

# Análise de Imagens de Mancha Bacteriana do Tomateiro

Maiara Oliveira  
Gustavo Rodrigues

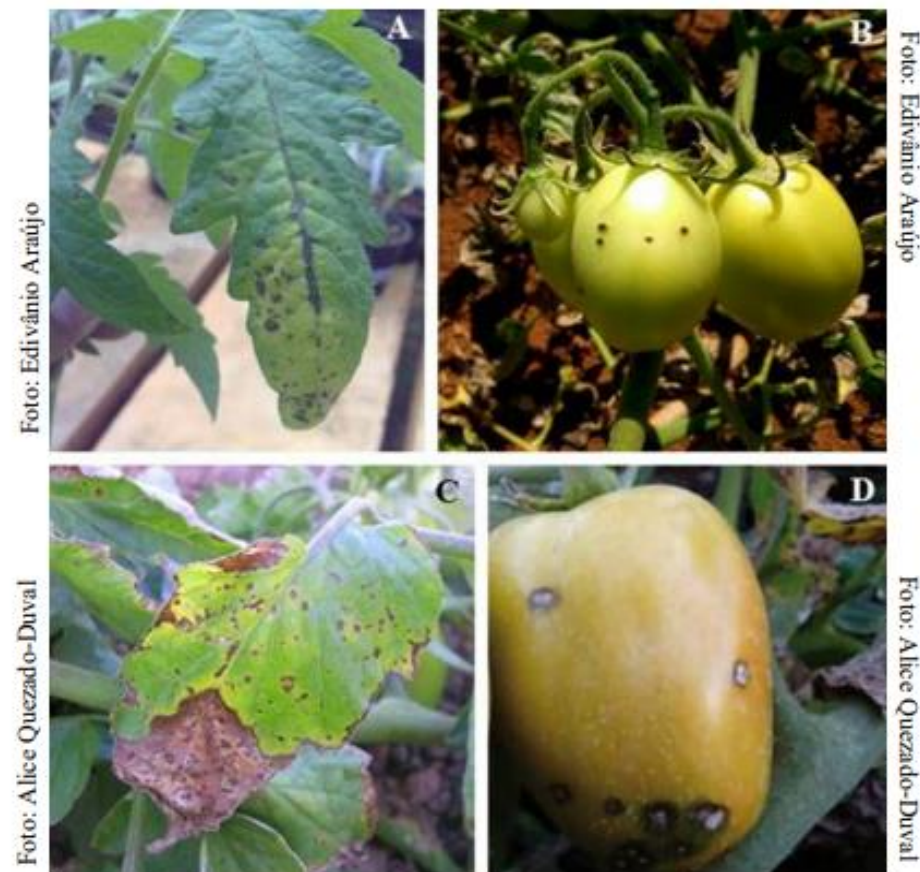


# Mancha Bacteriana do Tomateiro

- Espécies pertencentes ao gênero *Xanthomonas*;
- Relatada em praticamente todas as regiões produtoras de tomate no mundo
- Uma das doenças mais severas da cultura.
- Atinge toda a parte aérea da planta
  - Reduz a produtividade
  - Queda de flores em formação
  - Perda do valor comercial dos frutos.

# Mancha Bacteriana do Tomateiro

- Pequenas áreas encharcadas com formato irregular (Figura 1A);
- Posteriormente, formam-se lesões necróticas maiores com halos amarelados (Figuras 1C);
  - Sintomas são primeiramente observados nas folhas baixas
- Em frutos, aparecem manchas levemente elevadas (Figura 1B),
- Posteriormente de coloração marrom escura com centro deprimido (Figura 1D),



**Figura 1.** Sintomas de mancha-bacteriana, causada por *Xanthomonas* spp., em folhas e frutos de tomateiro. A – sintomas iniciais caracterizados por lesões encharcadas e pequenas; B – lesões levemente elevadas em frutos; C – lesões com aspecto mais seco e queima das folhas; D – lesões circulares com centro deprimido, nos frutos.

