OLIMPÍADA DE INTELIGÊNCIA ARTIFICIAL APLICADA 2025

AULÃO 3 · CAPÍTULO 3 DA IMERSÃO

Visão computacional

Na Fase 1 vocês transformaram cada foto em catorze medidas calculadas à mão. Agora ninguém calcula medida nenhuma: o modelo recebe a foto e descobre sozinho o que olhar.

COMO LER ESTE DOCUMENTO

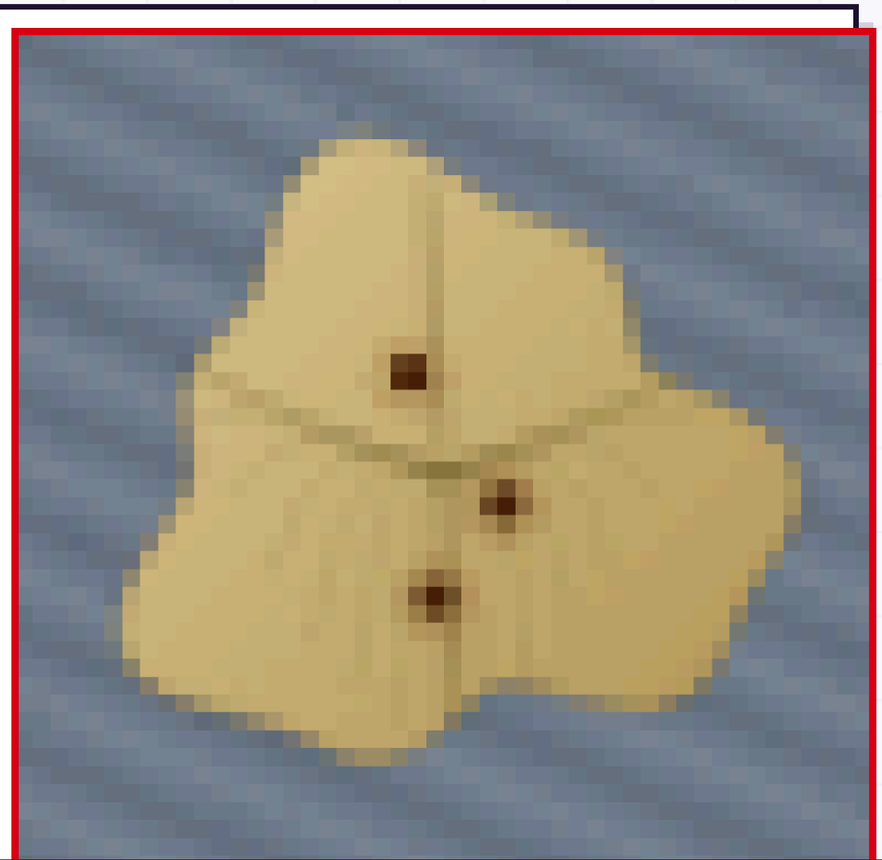
Este PDF é o registro do Aulão 3. Cada página mostra uma tela do material interativo já aberta, num estado escolhido, e a faixa de baixo explica o que está acontecendo ali. Os laboratórios aparecem em vários estados, para vocês compararem o que muda quando um controle se mexe — é justamente o que não cabe numa imagem só. O material vivo, com todos os laboratórios para mexer, está no endereço da última página.

O SALTO

De catorze números para a foto inteira

A mesma foto, nos dois tamanhos. Troquem e reparem no que some.

Fase 1 · 48 × 48 | Fase 2 · 224 × 224



FASE 1 · 48 × 48, E CATORZE MEDIDAS À MÃO

Vocês escreviam o código que media cor, textura e forma. O modelo recebia **catorze números** e nunca via a foto.

Repare: os três detalhes escuros do objeto sumiram. Nenhuma medida à mão ia recuperá-los.

