



Ministério da Educação
UNIVERSIDADE TECNOLÓGICA FEDERAL DO PARANÁ
Campus Toledo



CURSO: Engenharia de Computação (EC)

Projeto	05/07/2025
Disciplina: OPCPF-Processamento Digital de Imagens	
Professor: Fabio Alexandre SPANHOL, faspanhol@gmail.com	
Turma: CP94 Período: 9 Turno: Integral	
GitHub: https://github.com/Duarte-G/Choco-Classifier	

Alunos

Gustavo d'Avila	2370506
Gabriel Duarte	2382954
Jhony Willian Reina	1825755
Edson Pirassol Junior	2097176
Samuel Damiani Frigotto	2266113

Sumário

1. Criação da base de dados.....	3
2. Anotação das imagens.....	3
3. Aumento de dados.....	4
4. Normalização de dados.....	5
5. Segmentação dos objetos.....	6
6. Classificador.....	7
6.1. Datasets utilizados.....	7
6.2. Extração de features.....	7
6.3. Treinamento.....	8
6.4. Resultados do treinamento.....	8
6.5. Predições.....	9
7. CNN.....	11
7.1. Datasets utilizados.....	11
7.2. Arquitetura da CNN.....	11
7.3. Treinamento.....	12
7.4. Resultados do Treinamento.....	12
7.5. Relatório de Classificação.....	14
7.7. Comparações de Curvas ROC.....	15
8. Conclusão.....	16
9. Referências.....	17

1. Criação da base de dados

A base de dados foi construída a partir da captura manual de imagens de 10 tipos diferentes de chocolates: *Alpino*, *Amor Carioca*, *Charge*, *Chokito*, *Galak*, *Lollo*, *Negresco*, *Prestígio*, *Sensação* e *Smash*.

As fotos foram tiradas com a câmera de um celular Samsung Galaxy A22, sob iluminação artificial de lâmpadas, garantindo um ambiente controlado e com boa visibilidade dos objetos. Para representar variações realistas, cada chocolate foi fotografado em 3 ângulos diferentes, além de imagens com o chocolate em diferentes rotações, simulando mudanças de perspectiva.

Foram feitas 8 imagens com fundo preto e 8 com fundo branco para cada tipo de chocolate, totalizando 16 imagens por classe e 160 imagens ao todo.

Para manter a organização e facilitar o acesso durante o processo de anotação e treinamento, as imagens foram salvas em pastas separadas por tipo de chocolate, mantendo uma estrutura clara e intuitiva do conjunto de dados.

2. Anotação das imagens

No início do projeto, realizamos a anotação das imagens manualmente, desenhando bounding boxes ao redor dos objetos de interesse. Esse processo exigia bastante tempo e atenção, já que cada imagem precisava ser inspecionada individualmente.

Com o avanço do trabalho, identificamos que seria mais eficiente aplicar técnicas de segmentação de objetos para automatizar parte dessa etapa. A partir das máscaras geradas pela segmentação, conseguimos extrair automaticamente as bounding boxes, o que facilitou significativamente o processo de anotação. Essa abordagem nos permitiu manter a qualidade e a consistência dos dados, reduzindo o esforço manual e acelerando a criação do conjunto de treinamento.

Exemplo: prestigio\7-03-V1-B.png



(Exemplo da bounding box)

3. Aumento de dados

Com o objetivo de gerar variações artificiais das imagens originais para melhorar o treinamento de modelos, foi aplicado data augmentation (aumento de dados) em nosso dataset de imagens, aplicando as transformações.

Transformações principais:

- Transformação logarítmica (para realçar regiões escuras)
- Transformação exponencial (para realçar regiões claras)
- Filtro de média 3×3 (suavização)

Versões agressivas dessas transformações:

- Transformação logarítmica e exponencial com intensidade suavizada.
- Filtro de média 5x5 aplicado uma ou duas vezes.

Filtros extras:

- Gaussian Blur
- Motion Blur
- Additive Gaussian Noise
- Ajuste de contraste linear
- Deslocamento de brilho

Durante a fase de prototipação e testes, também foram testados **Flip horizontal/vertical** mas estes se tornaram problemas nesta fase, sendo resolvido posteriormente, facilitando a segmentação. A variação de **rotação aleatória** também foi testado mas gera artefatos, acarretando em problemas na segmentação.