# Alternativas para otimizar a avaliação da planta

---

- **Problemática:**

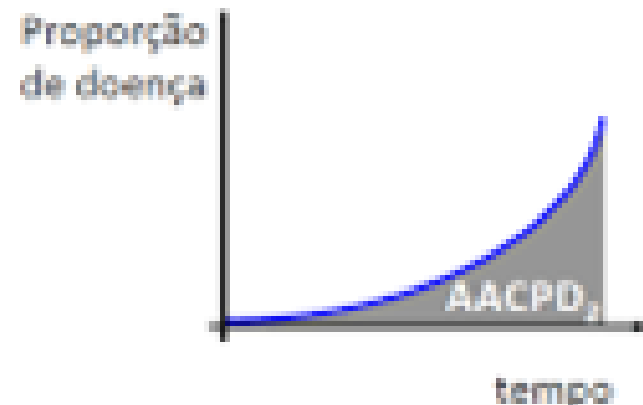
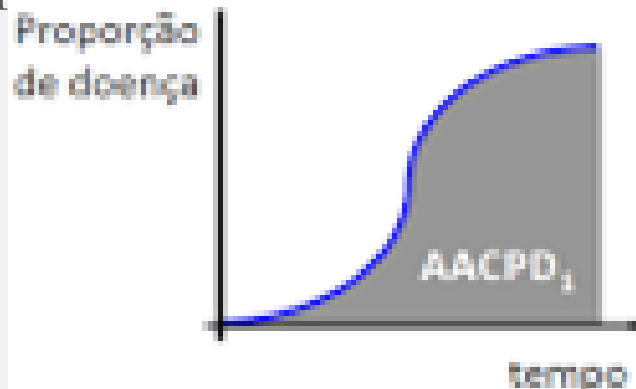
A partir desse método, algumas informações podem ser perdidas, pois a planta inteira não é avaliada e sim apenas amostras de algumas folhas da parcela.

# Alternativas para otimizar a avaliação da planta

- **Alternativa:**
- Avaliações realizadas mais de uma vez por ciclo, por exemplo, 7, 14 e 21 dias após a inoculação;
- Retirar foto da planta inteira e estabelecer uma área abaixo da curva de progresso da doença (AACPD).

$$AACPD = \Sigma \left[ \frac{(y_1 + y_2)}{2} * (t_2 - t_1) \right]$$

- Em que:  $y_1$  e  $y_2$  são duas avaliações consecutivas realizadas nos tempos  $t_1$  e  $t_2$ , respectivamente.





# Alternativas para otimizar a avaliação da planta

- **Alternativa:**
- A AACPD é um resumo quantitativo útil da intensidade da doença ao longo do tempo, esses modelos mostraram maior eficiência na determinação de danos causados por doenças foliares, quando comparados a outros modelo;
- Técnicas de fenotipagem de alto rendimento, ajudam no estudo de doenças de plantas, por viabilizarem o uso de variáveis da área foliar sadia na quantificação de danos.

# Alternativas para otimizar a avaliação da planta

- **Alternativa:**
- Desvantagem :
  - Mais avaliações seriam realizadas,
  - Mais mão de obra,
  - Capacidade computacional maior, pois o número de imagens obtidas seria alto.
- Porém com as técnicas de ML essa desvantagem pode ser minimizada.

