

# Detecção de Expressões Faciais

**Estudo e aplicação de uma CNN**

# Trabalho de Conclusão de Curso

- **Aluno:**
  - Marcelo Leite Lopes
- **Orientador:**
  - Rober Marccone Rosi
- **Curso:**
  - Graduação em Ciência da Computação, FAESA

# Agenda

- **Introdução**
  - Importância, problema e hipótese
- **Objetivos**
- **Metodologia**
- **Resultados**
  - Acurácia e matriz de confusão
- **Aplicação**
  - Uso da rede treinada
- **Conclusão**

# Introdução: importância

- **Por que detectar expressões faciais?**
  - Controle da dor
  - Monitoramento de pilotos, motoristas
  - Sistemas de vídeo-conferências
  - Sistemas de vigilância, controle de acesso
  - Afetividade a um produto

# Introdução: problema e hipótese

- **Problema:**

- Como implementar um método automatizado de reconhecimento e classificação das emoções dos indivíduos através da análise de suas expressões faciais?

- **Hipótese:**

- É possível treinar uma rede neural convolucional (CNN) para, a partir de fotografias de expressões faciais humanas, detectar, identificar e classificar as emoções.

# Objetivos

- **Objetivo geral:**

- Reproduzir o treinamento, teste, validação e aplicação de uma rede neural convolucional para classificar expressões faciais em imagens de faces

- **Objetivos específicos:**

- Obter e preparar um banco de imagens faciais
- Implementar um classificador de emoções utilizando uma rede neural convolucional
- Avaliar a precisão com a qual a rede neural classificou corretamente as expressões faciais
- Aplicar a rede neural treinada

# Metodologia

- Busca de **material de referência** a ser reproduzido
- Obtenção e processamento da **base de imagens**
- Definição da **infra-estrutura** necessária
- Definição da **arquitetura** da CNN
- **Treinamento**, teste e validação da CNN
- **Aplicação** da CNN treinada a uma imagem

# Referência e base de imagens

- **Deep Learning Lab: fer2013**, Kinli (2018)

- Aplicado a diversas bases de imagens
- Código relativamente simples para a CNN
- Disponível para reprodução na internet

- **Fer2013**

- Base open-source com 35.887 imagens faciais
- Utilizada em competição no Kaggle
- Permitirá comparar a acurácia da CNN a ser treinada com a acurácia obtida na competição

# Base de imagens: fer2013



Código	Emoção	Qtd. de Imagens
0	Raiva/Irritado	4.593
1	Nojo	547
2	Medo	5.121
3	Feliz	8.989
4	Triste	6.077
5	Surpresa	4.002
6	Neutro	6.198