Foi treinado e testado modelos utilizando entre 3 à 6 imagens aumentadas por imagem original, pelo pequeno tamanho de dados optamos por utilizar 6 imagens na versão final.



(Exemplo do data augmentation utilizado)

4. Normalização de dados

A normalização de um dataset de imagens aumentado (augmentado) foi feito através de técnicas de equalização de histograma. O objetivo é melhorar o contraste das imagens para posterior processamento ou treinamento de modelos de aprendizado de máquina.

Para cada imagem em cada classe:

- Verifica se é um arquivo de imagem válido (.png, .jpg, .jpeg)
- Carrega a imagem em escala de cinza
- Aplica equalização de histograma global usando **cv2.equalizeHist()**
- Salva a imagem processada no diretório de saída correspondente

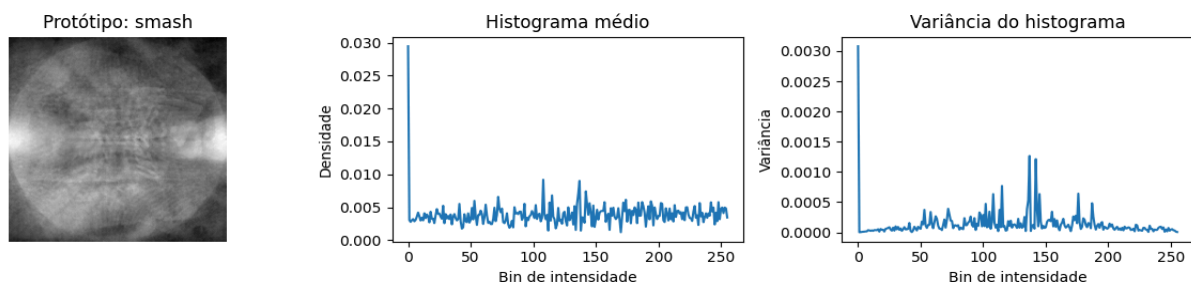
A equalização de histograma global pode não ser ideal para todas as imagens, especialmente aquelas com variações locais de contraste. O CLAHE (que está opcionalmente presente) geralmente produz melhores resultados para muitos tipos de imagens, mas para nosso caso não se tornou necessário.

Seguidamente, é calculado protótipos (imagens médias) e estatísticas de histograma (média e variância) para cada classe de um dataset normalizado de imagens em escala de cinza.

Para cada classe no dataset normalizado é realizado:

- Carregamento das imagens:
 - Converte imagens para float32 e normaliza valores para [0,1]
 - Calcula histogramas normalizados de 256 bins
- Cálculo de estatísticas:
 - Protótipo: média pixel a pixel de todas as imagens da classe
 - Média e variância dos histogramas da classe
- Visualização:
 - Gera uma figura com 3 subplots:
 1. Imagem do protótipo
 2. Gráfico do histograma médio
 3. Gráfico da variância do histograma

Exemplo de visualização:



(Exemplo de saída para o protótipo bombom smash)

5. Segmentação dos objetos

O Ground Truth foi proposto de ser gerado/composto por pelo menos 25% das amostras de cada classe, como foi aplicado métodos automáticos de segmentação, decidimos por termos metade das segmentações feitas manualmente e outra metade foi automatizada. Ground Truth (ou "verdade fundamental" em português) é um termo utilizado em ciência de dados, aprendizado de máquina e visão computacional para se referir aos dados de referência confiáveis e precisos que servem como base para treinar, validar e testar modelos. Ele representa a informação correta e incontestável contra a qual os resultados de um algoritmo são comparados.



(Exemplo de ground truth aplicado à uma imagem original)

No método manual, realizamos inicialmente a segmentação completa dos objetos pós-processados, gerando assim imagens que preservam as características da imagem original. As demais imagens correspondem às versões aumentadas (data augmentation), que em determinados cenários podem sofrer inversão horizontal (flipH) ou vertical (flipV), conforme indicado pela nomenclatura no final de cada arquivo.

A segmentação dessas imagens aumentadas é gerada automaticamente através da aplicação das mesmas transformações geométricas realizadas nas imagens originais. Esse processo é executado sobre a ground truth previamente estabelecida, garantindo perfeita correspondência espacial entre as imagens e suas máscaras de segmentação. A seguir, um exemplo de ground truth aplicado à uma imagem aumentada.



