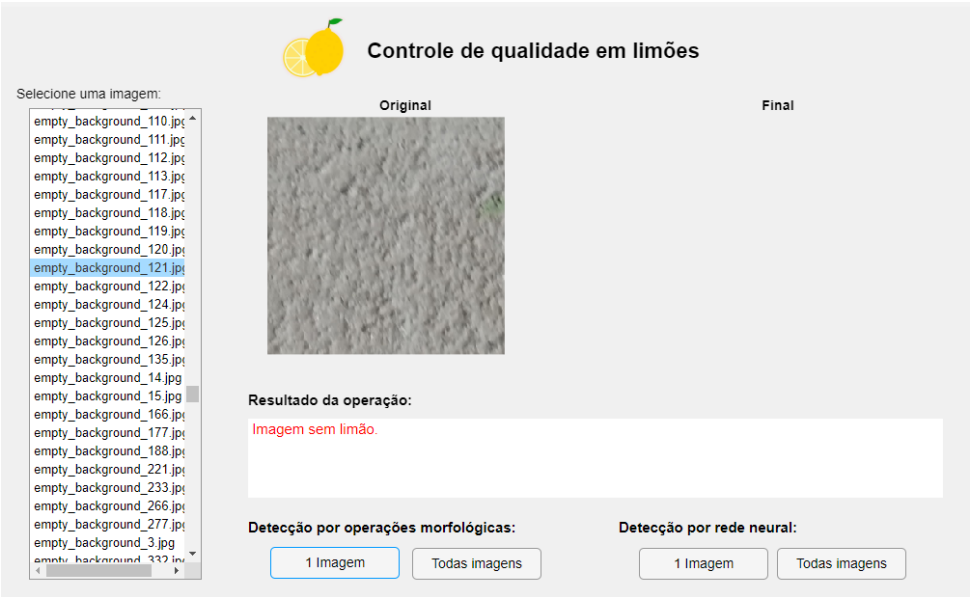

Figura 3.3: Resultado de imagem sem limão.

Por fim, a Figura 3.4 demonstra o resultado da aplicação para uma detecção de limão em boa qualidade utilizando a operação de rede neural.

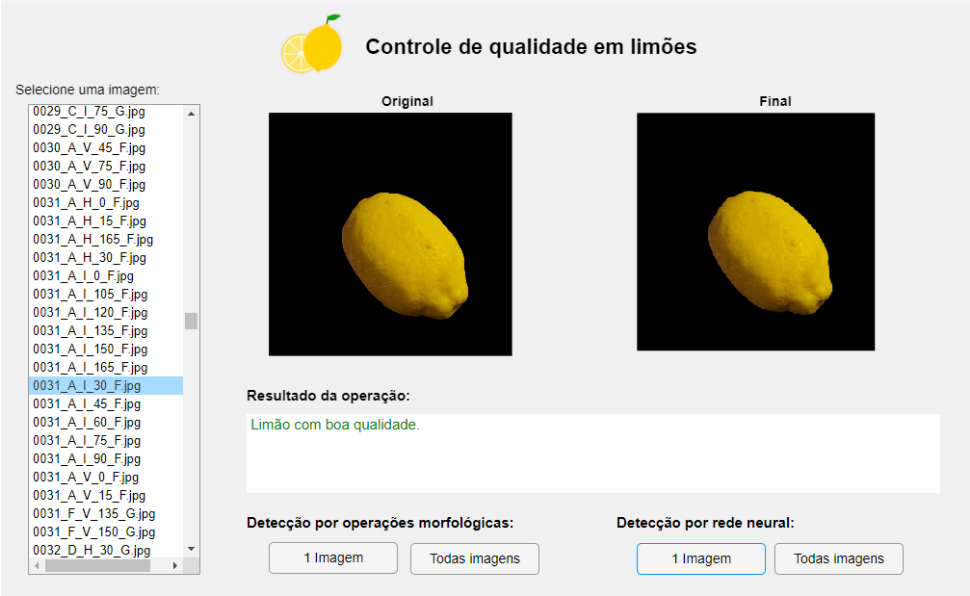

Figura 3.4: Imagem de limão com boa qualidade.

Adicionalmente, para a execução da verificação em "Todas imagens", a Figura 3.5 ilustra os resultados apresentados na interface após a execução da funcionalidade.

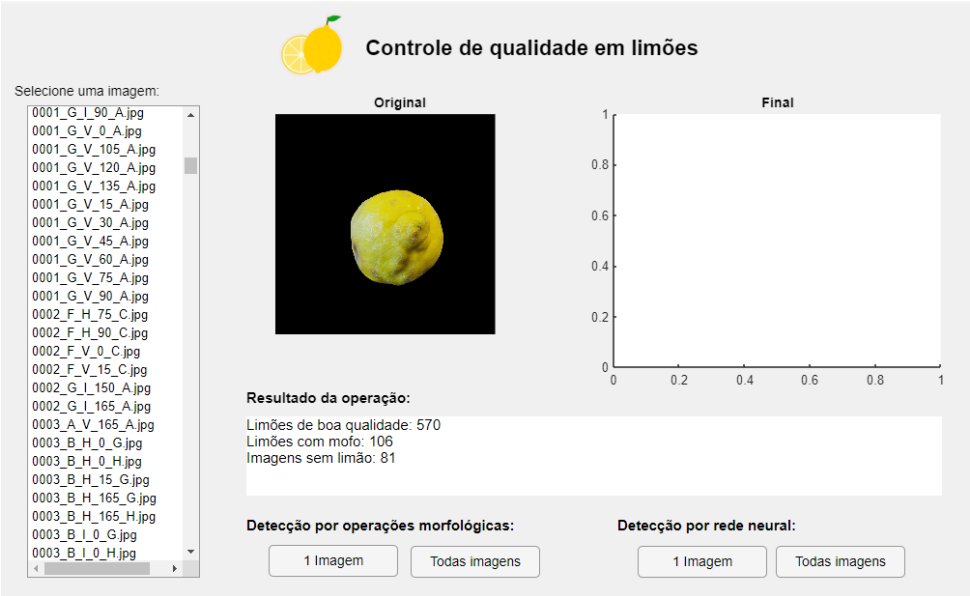

Figura 3.5: Resultados para todas imagens com operações morfológicas.