O QUE MUDA · A FOTO DE 48 PIXELS

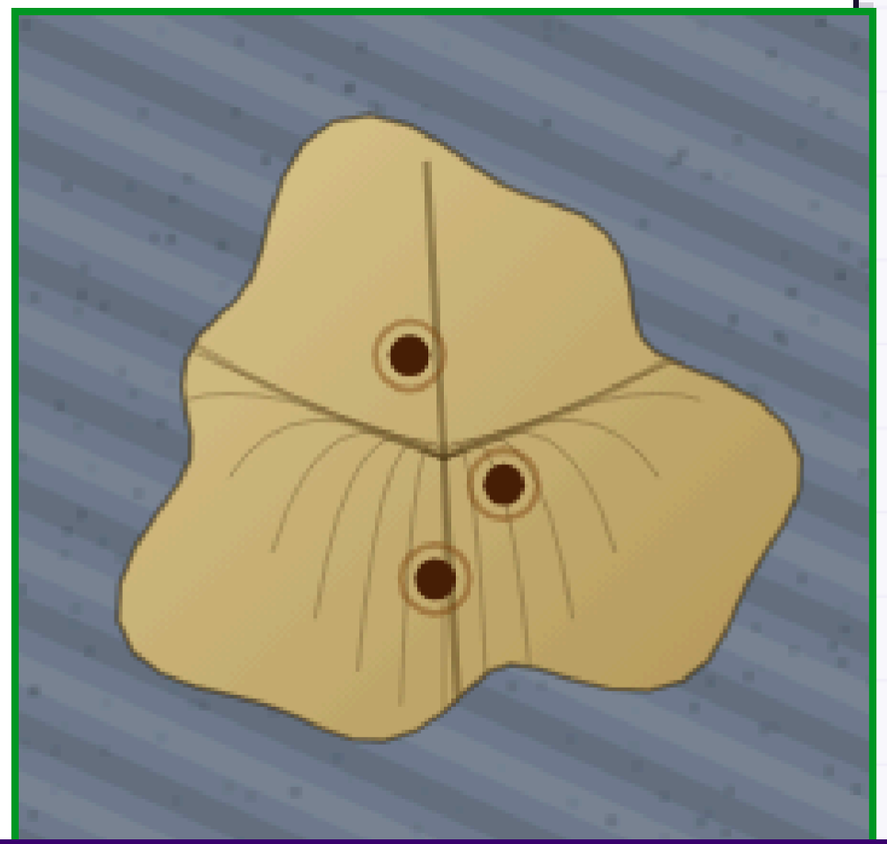
Na Fase 1 a entrada do modelo era uma linha de tabela. Aqui é esta imagem — e a primeira decisão é quantos pixels dela o modelo vai receber. Com 48 pixels de lado, o que sobra é mancha: dá para ver que tem alguma coisa, não dá para dizer o quê. Nenhum modelo acerta o que vocês também não conseguem enxergar.

O SALTO

De catorze números para a foto inteira

A mesma foto, nos dois tamanhos. Troquem e reparem no que some.

Fase 1 · 48 × 48 | Fase 2 · 224 × 224



FASE 2 · 224 × 224, E NENHUMA MEDIDA À MÃO

Ninguém calcula medida nenhuma. O modelo recebe a **foto inteira** e descobre sozinho o que olhar.

Vinte vezes mais informação — e os detalhes que decidem a classe voltaram a existir.

O QUE MUDA · A FOTO DE 224 PIXELS

A mesma foto a 224 pixels, que é a resolução em que os três modelos da trilha foram treinados. Comparem com a página anterior: é a mesma imagem, e a diferença entre as duas é toda a informação que vocês entregam ou jogam fora na primeira linha do preparo.

O MAPA DO CAPÍTULO

Os quatro blocos do desafio

A trilha é a mesma das outras duas. Cliquem num bloco para ver o que é novo aqui.

01 · EXPLORAR | 02 · CONFIGURAR | 03 · MODELAR | 04 · AVALIAR

O QUE JÁ ERA ASSIM

Abrir a tabela de medidas, contar quantos casos tem cada classe.

O QUE É NOVO AGORA

Abrir as **fotos**. Ver quantas existem de cada classe, como elas variam de luz e de enquadramento, e medir o piso: quanto tira um modelo que não aprende nada.

Novo aqui: resolução e aumento de dados. Os dois laboratórios a seguir são deste bloco.

O MAPA · BLOCO 1 · EXPLORAR

O caderno da trilha tem quatro blocos, e só um deles tem código para escrever. No primeiro vocês olham as fotos antes de treinar qualquer coisa: quantas tem cada classe, quanto vale um modelo que não aprende nada, e o que a resolução e o aumento de dados fazem. Parece tempo perdido. É a parte que decide o resto.

O MAPA DO CAPÍTULO

Os quatro blocos do desafio

A trilha é a mesma das outras duas. Cliquem num bloco para ver o que é novo aqui.

01 · EXPLORAR | 02 · CONFIGURAR | 03 · MODELAR | 04 · AVALIAR

O QUE JÁ ERA ASSIM

Escolher entre árvore, floresta e regressão, e mexer nos botões de cada uma.

O QUE É NOVO AGORA

Escolher entre **três modelos que já sabem enxergar** — e que vêm com milhões de pesos já treinados. Vocês não treinam do zero; vocês aproveitam.