(Exemplo de ground truth aplicado à uma imagem aumentada)

6. Classificador

Para a classificação dos bombons, adotamos uma abordagem de Machine Learning clássico. Preparamos e aumentamos nosso dataset de imagens e, a partir delas, extraímos um conjunto de características visuais de cor, textura e forma. Essas features foram então utilizadas para treinar um modelo Random Forest, capacitando-o a realizar as predições.

6.1. Datasets utilizados

Todas variações de datasets foram criadas, porém o uso do dataset normalizado apresentou diversos problemas no classificador, optamos por utilizar apenas o dataset augmented.

O uso de técnicas para Augmentation um pouco agressiva ajudou o modelo a sair da “zona de conforto”, no qual ele apresentava uma precisão e acurácia bastante alto no teste/validação, mas uma certa dificuldade para identificar imagens retiradas da internet.

6.2. Extração de features

A classe disponibiliza três métodos principais de extração de características que transformam cada imagem em vetores descritivos de cor, textura e forma:

- **extract_color_features (27 features):** converte a imagem para os espaços de cor BGR, HSV e LAB e, em cada canal desses três espaços, calcula estatísticas básicas (média, desvio-padrão, mediana, mínimo e máximo) para descrever de forma completa a distribuição e vivacidade das cores.
- **extract_texture_features (12 features):** transforma a imagem em tons de cinza e aplica filtros de Gabor em quatro direções para capturar padrões de riscas, além de calcular gradientes horizontais e verticais (filtros Sobel) e extrair suas estatísticas (média e

desvio-padrão), fornecendo um resumo da textura e da presença de bordas.

- **extract_shape_features (5 features):** também na versão em tons de cinza, identifica o maior contorno (presumivelmente o bombom), e a partir dele mede área, perímetro, circularidade (similaridade a um círculo), razão de aspecto do retângulo envolvente e solidez (relação entre área do objeto e área do casco convexo) para caracterizar o formato do objeto.

6.3. Treinamento

O modelo Random Forest foi treinado utilizando um conjunto de hiperparâmetros otimizados para a classificação de imagens. Foram configuradas 100 árvores de decisão (`n_estimators`) com profundidade máxima de 10 níveis (`max_depth`) para evitar o overfitting.

Um controle adicional de complexidade foi implementado pelos parâmetros `min_samples_split = 5` e `min_samples_leaf = 2`, que garantem que cada divisão de nó precise pelo menos 5 amostras e cada folha contenha no mínimo 2 amostras.

O dataset foi dividido em 80% para treinamento, 10% para validação e 10% para teste, como requisitado pelo trabalho.

6.4. Resultados do treinamento

O modelo apresentou uma performance e acurácia satisfatória para o conjunto de treinamento, validação e teste, representados pela sua pontuação de acurácia e f1-score abaixo.