# Infraestrutura

- **Google Colab**

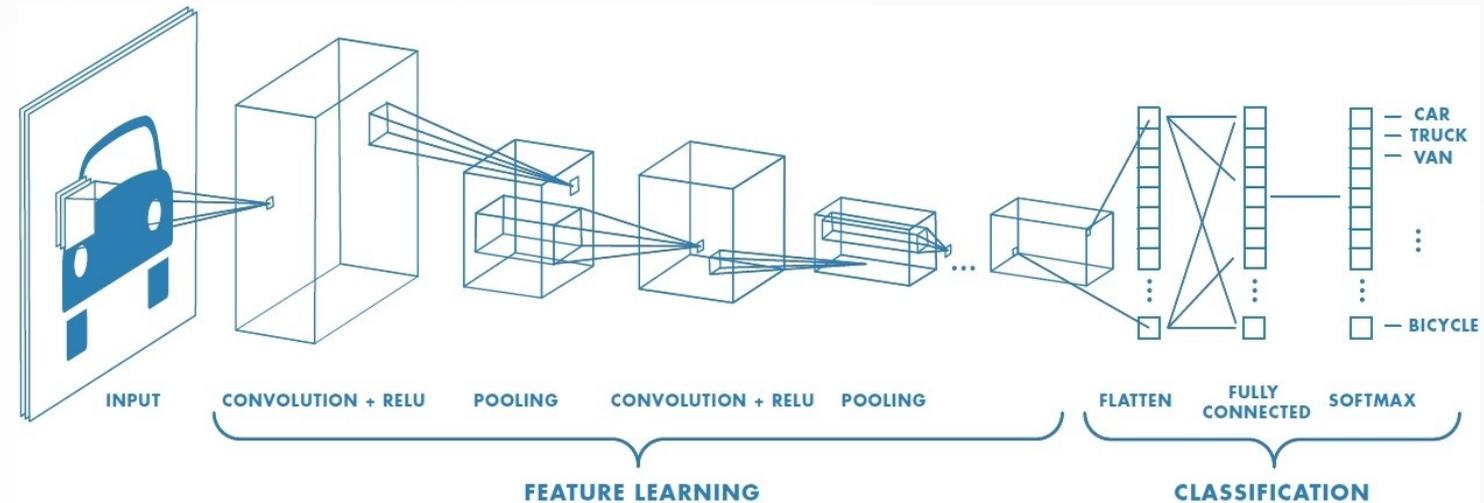
- Acesso gratuito à GPU Tesla T4 da NVIDIA
  - 20s para o treinamento de cada época, contra 2h sem GPU
- Hardware e software já prontos e disponíveis
- Python, Keras, TensorFlow, bibliotecas
- Markdown

```
!nvidia-smi
Fri Jul 10 02:20:47 2020
+-----+
| NVIDIA-SMI 450.36.06      Driver Version: 418.67      CUDA
+-----+-----+
| GPU   Name                Persistence-M| Bus-Id        Disp.A | Vol
| Fan   Temp  Perf  Pwr:Usage/Cap|                Memory-Usage | GPU
+-----+-----+-----+-----+-----+-----+
|    0  Tesla T4             off          | 00000000:00:04.0 Off  |
| N/A   75C    P0     34W / 70W | 219MiB / 15079MiB |
+-----+-----+-----+-----+-----+-----+
+-----+
| Processes:
| GPU   GI    CI          PID    Type    Process name
|      ID    ID
+-----+-----+-----+-----+-----+
| No running processes found
+-----+
```

# Arquitetura da CNN

- Kinli (2018):

- Diversas camadas de convolução, com amostragem
- 100 épocas de treinamento, com função de parada
- Uso de SOFTMAX para estimar as probabilidades