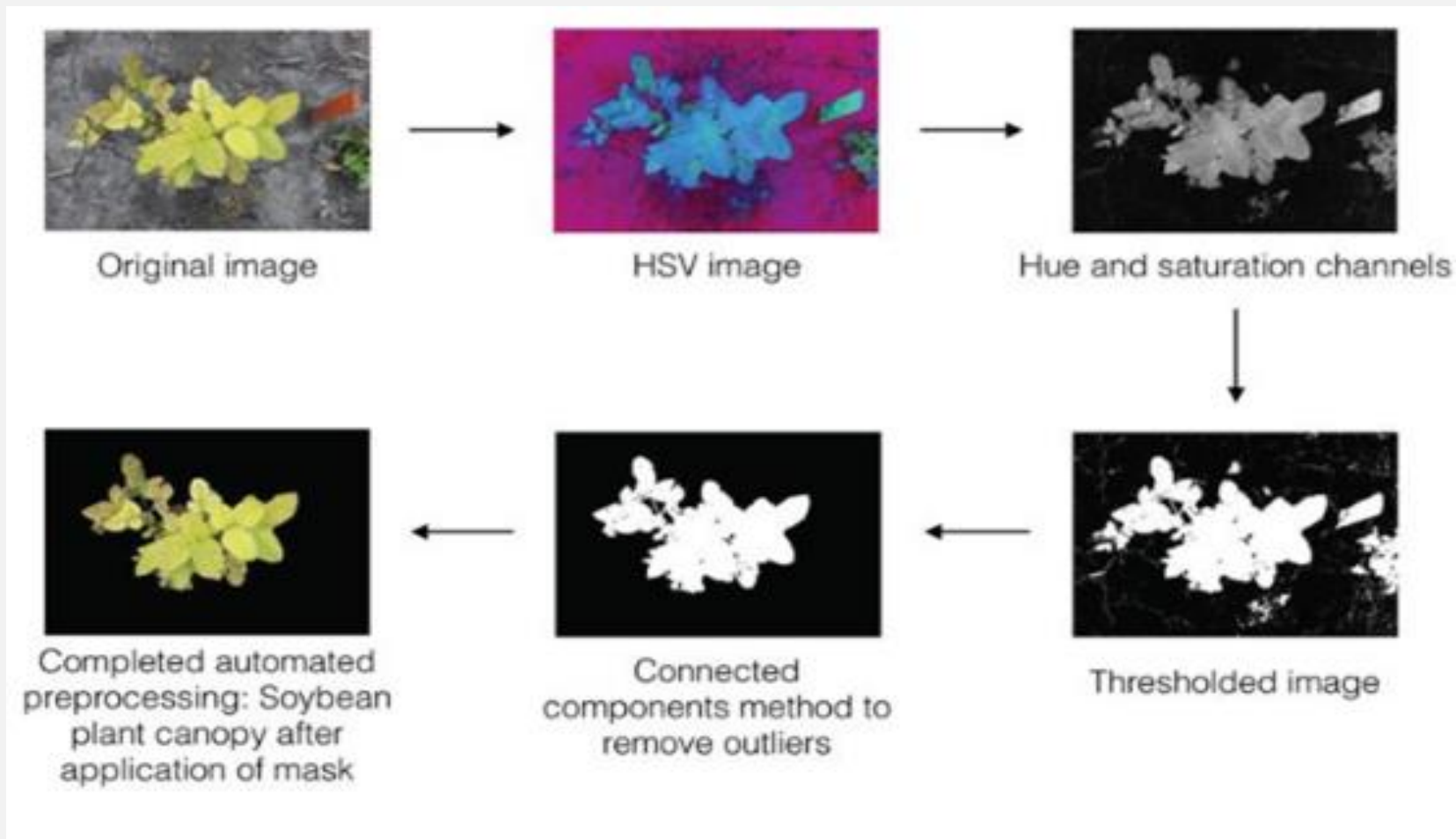
# Alternativas para otimizar a avaliação da planta

- **Primeiro Passo: Pré-processamento de imagens**
- Conversão para RGB: segmentar eficientemente o primeiro plano (planta) do segundo plano (solo);
- Método dos componentes conectados para remover os outliers e o ruído da imagem.
  - Identificação de grupos de pixels conectados, rotulando-os e identificando o maior componente conectado (a planta) e removendo outros componentes conectados que contivessem menos pixels que o principal.