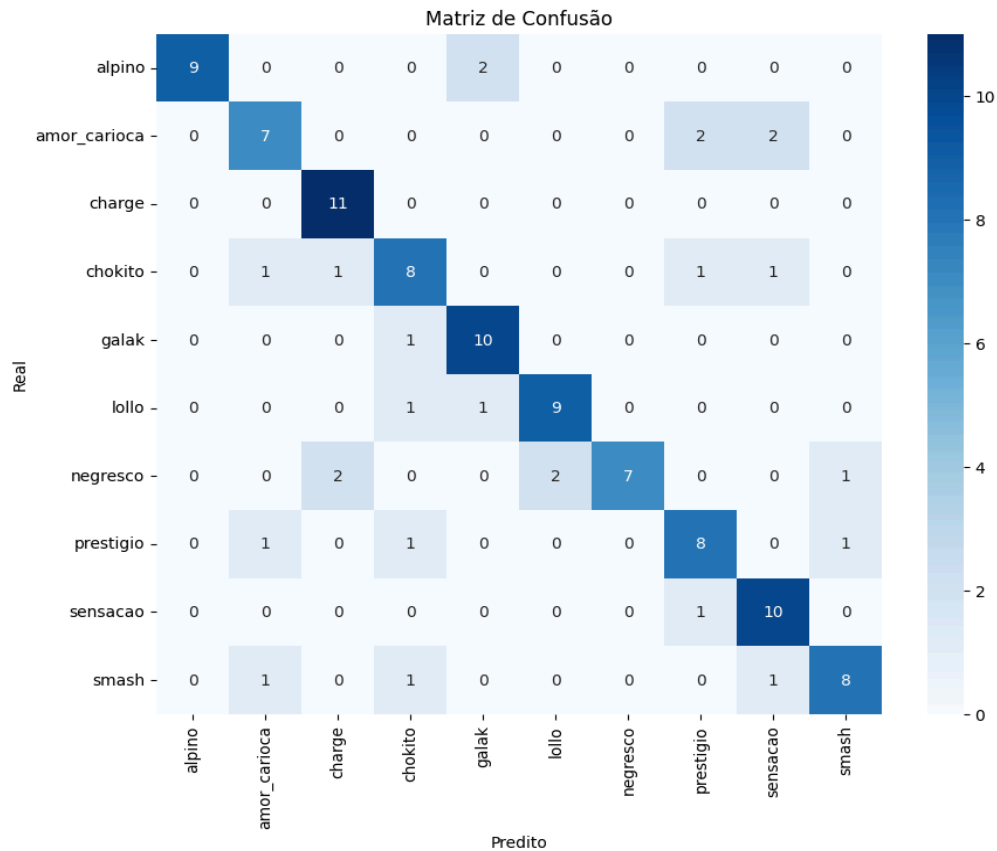
```
Acurácia de treino: 0.992
Acurácia de validação: 0.866
Acurácia de teste: 0.777

Relatório de classificação (conjunto de teste):
```

	precision	recall	f1-score	support
alpino	1.00	0.82	0.90	11
amor_carioca	0.70	0.64	0.67	11
charge	0.79	1.00	0.88	11
chokito	0.67	0.67	0.67	12
galak	0.77	0.91	0.83	11
lollo	0.82	0.82	0.82	11
negresco	1.00	0.58	0.74	12
prestigio	0.67	0.73	0.70	11
sensacao	0.71	0.91	0.80	11
smash	0.80	0.73	0.76	11
accuracy			0.78	112
macro avg	0.79	0.78	0.78	112
weighted avg	0.79	0.78	0.77	112

(Tabela de informações pós treinamento)

A matriz de confusão mostra o desempenho detalhado para cada classe de bombom, permitindo a visualização dos lugares na qual o modelo teve facilidade e dificuldade para a identificação.



(Matriz de confusão gerada após o treinamento)

Mesmo que os resultados apontem uma precisão abaixo de 80%, pela grande variedade de classes e a pequena diferença presente entre elas, o resultado foi bastante agradável e apresenta um bom modelo de classificador.

6.5. Predições

O modelo está sendo utilizado para detectar 8 imagens de diferentes bombons encontrados pela internet, todos mantendo um padrão de fundo branco para facilitar a detecção.

A predição devolve as três maiores confianças detectadas durante o processo, para facilitar e ajudar a detecção de problemas e/ou objetos que possuem bastante similaridade, abaixo estão os resultados encontrados:

- Resultados certos

```
=====
Imagem: alpinoBranco.png
Top 3 predições:
  1º: alpino (Confiança: 0.502)
  2º: chokito (Confiança: 0.151)
  3º: smash (Confiança: 0.096)

=====
Imagem: amorcariocaBranco.png
Top 3 predições:
  1º: amor_carioca (Confiança: 0.620)
  2º: smash (Confiança: 0.129)
  3º: chokito (Confiança: 0.085)

=====
Imagem: chargeBranco.png
Top 3 predições:
  1º: charge (Confiança: 0.287)
  2º: chokito (Confiança: 0.168)
  3º: amor_carioca (Confiança: 0.133)

=====
Imagem: chokitoBranco.png
Top 3 predições:
  1º: chokito (Confiança: 0.341)
  2º: amor_carioca (Confiança: 0.185)
  3º: smash (Confiança: 0.169)
```

