

Gaussian Naive Bayes

Analizando acidentes de trânsito

Equipe

Conrado Einstein

Charles Lima

André Lucas

Paulo Vinícius

Jorge Lucas

Porquê o Gaussian Naive Bayes (GNB) ?

01

Simplicidade

Modelo fácil de entender, permitindo explicar como as decisões são tomadas a partir das features das imagens

02

Alta Performance

Treinamento e classificação rápidos, com pouca demanda de processamento e memória

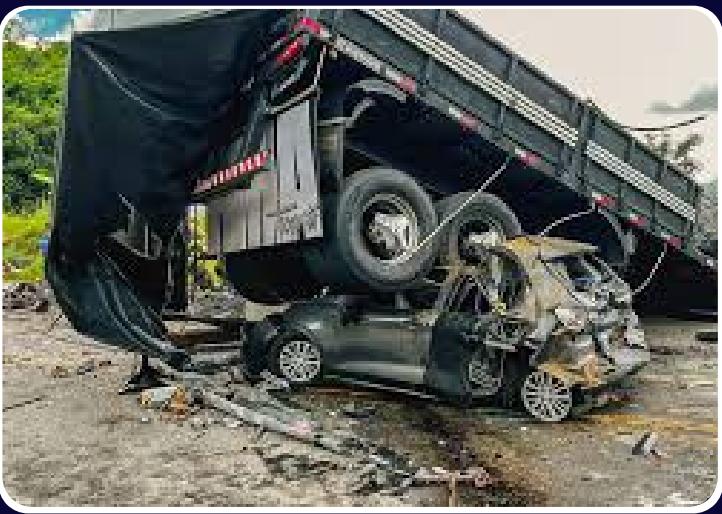
03

Adequado para Poucos Dados

Apresenta desempenho aceitável mesmo com conjuntos de dados reduzidos

Features são características numéricas extraídas das imagens que representam informações relevantes para o modelo realizar a classificação.

Exemplos de classes



01

Acidente de trânsito grave

Caminhão esmaga carro



02

Acidente de trânsito moderado

Barruada na porta do carro



03

Não é acidente

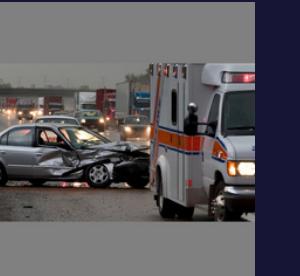
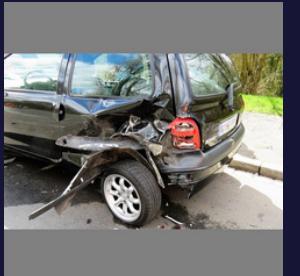
Carros estacionados em fileira

Cada imagem do conjunto de dados é associada a uma dessas classes, e o objetivo do modelo é aprender a distinguir entre elas com base nas features extraídas