# Alternativas para otimizar a avaliação da planta

- Primeiro Passo: Pré-processamento de imagens



# Alternativas para otimizar a avaliação da planta

- **Segundo Passo: Extração de recursos**
- A mancha bacteriana do tomateiro é caracterizada pela necrose amarronzada na folha portanto, as mudanças de cor seriam usadas como a medida de severidade.

$$Severidade = \frac{\text{Área Marrom}}{\text{Área Total}} * 100$$

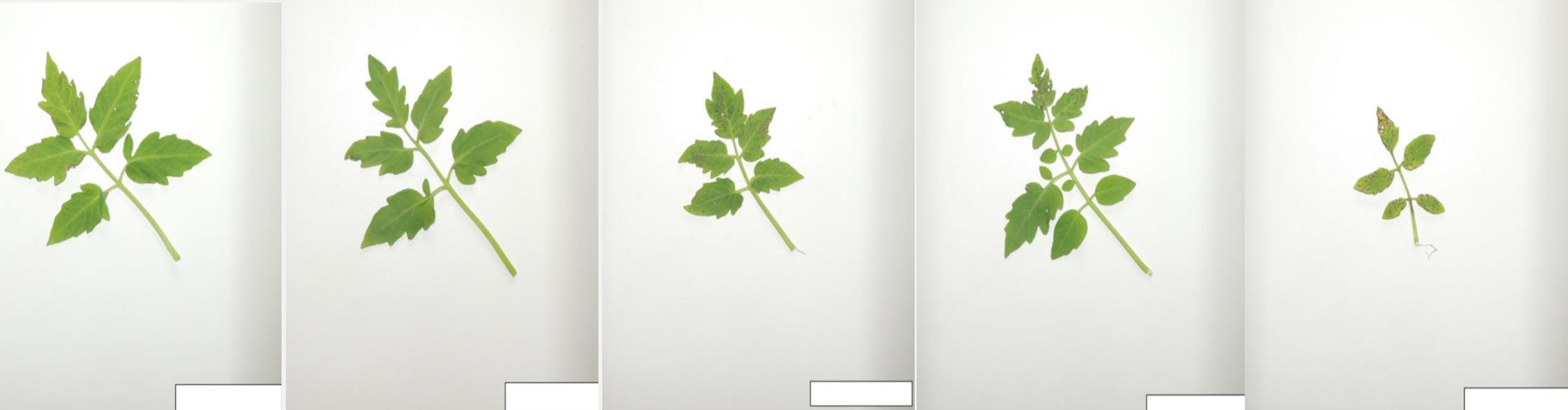
# Como a AM poderia ser utilizada para identificar diferenças na morfologia das folhas

- **Problemática:**

Como as diferenças na morfologia das folhas de cada genótipo podem ser levadas em conta na construção do modelo.

# Como a AM poderia seria ser utilizada para identificar diferenças na morfologia das folhas

- **Problemática:**
- Se esse fator não for levado em conta, folhas pequenas que possuem poucos sintomas, teriam maior severidade do que uma folha grande que possui muitos sintomas.



# Como a AM poderia seria ser utilizada para identificar diferenças na morfologia das folhas

---

- **Alternativa:**
- Uso de técnicas de aprendizagem de maquina para separar os genótipos em grupos de acordo com a morfologia foliar e depois disso, estabelecer a metodologia para delimitar a área infectada pela doença e a área sadia.