Entrada

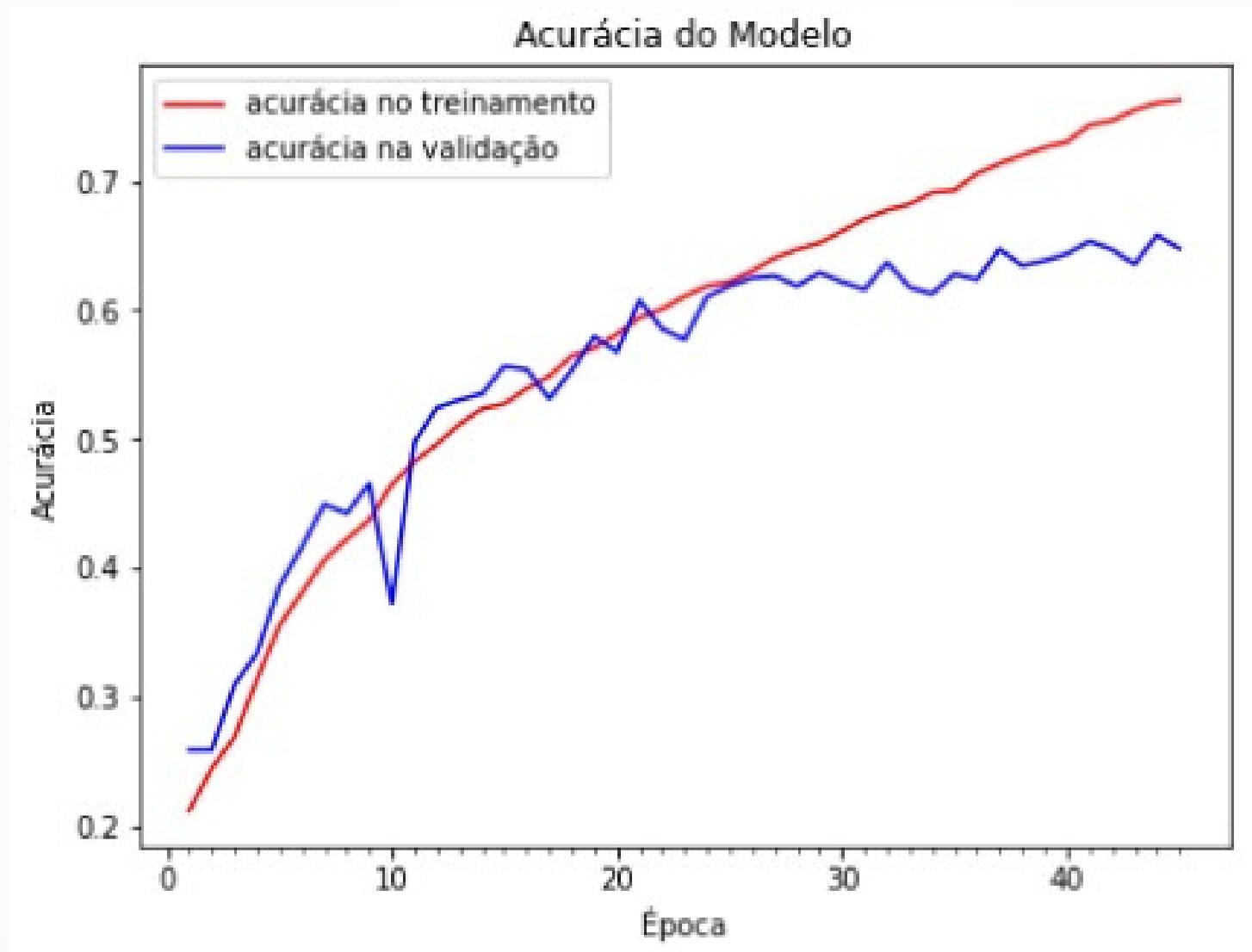
- (2 x CONV (3x3)) – MAXP (2x2) – DROPOUT (0.5)
- (2 x CONV (3x3)) – MAXP (2x2) – DROPOUT (0.5)
- (2 x CONV (3x3)) – MAXP (2x2) – DROPOUT (0.5)
- (2 x CONV (3x3)) – MAXP (2x2) – DROPOUT (0.5)
- DENSE (512) – DROPOUT (0.5)
- DENSE (256) – DROPOUT (0.5)
- DENSE (128) – DROPOUT (0.5)