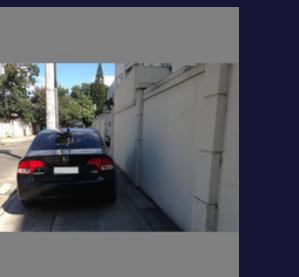
Base de dados tratada



Acidente de trânsito grave

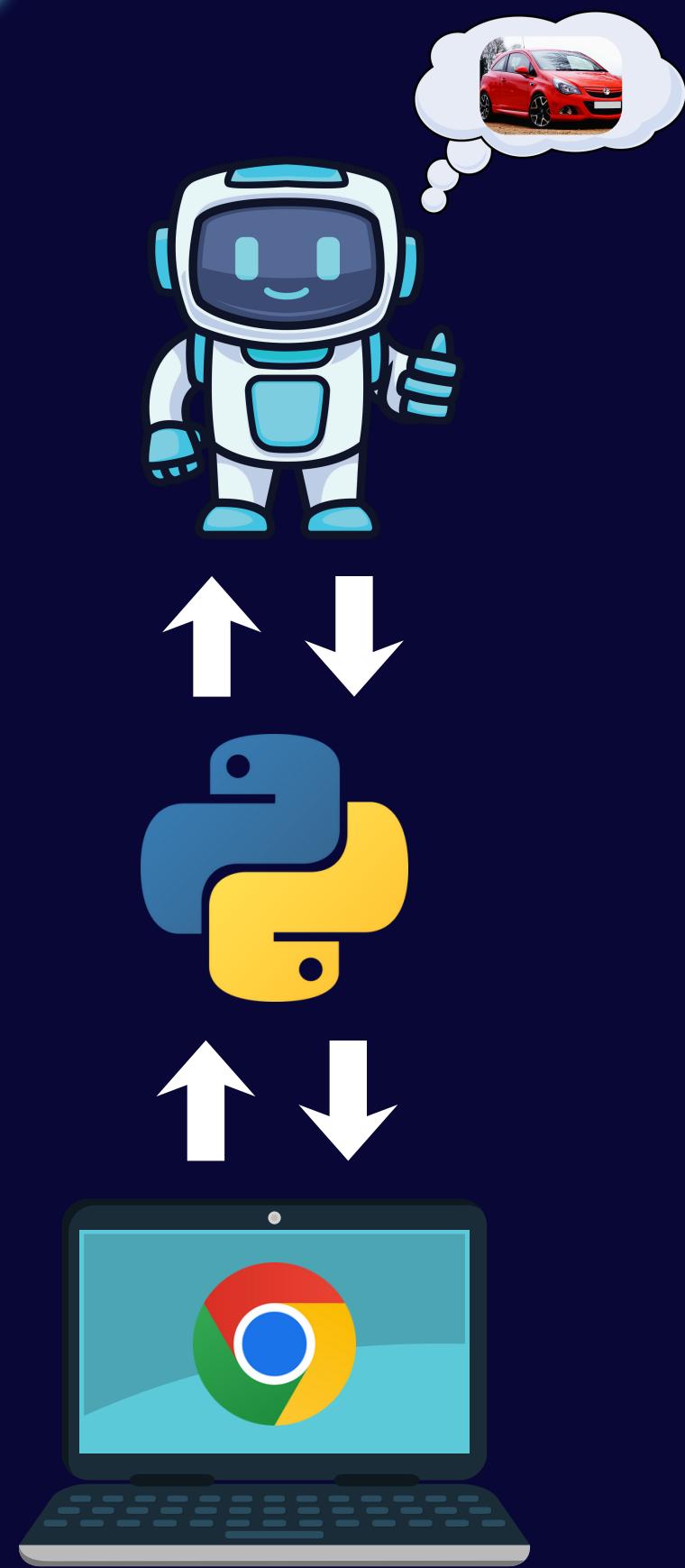


Acidente de trânsito moderado



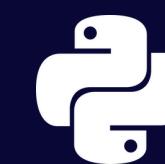
Não é acidente

A Aplicação



Modelo GNB

Treinado utilizando a biblioteca Scikit-Learn, e exportado com a Joblib, foi importado no backend para realizar as análises



Backend

Recebe a imagem enviada pelo frontend, passa o arquivo para o modelo GNB analisar, e então retorna o resultado para o frontend