- Resultados próximo ao correto

```
=====
Imagem: galakBranco.png
Top 3 predições:
  1º: alpino (Confiança: 0.286)
  2º: chokito (Confiança: 0.210)
  3º: charge (Confiança: 0.153)

=====
Imagem: lolloBranco.png
Top 3 predições:
  1º: negresco (Confiança: 0.255)
  2º: charge (Confiança: 0.158)
  3º: lollo (Confiança: 0.145)

=====
Imagem: negrescoBranco.png
Top 3 predições:
  1º: charge (Confiança: 0.275)
  2º: negresco (Confiança: 0.151)
  3º: chokito (Confiança: 0.134)
```

- Resultados completamente errados

```
=====
Imagem: smashBranco.png
Top 3 predições:
  1º: amor_carioca (Confiança: 0.420)
  2º: chokito (Confiança: 0.216)
  3º: smash (Confiança: 0.184)
```

7. CNN

7.1. Datasets utilizados

Reaproveitamos exatamente as mesmas partições de imagens que definimos no capítulo anterior — 80 % para treino, 10 % para validação e 10 % para teste — garantindo uma comparação justa entre os dois classificadores. Ao contrário do classificador baseado em features, aqui utilizamos as imagens originais em sua resolução nativa ($H \times W \times 3$), sem recortes nem vetores descritivos pré-calculados.

Antes do treinamento, todas as imagens foram normalizadas para o intervalo $[0, 1]$ e redimensionadas para 224×224 pixels, mantendo a proporção de aspecto. Para aumentar a robustez do modelo, aplicamos augmentations durante o treino: flips horizontais e verticais, borrões, mudança de contraste etc.

7.2. Arquitetura da CNN

A rede é estruturada em três blocos convolucionais empilhados, seguidos por camadas densas:

1. Bloco Conv1

- a. Conv2D com 16 filtros, kernel 3×3 , padding “same”
- b. ReLU
- c. MaxPooling2D (pool size 2×2)
- d. Dropout(0.25)

2. Bloco Conv2

- a. Conv2D com 32 filtros, kernel 3×3 , padding “same”
- b. ReLU
- c. MaxPooling2D (pool size 2×2)
- d. Dropout(0.25)

3. Bloco Conv3

- a. Conv2D com 64 filtros, kernel 3×3 , padding “same”
- b. ReLU
- c. MaxPooling2D (pool size 2×2)
- d. Dropout(0.25)

4. Após a extração de características:

- a. Flatten
- b. Dropout(0.5)
- c. Dense(128) + ReLU
- d. Dropout(0.5)
- e. Dense(NUM_CLASSES) + Softmax

Dessa forma garantimos que cada conjunto de filtros preserve as dimensões espaciais iniciais (graças ao padding “same”) e adicionamos

camadas de Dropout tanto entre convoluções quanto antes das densas, reduzindo overfitting e melhorando a robustez do modelo.

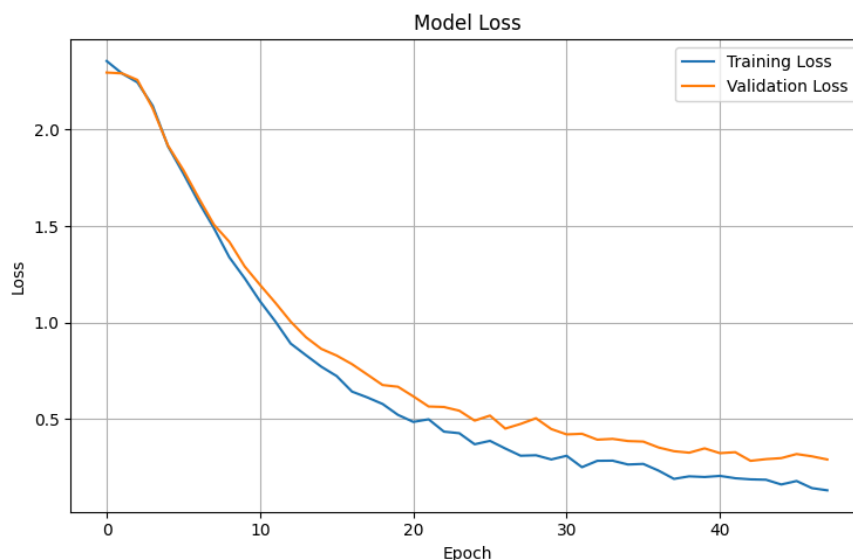
7.3. Treinamento

O treinamento da CNN foi configurado exatamente conforme o código enviado, destacando:

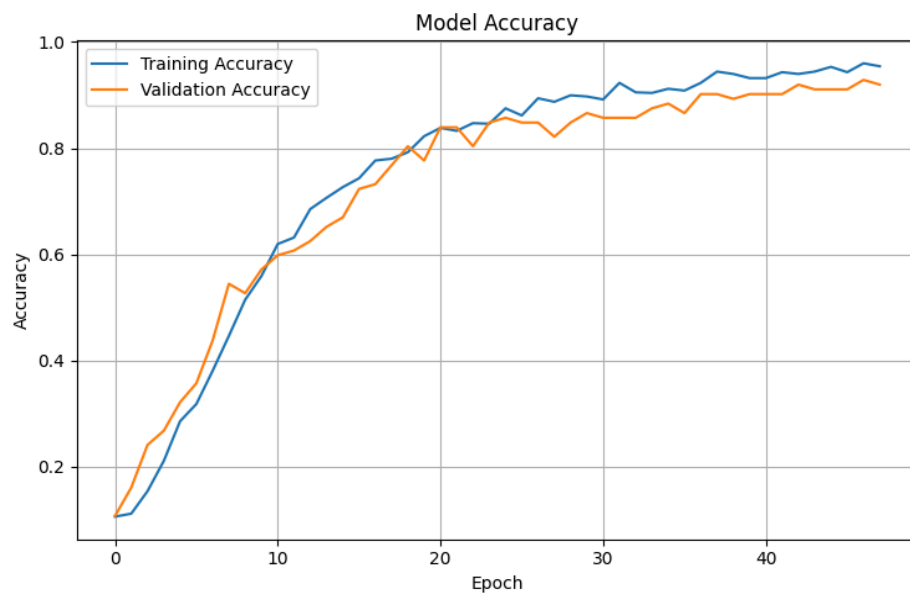
- Compilação do modelo
 - Otimizador: Adam com learning rate definido em `LEARNING_RATE` (por exemplo, 1×10^{-4}).
 - Função de perda: `sparse_categorical_crossentropy`, compatível com labels inteiras.
 - Métrica: acurácia (accuracy).
- Callbacks utilizados
 - EarlyStopping: `patience=5` (para interromper após 5 épocas sem melhora)
 - ReduceLROnPlateau: `patience=3` (aplica redução após 3 épocas sem queda na loss)
 - Parâmetros de execução:
 - `batch_size = 16`
 - `epochs = 50`
 - `validation_data = (X_val, y_val)`
 - `verbose = 1` (exibe progresso por época)

7.4. Resultados do Treinamento

Curvas de treinamento:

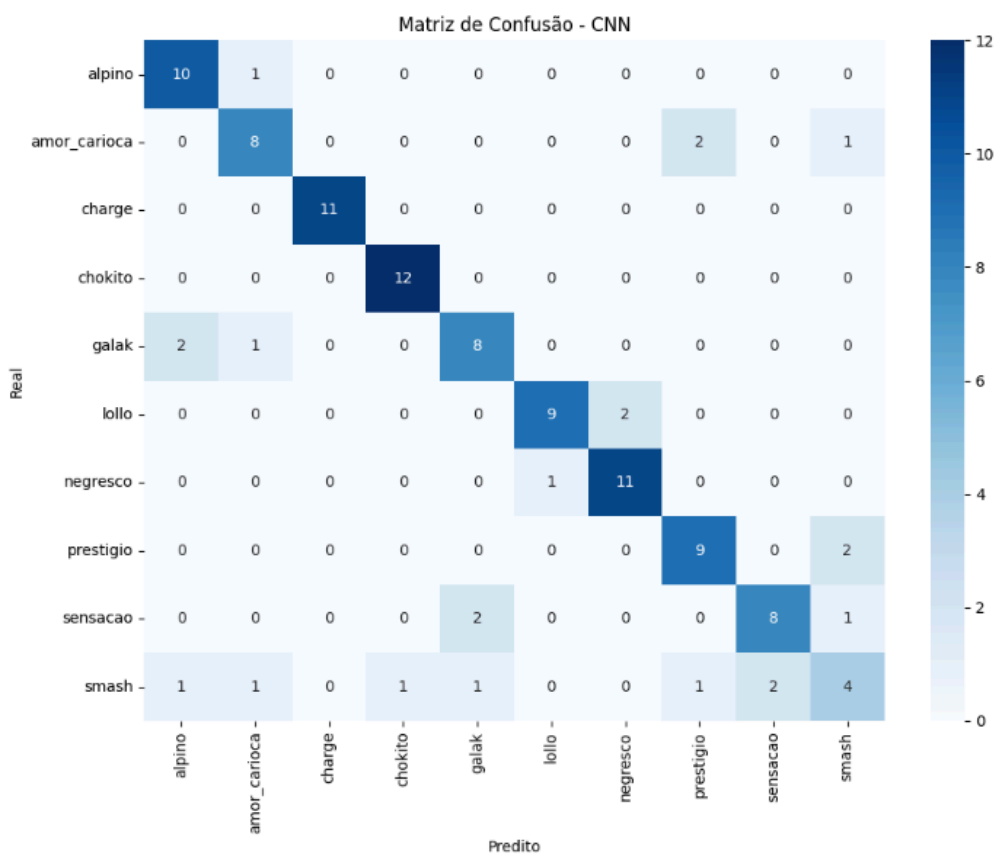


(Tabela - Perda do Modelo)



(Tabela - Acurácia do Modelo)

As tabelas acima apresentam uma perda do modelo baixa que se estabiliza com o tempo e uma acurácia muito boa de 0.84.



(Matriz de Confusão da CNN)

A Matriz de Confusão apresenta a distribuição de acertos e erros por classe no conjunto de testes.

- A classe Smash é a que contém mais confusões, possivelmente causadas pelas cores azuis presentes em outros chocolates, já o caso de alpino e galak ainda concentram uma grande taxa de confusão entre eles por terem formas muito parecidas.
- Outras classes menos frequentes, como chokito e negresco apresentam acurácia por classe acima de 90 %.

7.5. Relatório de Classificação

Relatório de Classificação:				
	precision	recall	f1-score	support
alpino	0.83	0.91	0.87	11
amor_carioca	0.82	0.82	0.82	11
charge	0.91	0.91	0.91	11
chokito	0.92	1.00	0.96	12
galak	0.83	0.91	0.87	11
lollo	0.91	0.91	0.91	11
negresco	0.92	0.92	0.92	12
prestigio	0.80	0.73	0.76	11
sensacao	0.75	0.82	0.78	11
smash	0.62	0.45	0.53	11
accuracy			0.84	112
macro avg	0.83	0.84	0.83	112
weighted avg	0.83	0.84	0.83	112

(Tabela - Classificação da Precisão do Modelo)

O modelo apresentou uma acurácia geral de 84 % no conjunto de teste, indicando que 84 % das amostras foram classificadas corretamente entre as 10 categorias de bombons. Através desse Relatório de Classificação é possível ter uma noção da qualidade do modelo.

7.6. Predições

No estágio de inferência, aplicamos o modelo treinado a um conjunto de 8 imagens de bombons, coletadas de diferentes fontes online, todas com fundo branco para uniformizar o cenário. Verificamos que ocorreram algumas confusões e acertos no modelo no estágio de testes, como é visto abaixo:

```

Predizendo imagens em: /content/drive/MyDrive
1/1 _____ 0s 51ms/step
alpinoBranco.png → galak
1/1 _____ 0s 60ms/step
galakBranco.png → galak
1/1 _____ 0s 52ms/step
chokitoBranco.png → smash
1/1 _____ 0s 74ms/step
chargeBranco.png → charge
1/1 _____ 0s 77ms/step
lolloBranco.png → lollo
1/1 _____ 0s 104ms/step
negrescoBranco.png → negresco
1/1 _____ 0s 101ms/step
smashBranco.png → smash
1/1 _____ 0s 164ms/step
amorcariocaBranco.png → amor_carioca