> Novo aqui: pesos pré-treinados, treino em dois estágios, e a camada de saída que tem nome diferente em cada modelo.

O MAPA · BLOCO 2 · CONFIGURAR

No segundo bloco não se escreve código: escolhe-se o modelo e giram-se os controles. O que vocês decidirem aqui é o que as lacunas do Bloco 3 vão buscar. Se um valor parecer estranho lá na frente, o lugar de corrigir é este painel, não a linha de código.

O MAPA DO CAPÍTULO

Os quatro blocos do desafio

A trilha é a mesma das outras duas. Cliquem num bloco para ver o que é novo aqui.

01 · EXPLORAR | 02 · CONFIGURAR | 03 · MODELAR | 04 · AVALIAR

O QUE JÁ ERA ASSIM

Chamar `.fit` e esperar.

O QUE É NOVO AGORA

As **mesmas três células** do capítulo anterior: A prepara, B treina, C avalia. A forma não muda de trilha para trilha — só o conteúdo da célula A muda de verdade.

> É aqui que ficam as lacunas do desafio. Todas elas.

O MAPA · BLOCO 3 · MODELAR

O terceiro bloco é onde está todo o código a completar — as lacunas marcadas com quatro sublinhados. Cada uma tem, logo acima, um comentário dizendo o que aquela linha precisa fazer e onde o assunto foi tratado. A pergunta que sobra para vocês é sempre o *como*.

O MAPA DO CAPÍTULO

Os quatro blocos do desafio

A trilha é a mesma das outras duas. Cliquem num bloco para ver o que é novo aqui.

01 · EXPLORAR | 02 · CONFIGURAR | 03 · MODELAR | 04 · AVALIAR

O QUE JÁ ERA ASSIM

Olhar a métrica e a matriz de confusão.

O QUE É NOVO AGORA

A métrica e a matriz continuam. Entra uma pergunta nova, que **nenhuma métrica responde**: o modelo olhou para o objeto, ou para o fundo?

Novo aqui: mapa de atenção, e a leitura de curvas com as causas próprias da imagem.

O MAPA · BLOCO 4 · AVALIAR

O quarto bloco roda sozinho e é onde metade do relatório nasce: as curvas, a matriz de confusão e o mapa de atenção. A agrônoma da narrativa pediu uma coisa só, e é aqui que ela se responde — não um número, mas onde o modelo olhou.

01

BLOCO 1

Explorar

- Quantas fotos tem cada classe, e como elas variam
- **Resolução:** o que reaparece de 48 para 224
- **Aumento de dados:** uma foto vira quatro — e a regra de ouro

ABERTURA DO BLOCO 1

A partir daqui, os dois laboratórios que tratam das duas decisões do preparo das imagens: quanta resolução entregar, e quanto distorcer a foto no treino.

De 48 pixels para 224

Vinte vezes mais informação. Arrastem e vejam o que reaparece.

resolução 24 × 24

A FOTO CHEGA COM

0,6 mil

pixels para o modelo olhar.

OS DETALHES SUMIRAM

Os três pontos escuros do objeto viraram borrão. Nenhum modelo acha o que não está mais na imagem.

Os três pontos escuros do objeto são

LAB 13 · 24 PIXELS

O extremo de baixo. A foto virou um borrão de poucos quadrados e cabem 576 números na entrada do modelo. É rápido e é inútil: a informação que separava uma classe da outra foi destruída antes de o modelo ver qualquer coisa.

De 48 pixels para 224

Vinte vezes mais informação. Arrastem e vejam o que reaparece.

resolução **96 × 96**

A FOTO CHEGA COM

9,2 mil

pixels para o modelo olhar.

DÁ PARA ADIVINHAR

Os detalhes voltaram como manchas. O modelo acerta às vezes — e erra justamente nos casos difíceis.

Os três pontos escuros do objeto são

LAB 13 · 96 PIXELS

No meio do caminho. Já dá para reconhecer a forma geral, e o custo de processamento é bem menor que o da resolução cheia. É uma escolha defensável quando falta placa de vídeo — desde que vocês saibam que escolheram, e escrevam isso no relatório.

BLOCO 1 · LABORATÓRIO 13 ★ ARRASTE

De 48 pixels para 224

Vinte vezes mais informação. Arrastem e vejam o que reaparece.