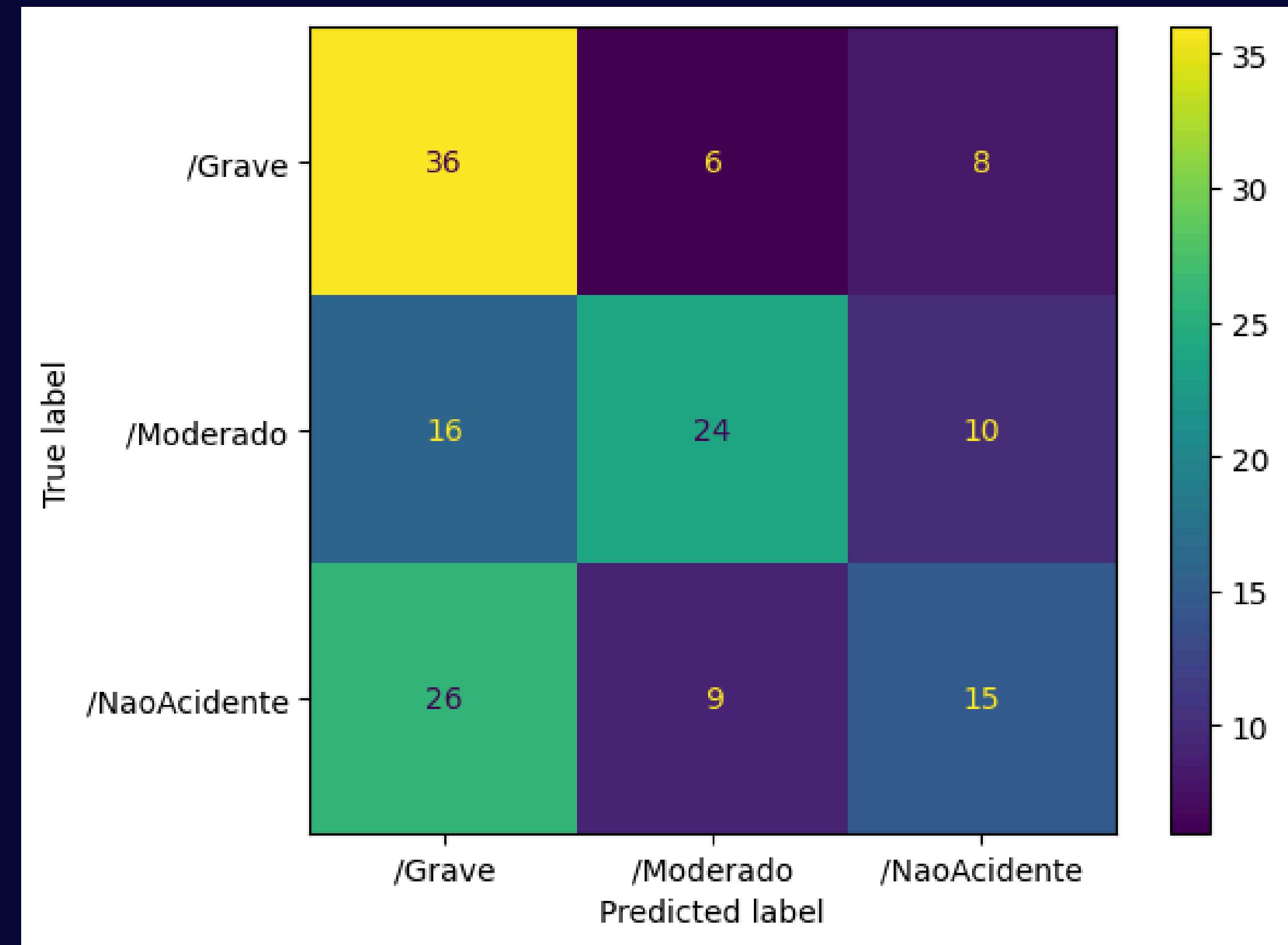
Frontend

Utilizando React, desenvolvemos um simples formulário para receber a imagem do usuário, e após o processamento retornar a classe que a pertence

Modelo

- GaussianNB.
- Classes: Grave, moderado e não acidente.
- Base de dados: 150 imagens (50 de cada classe).
- Proporção de dados:
 - 80% para treinamento
 - 20% para testes
- Padronização: Scaler
- Redimensionando: 64 x 64