# Como a AM poderia seria ser utilizada para identificar diferenças na morfologia das folhas

- Alternativa:

Original papers

## Deep learning for plant identification using vein morphological patterns




Guillermo L. Grinblat, Lucas C. Uzal  , Mónica G. Larese, Pablo M. Granitto

 [Show more](#)

<https://doi.org/10.1016/j.compag.2016.07.003>

[Get rights and content](#)

## Using Deep Learning for Image-Based Plant Disease Detection

 Sharada P. Mohanty<sup>1,2,3</sup>,  David P. Hughes<sup>4,5,6</sup> and  Marcel Salathé<sup>1,2,3\*</sup>

<sup>1</sup>Digital Epidemiology Lab, EPFL, Geneva, Switzerland

<sup>2</sup>School of Life Sciences, EPFL, Lausanne, Switzerland

<sup>3</sup>School of Computer and Communication Sciences, EPFL, Lausanne, Switzerland

<sup>4</sup>Department of Entomology, College of Agricultural Sciences, Penn State University, State College, PA, USA

<sup>5</sup>Department of Biology, Eberly College of Sciences, Penn State University, State College, PA, USA

<sup>6</sup>Center for Infectious Disease Dynamics, Huck Institutes of Life Sciences, Penn State University, State College, PA, USA

## An efficient leaf recognition algorithm for plant classification using support vector machine

**3 Author(s)**

C. Arun Priya ; T. Balasaravanan ; Antony Selvadoss Thanamani [View All Authors](#)

15  
Paper  
Citations

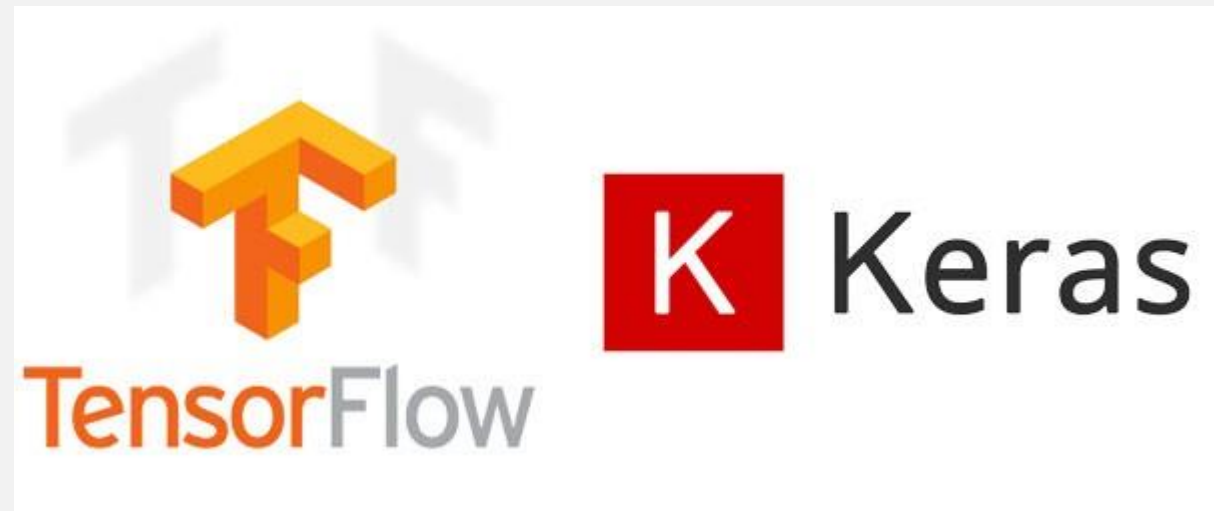
964  
Full  
Text Views





# AM utilizando o Keras com Tensorflow

- O API (application program interface) Keras utilizando as libraries de AM do Tensordflow é tem potencial de construir CNNs (redes neurais convolucionais) a partir de dados de imagem
- Bases
  - Configurar as camadas da CNN
  - Separar dados de imagem em treino, validação e teste
  - Compilar o modelo, treinar a rede (treino e validação)
  - Testar o modelo treinado
- Processo extremamente tecnicista
  - Porém, existem redes pré-configuradas disponíveis na internet



# Keras-Tensorflow: Pré-requisitos

- Keras 2.0 e Tensorflow
- Python
  - Recomendado console Jupyter-notebook para testar blocos de código
- Recomendado: GPU Nvidia com CUDA® Compute Capability 3.5 ou superior (NVIDIA GeForce GTX 1050)
  - Driver Nvidia 410 (ou superior), CUDA Toolkit 10.0, cudNN 7 (ou superior)
- Uma série de pacotes, a maioria pode ser obtido no repositório público de python (pip)