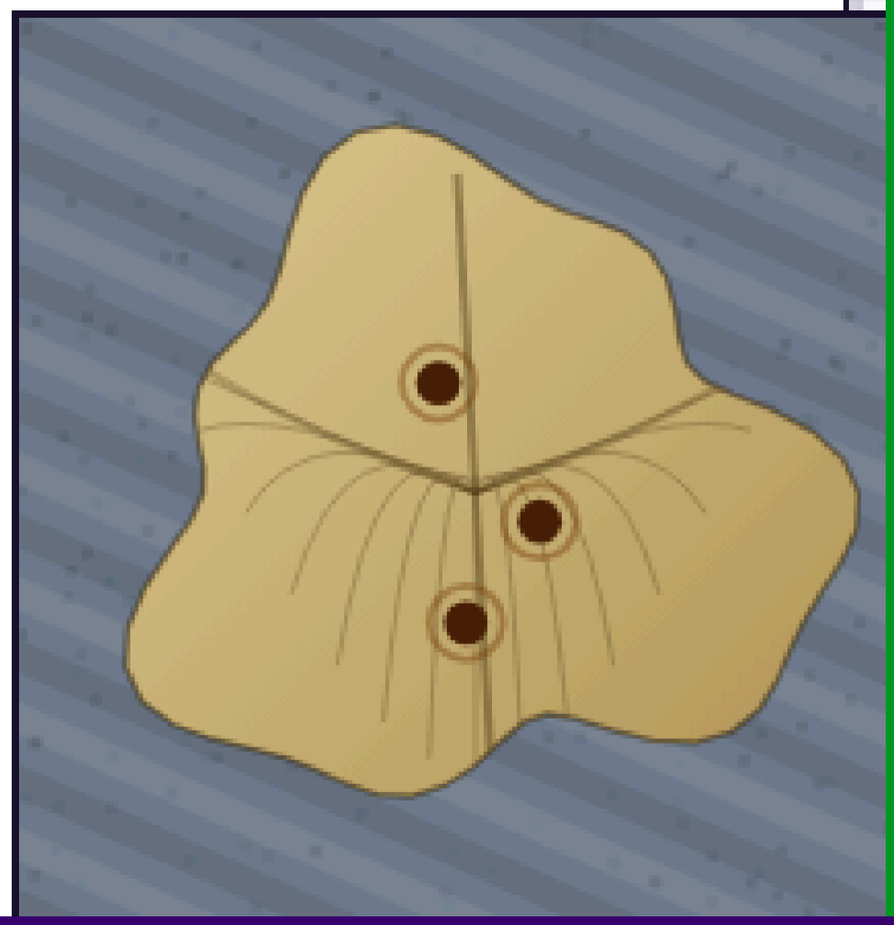
resolução 224 × 224

A FOTO CHEGA COM

50 mil

pixels para o modelo olhar.

Os três pontos escuros do objeto são



224 É O TAMANHO QUE OS PESOS ESPERAM

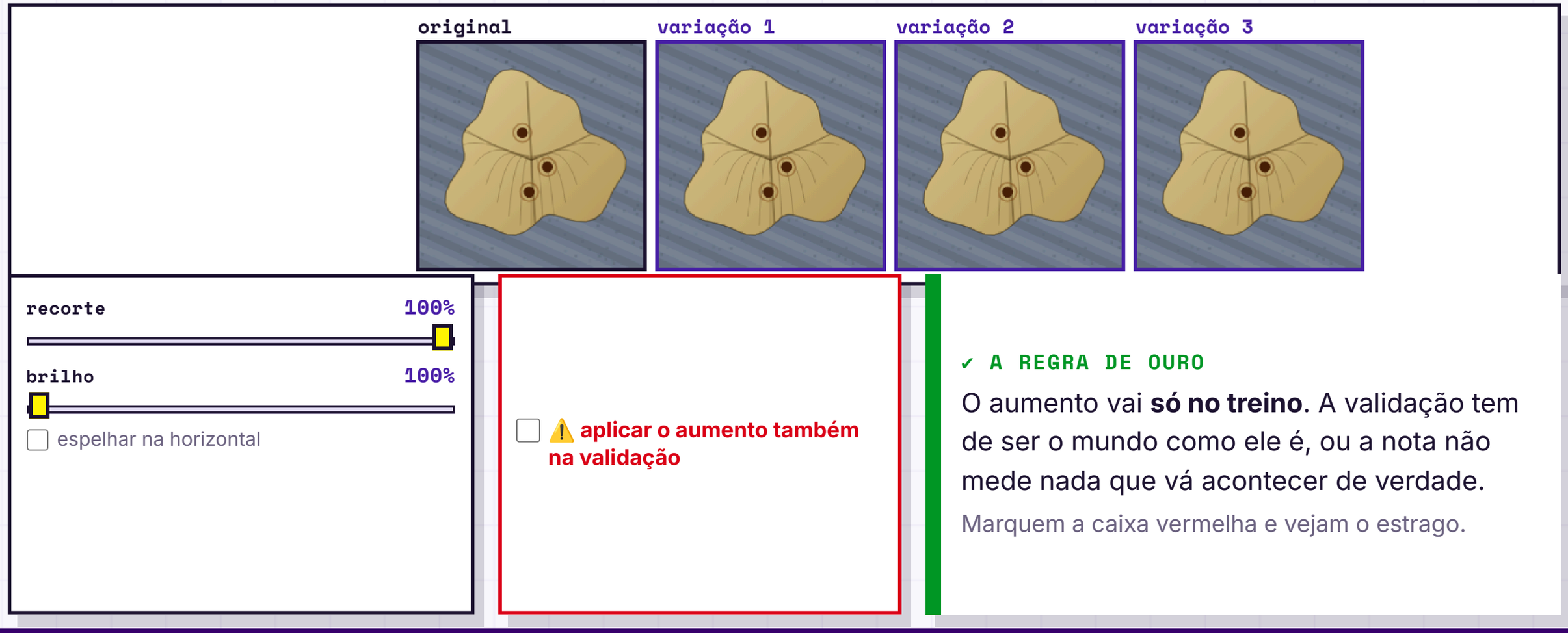
Os três modelos foram treinados nesse tamanho. Entregar menos joga fora informação; entregar mais custa memória sem ganhar quase nada.