Modelo - Matriz de confusão

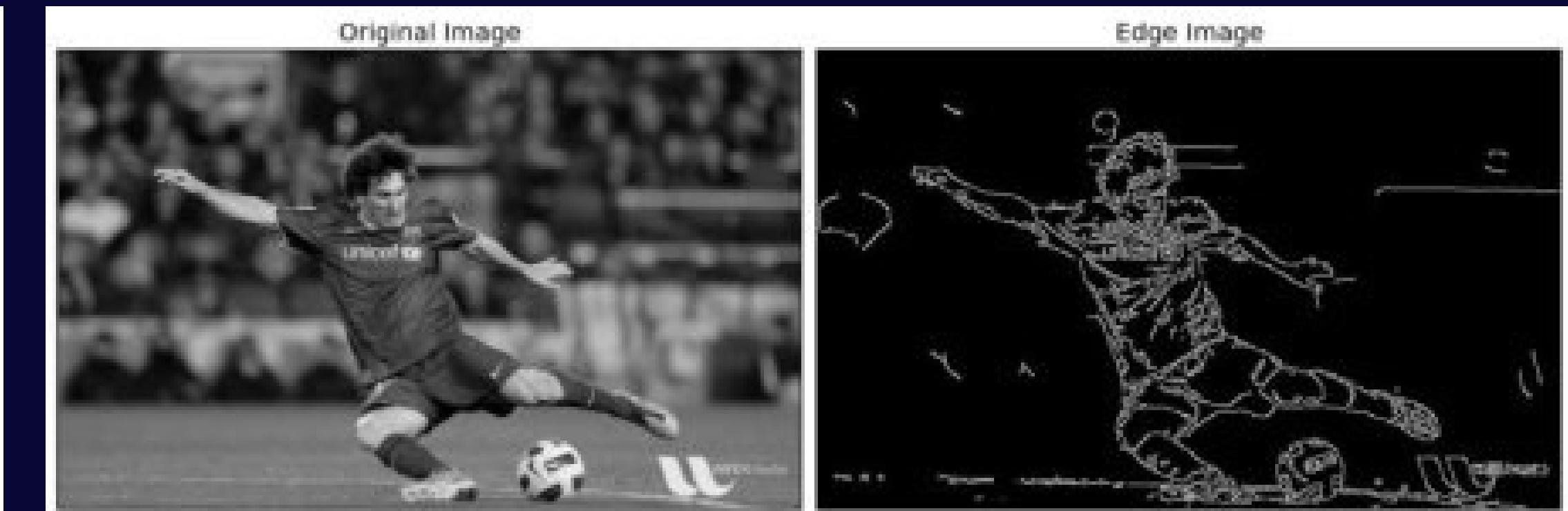
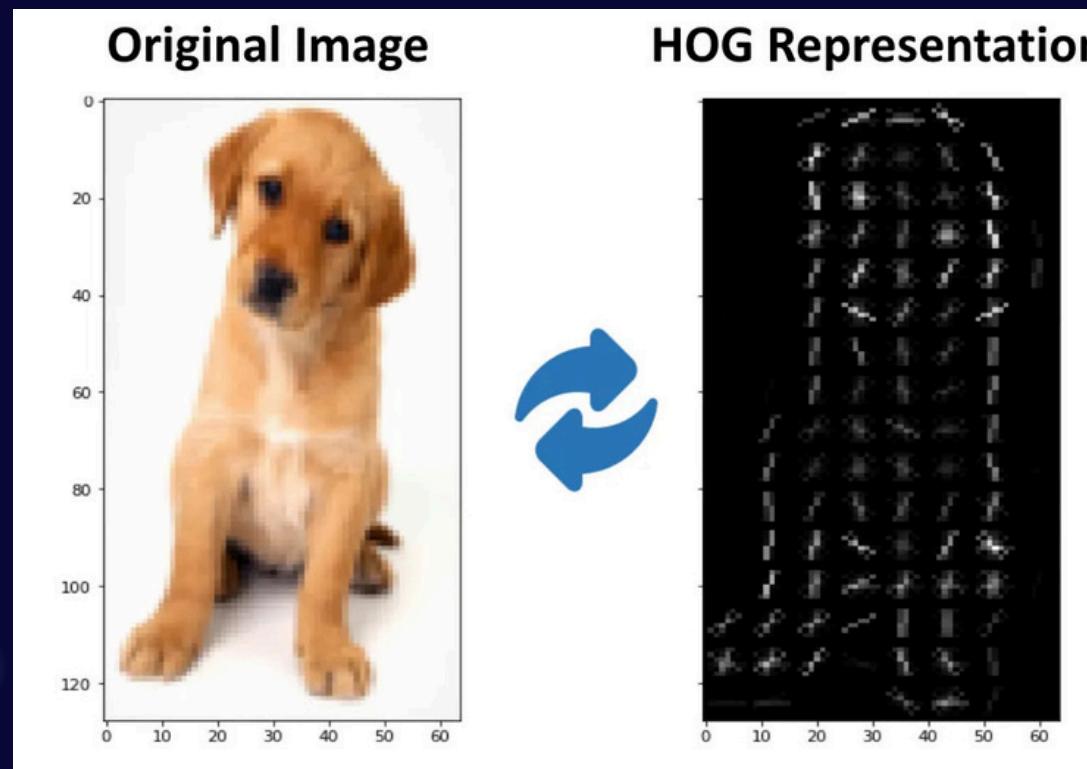