# Utilizando o Keras Tensorflow

- CNN pré-configurada para distinguir vestuário e cor das peças (<https://www.pyimagesearch.com/2018/05/07/multi-label-classification-with-keras/>)
  - Serviu de base para este trabalho. Possui códigos generalistas
  - Como o código é generalista, necessita de pouca intervenção do usuário e pode ser aplicado a diversos conjuntos de dados
    - Basicamente, basta seguir a estrutura das pastas ao colocar o conjunto de imagens
- CNN com três camadas: width, height e depth
  - smallerVGGNet, versão simplificada de VGGNet (Simonyan e Zisserman, 2014, <https://arxiv.org/pdf/1409.1556/>)

# smallerVGGNet no Keras Tensorflow

- Treinar o modelo
  - `python train.py --dataset dataset --model xxxx.model --label mlb.pickle`
  - dataset = pasta contendo subpastas com os arquivos de imagens a serem treinados
    - Neste caso, separamos o conjunto de treino pelo nível de infecção: severe, moderate e light
  - Único parâmetro a ser controlado: `batch_size`, não pode ser maior do que a amostra
  - A rede por default re-escala as imagens para 96x96. É possível alterar este parâmetro sem comprometer a rede, mas o consumo de memória aumenta
  - 75 epochs

# Resultado do modelo gerado pela smallerVGGNet no Keras Tensorflow

- O esperado era que enquanto a train\_loss diminuísse a cada epoch, a val\_loss aumentasse
  - A val\_loss possivelmente é aberrante pois o conjunto de dados é muito pequeno
    - Em tutoriais disponíveis na internet, 2000 imagens é considerado um conjunto pequeno
- Porém, é possível rodar múltiplas vezes e amostrar um modelo com val\_loss próxima de train\_loss e menor do que train\_acc e val\_acc



# Utilizando o modelo para prever imagens sem label

- python classify.py --model xxxx.model --label mlb.pickle --image local\_da\_imagen/imagen.jpg

7835.jpg 7872.jpg 7889.jpg 7901.jpg 7911.jpg 7912.jpg 7928.jpg 7932.jpg 7944.jpg

**severe: 100.00%**  
**moderate: 35.26%**

(x=186, y=4) ~ R:245 G:244 B:240

```
gustavo@gustavo-Aspire-VX5-591G: ~/Desktop/machine_learning_topicos/keras-multi-label
File Edit View Search Terminal Help
2019-06-10 22:24:46.797517: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1003] 0: N
2019-06-10 22:24:46.797731: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1115] Created TensorFlow device (/job:localhost/replica:0/task:0/device:GPU:0 with 3380 MB memory) -> physical GPU (device: 0, name: GeForce GTX 1050, pci bus id: 0000:01:00.0, compute capability: 6.1)
WARNING:tensorflow:From /usr/local/lib/python2.7/dist-packages/keras/backend/tensorflow_backend.py:3445: calling dropout (from tensorflow.python.ops.nn_ops) with keep_prob is deprecated and will be removed in a future version.
Instructions for updating:
Please use 'rate' instead of 'keep_prob'. Rate should be set to 'rate = 1 - keep_prob'.
WARNING:tensorflow:From /usr/local/lib/python2.7/dist-packages/tensorflow/python/ops/math_ops.py:3066: to_int32 (from tensorflow.python.ops.math_ops) is deprecated and will be removed in a future version.
Instructions for updating:
Use tf.cast instead.
[INFO] classifying image...
2019-06-10 22:24:50.237055: I tensorflow/stream_executor/dso_loader.cc:152] successfully opened CUDA library libcublas.so.10.0 locally
light: 0.14%
moderate: 35.26%
severe: 100.00%
```