E a partir dessa execução para todas as imagens, foram obtidos resultados como os exemplificados nas Tabelas 3.1 e 3.2, que indicam os resultados gerados a partir das operações morfológicas e da rede neural, respectivamente.

Tabela 3.1: Resultados gerados com operações morfológicas.

Imagem	Resultado
0001_F_V_0_C.jpg	Limão com boa qualidade
0001_F_V_15_C.jpg	Limão com boa qualidade
0001_F_V_30_C.jpg	Limão com boa qualidade
0001_F_V_75_C.jpg	Limão com boa qualidade
0001_F_V_90_C.jpg	Limão com boa qualidade
0001_G_I_150_A.jpg	Limão com mofo
0001_G_I_165_A.jpg	Limão com mofo
0002_F_H_75_C.jpg	Limão com boa qualidade

Tabela 3.2: Resultados gerados com rede neural.

Imagem	Resultado
0001_F_V_0_C.jpg	good_quality
0001_F_V_15_C.jpg	good_quality
0001_F_V_30_C.jpg	good_quality
0001_F_V_75_C.jpg	good_quality
0001_F_V_90_C.jpg	good_quality
0001_G_I_150_A.jpg	bad_quality
0001_G_I_165_A.jpg	bad_quality
0002_F_H_75_C.jpg	good_quality

Como observado, ambas as ferramentas geram resultados semelhantes. No entanto, é importante ressaltar que os resultados obtidos através da rede neural são baseados nas pastas em que as imagens foram separadas, o que explica os resultados em inglês.

A Tabela 3.3 apresenta os resultados obtidos pela detecção através de operações morfológicas em todas as imagens da pasta.

Tabela 3.3: Resultados das operações morfológicas para todas imagens.

Categoria	Total	Acertos	Taxa de acerto (%)
Limão de boa qualidade	570	520	91,22%
Limão com mofo	106	96	90,57%
Imagem sem limão	81	81	100%

Como observado, para este tipo de detecção, a aplicação mostrou uma taxa de acerto de 100% para imagens sem limão. Para imagens de limões, os resultados também foram satisfatórios, apresentando uma taxa de acerto acima de 90%.

A Tabela 3.4 apresenta os resultados obtidos pela detecção através da rede neural previamente treinada em todas as imagens da pasta.

Tabela 3.4: Resultados da rede neural para todas imagens.

Categoria	Total	Acertos	Taxa de acerto (%)
Limão de boa qualidade	570	530	92,98%
Limão com mofo	106	104	98,11%
Imagem sem limão	81	81	100%

Os resultados da rede neural também foram satisfatórios, com uma taxa de acerto de 100% para imagens sem limão. Para imagens de limões, a rede neural apresentou uma taxa de acerto acima dos 90%.

3.1 Análise de resultados

Dessa forma, observa-se que a interface desenvolvida apresentou bons desempenhos para as imagens previamente selecionadas. Contudo, é importante destacar que a interface não possui uma precisão de 100% devido às condições variáveis das imagens. Defeitos como iluminação inadequada, borramento, e até mesmo a condição do limão podem não satisfazerem os limites aplicados para detectar corretamente a condição do limão.

Além disso, outra limitação da aplicação que cabe ressaltar é que, em diversos casos a aplicação detectou limão com mofo sendo limão em boa qualidade, devido a pequenas imperfeições do limão que não foram detectadas. E mais uma limitação da aplicação é a segmentação dos limões em imagens de fundo cinza, que devido a defeitos de iluminação na imagem, interferiram na segmentação.

Todavia, observa-se que, ambas formas apresentara alta precisão, com destaque para rede neural, que demonstra uma vantagem em termos de precisão na detecção de limões em boa qualidade e com mofo. Isso pode ser atribuído à capacidade da rede neural de aprender e generalizar padrões mais complexos nas imagens, tornando-a mais robusta em cenários variados.

Capítulo 4

Conclusão

Os resultados obtidos neste trabalho demonstram que ambas as metodologias são capazes de detectar limões com uma alta taxa de acerto. Foram identificadas algumas limitações, como a dificuldade da aplicação em detectar pequenas imperfeições nos limões e problemas de iluminação nas imagens, que afetaram a segmentação dos limões, especialmente em fundos cinza.

Apesar dessas limitações, que eram esperadas devido às diferenças nos conjuntos de imagens, ambas as metodologias apresentaram um desempenho satisfatório na detecção de limões. A rede neural mostrou uma vantagem em termos de precisão, especialmente na identificação de limões com mofo, devido a sua capacidade de aprender e generalizar padrões complexos.

Em resumo, os métodos implementados, os resultados obtidos e as análises realizadas foram eficazes em cumprir os objetivos propostos, fornecendo uma base sólida para futuras aplicações no campo de visão computacional e contribuindo para o avanço na automação da classificação de frutas. Para trabalhos futuros, aprimoramentos podem ser realizados para aumentar a robustez das detecções, como o tratamento prévio de imagens para corrigir falhas, a melhoria na qualidade das imagens e o treinamento de redes neurais mais robustas.