```

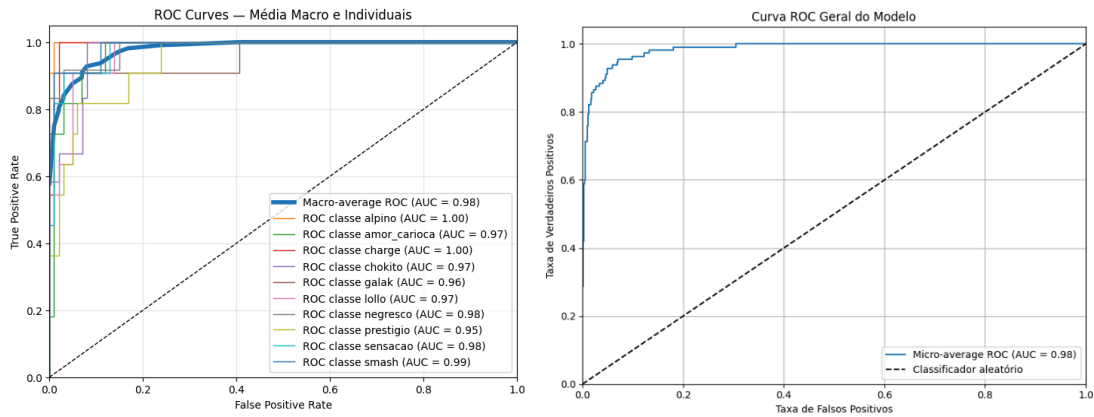
(Imagem - Aplicação do Modelo)

- Alpino confundiu com galak, algo comum por conta da semelhança das embalagens;
- Galak correto;
- Chokito foi confundido com o smash, por conta de formato e cores serem parecidos
- Charge correto;
- Lollo correto;
- Negresco correto;
- Smash correto;
- Amor Carioca correto;

7.7. Comparações de Curvas ROC

A curva ROC presente em ambos os casos não apresenta uma diferença muito notável, ambos demonstram uma curva semi-perfeita com um resultado final ótimo (próximo de 1).

Este comportamento é notável pelo fato de não apresentar uma diversidade grande nos dados anotados, fazendo com que o modelo fique confiante de adivinhar aquelas imagens no qual ele está “acostumado”.



(Curvas ROC - Random Forest na esquerda e CNN na direita)

8. Conclusão

Em conclusão, este trabalho apresentou resultados promissores, evidenciando o potencial da inteligência artificial em cenários nos quais todas as etapas — desde a coleta e o pré-processamento dos dados até o treinamento e a validação do modelo — são executadas de forma adequada. Ao longo do desenvolvimento, a equipe adquiriu conhecimentos práticos valiosos na implementação dos algoritmos e aprofundou a compreensão dos conceitos discutidos em sala de aula.

Embora o desempenho obtido tenha sido satisfatório, ainda há diversos pontos em que podemos avançar para tornar o sistema mais robusto e preciso. Entre as principais melhorias a serem exploradas, destacam-se:

- Refinamento do dataset inicial, ampliando a variedade de amostras e equilibrando melhor as classes;
- Ajustes nas técnicas de data augmentation, testando novos filtros, transformações geométricas e variações de intensidade;
- Experimentação com abordagens de transfer learning, utilizando modelos pré-treinados para acelerar a convergência e potencialmente aumentar a acurácia;
- Otimização de hiperparâmetros e adoção de estratégias de validação mais rigorosas, como validação cruzada e análise detalhada de curvas de aprendizado.

Essas alterações têm o objetivo de aprimorar ainda mais a qualidade das previsões e garantir que o classificador mantenha alta performance em ambientes visualmente mais desafiadores.

9. Referências

1. **Data Augmentation em Visão Computacional**
SHORTEN, C.; KHOSHGOFTAAR, T. M. A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data*, v. 6, n. 1, p. 1-48, 2019.
(Fundamentação teórica sobre técnicas de aumento de dados)
2. **Análise de imagens digitais: princípios, algoritmos e aplicações.** PEDRINI, Hélio; SCHWARTZ, William Robson. São Paulo, SP: Thomson, 2008. 508 p. ISBN 9788522105953.
3. **Pré-processamento de Imagens**
GONZALEZ, R. C.; WOODS, R. E. *Digital Image Processing*. 4th ed. Pearson, 2018.
(Capítulos 3 e 4 sobre transformações de intensidade e filtragem espacial)
4. SOLOMON, Chris; BRECKON, Toby. **Fundamentos de processamento digital de imagens: uma abordagem prática com exemplos em Matlab.** 1. ed. Rio de Janeiro, RJ: LTC, 2013. xvi, 289 p. ISBN 9788521623472.