Modelo - Resultados

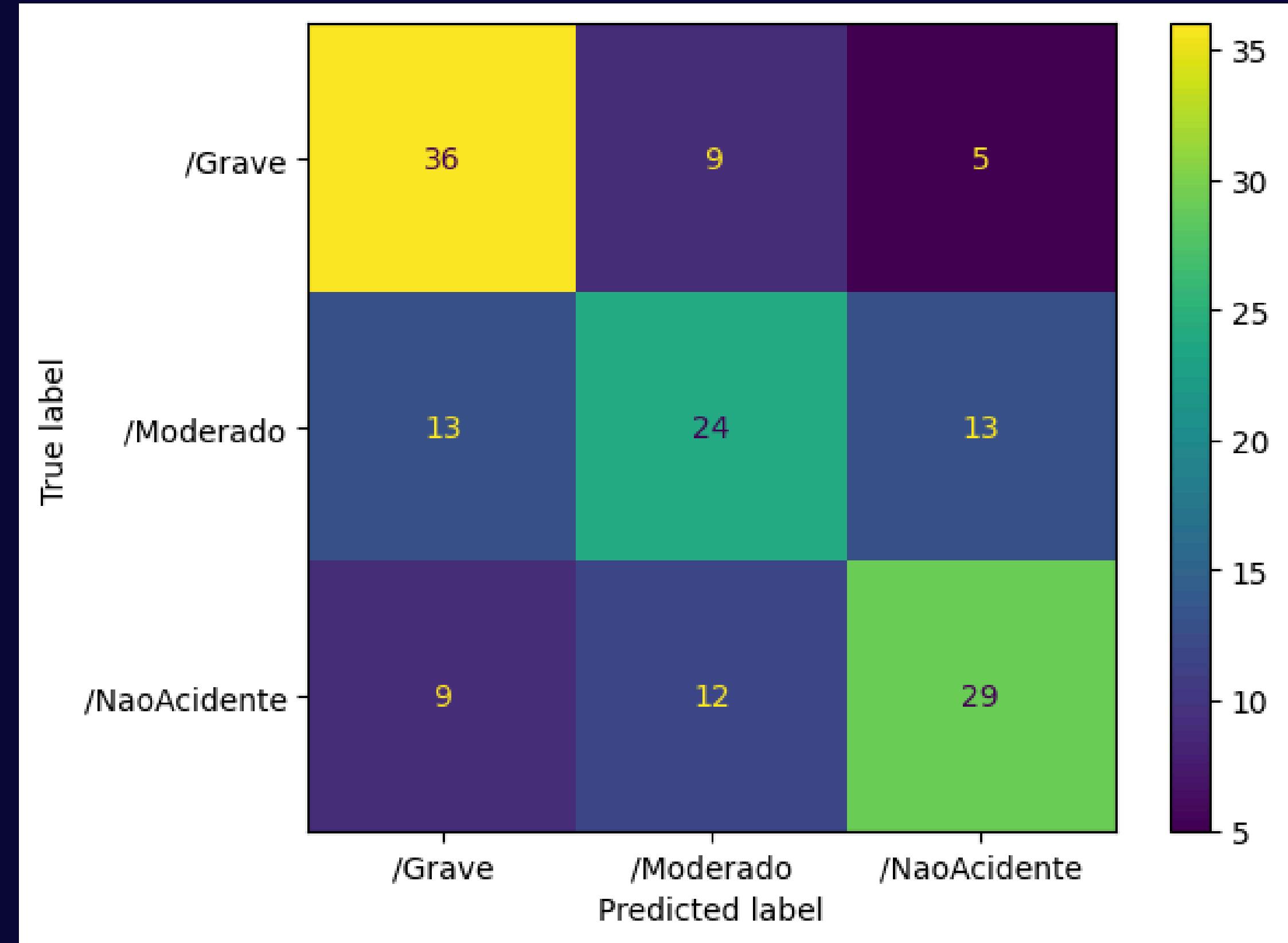
	precision	recall	f1-score	support
/Grave	0.64	0.64	0.64	14
/Moderado	0.50	0.83	0.62	6
/NaoAcidente	0.33	0.20	0.25	10
accuracy			0.53	30
macro avg	0.49	0.56	0.51	30
weighted avg	0.51	0.53	0.51	30

Modelo - Mudanças

- Incorporação de novos algoritmos para aumentar a precisão:
 - HOG - Histogram of Oriented Gradients
 - Canny - Detecção de bordas
- Redimensionando: 128 x 128



Modelo - Nova matriz



Modelo - Novos resultados

	precision	recall	f1-score	support
/Grave	0.79	0.85	0.81	13
/Moderado	0.43	0.43	0.43	7
/NaoAcidente	0.67	0.60	0.63	10
accuracy			0.67	30
macro avg	0.63	0.62	0.62	30
weighted avg	0.66	0.67	0.66	30

Imagens - Inicial

Detector de Acidentes

Envie uma imagem para análise



Clique ou arraste uma imagem aqui

Imagens - Não acidente

Detector de Acidentes

Envie uma imagem para análise



Analisar Outra Imagem

Realizar Previsão

Sem acidente

Fluxo normal.

Tenho 100% de certeza em relação a essa previsão

Imagens - Acidente moderado

Detector de Acidentes

Envie uma imagem para análise



Analisar Outra Imagem

Realizar Previsão

Acidente de trânsito moderado

Autoridades de trânsito notificadas.

Tenho 100% de certeza em relação a essa previsão

Imagens - Acidente grave

Detector de Acidentes

Envie uma imagem para análise



Analisar Outra Imagem

Realizar Previsão

Acidente de trânsito grave

Serviços de emergência notificados.

Tenho 100% de certeza em relação a essa previsão

Referências

- <https://medium.com/@skillcate/histogram-of-oriented-gradients-hog-simplest-intuition-2392995f8010>
- https://docs.opencv.org/4.x/d22/tutorial_py_canny.html
- https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.GaussianNB.html
- <https://www.geeksforgeeks.org/machine-learning/gaussian-naive-bayes/>

Obrigado!