Saída

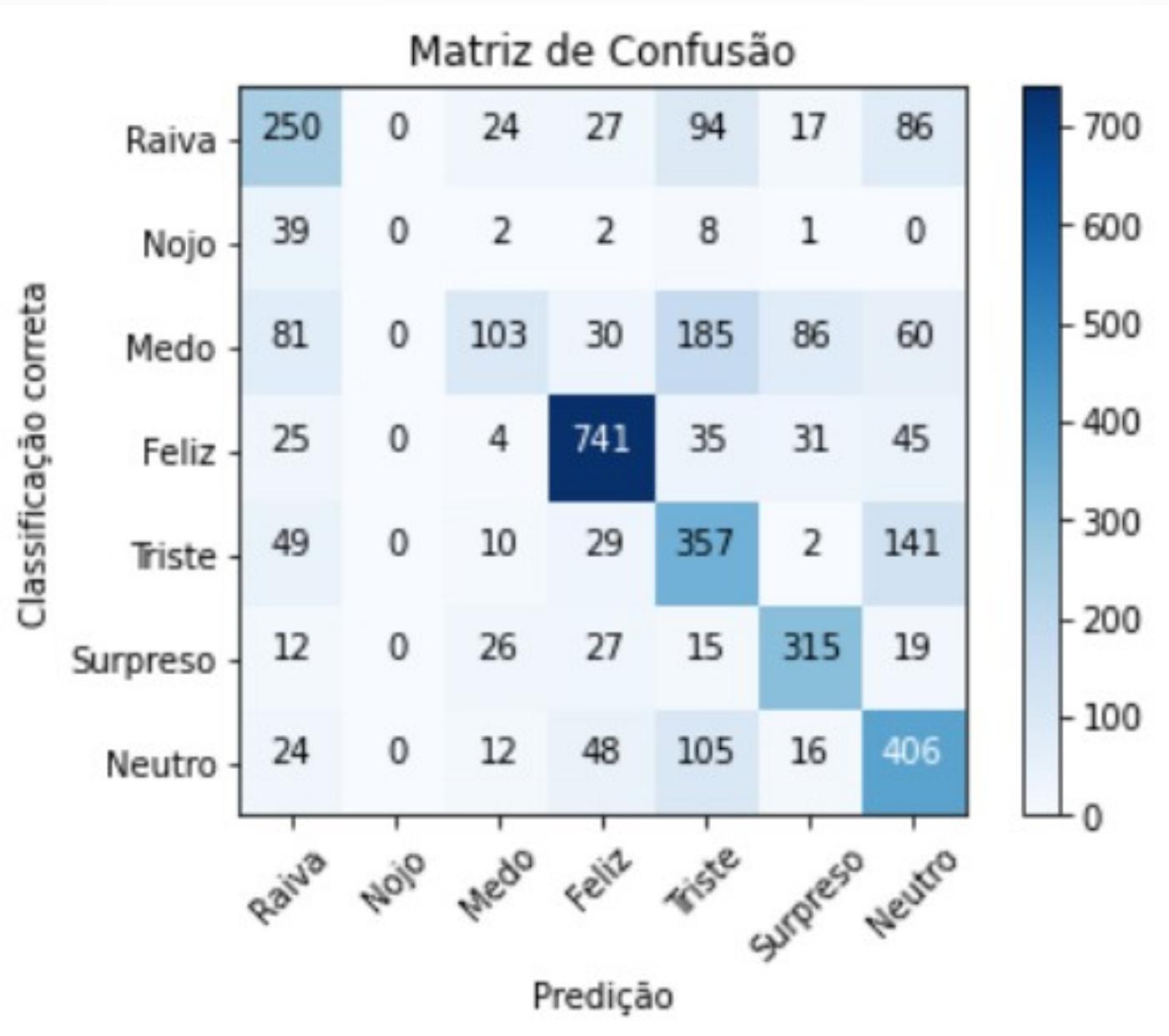
# Treinamento, teste, validação

- Treinamento/Teste
  - Divisão aleatória das imagens da fer2013 em grupos de treinamento, teste e validação
  - Função de parada se não houver mais ganho da acurácia na validação
  - Cálculo da acurácia global e uso de matriz de confusão

# Resultado: 63,97% de acurácia



# Resultado: matriz de confusão



# Aplicação: pré-processamento



Haar Cascade



Normalização  
Ajuste no tamanho  
Transformação em array  
Ajuste dimensional



# Aplicação: resultado



CNN treinada



```
0 Raiva 4.93%
1 Nojo 0.0%
2 Medo 2.92%
3 Feliz 12.07%
4 Triste 6.85%
5 Surpreso 0.41%
6 Neutro 72.81%
```

# Conclusão

- Foi possível treinar uma CNN para detectar expressões faciais com boa acurácia: 63,97%
- Apenas 3,51 pontos percentuais abaixo do 4º colocado na Kaggle
- Maior problema: incapacidade de detectar a expressão de nojo (poucas imagens?)

- Vencedores Kaggle:

Yichuan Tang:	71,13%
Yinbo Zhou:	69,27%
Maxim Milakov:	68,82%
Radu Ionescu:	67,48%

**Obrigado!**