LAB 13 · 224 PIXELS

A resolução em que os pesos pré-treinados foram feitos, e por isso a escolha padrão da trilha. Repare no número de pixels: são cerca de cem vezes mais que no primeiro estado. Resolução não é um botão de qualidade — é quanto trabalho vocês dão ao modelo e quanto tempo de GPU isso custa.

Uma foto vira quatro

Com poucas imagens, mostra-se a mesma foto várias vezes, cada vez um pouco diferente.

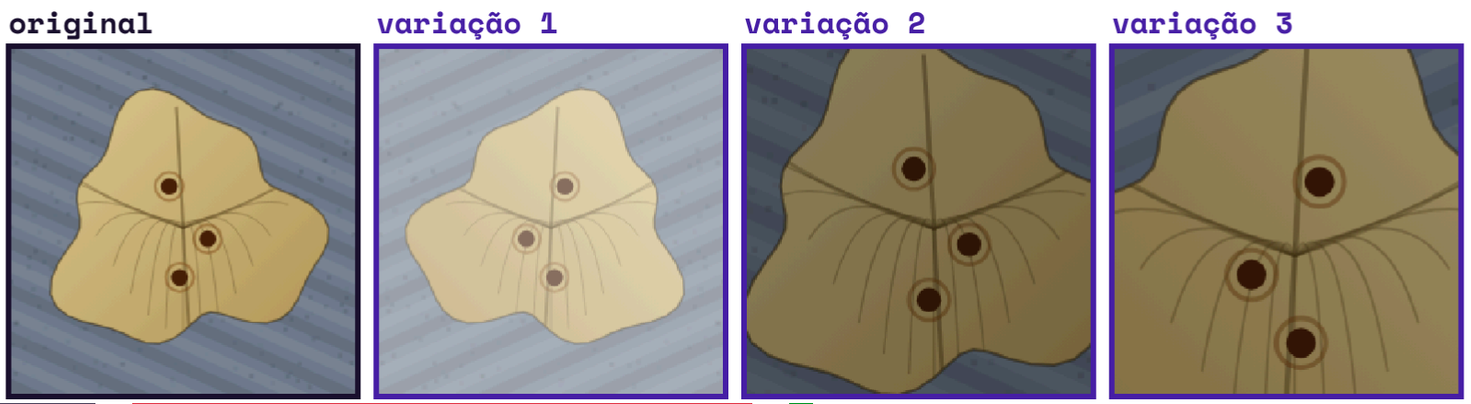


LAB 14 · SEM AUMENTO

O ponto de partida: a foto como ela é. Com poucos milhares de imagens e milhões de pesos, treinar assim leva o modelo a guardar as fotos em vez de aprender o que elas têm em comum. O aumento de dados existe para impedir isso.

Uma foto vira quatro

Com poucas imagens, mostra-se a mesma foto várias vezes, cada vez um pouco diferente.



recorte 55%

brilho 165%

☑ espelhar na horizontal

☐ ⚠️ **aplicar o aumento também na validação**

✔ **A REGRA DE OURO**

O aumento vai **só no treino**. A validação tem de ser o mundo como ele é, ou a nota não mede nada que vá acontecer de verdade.

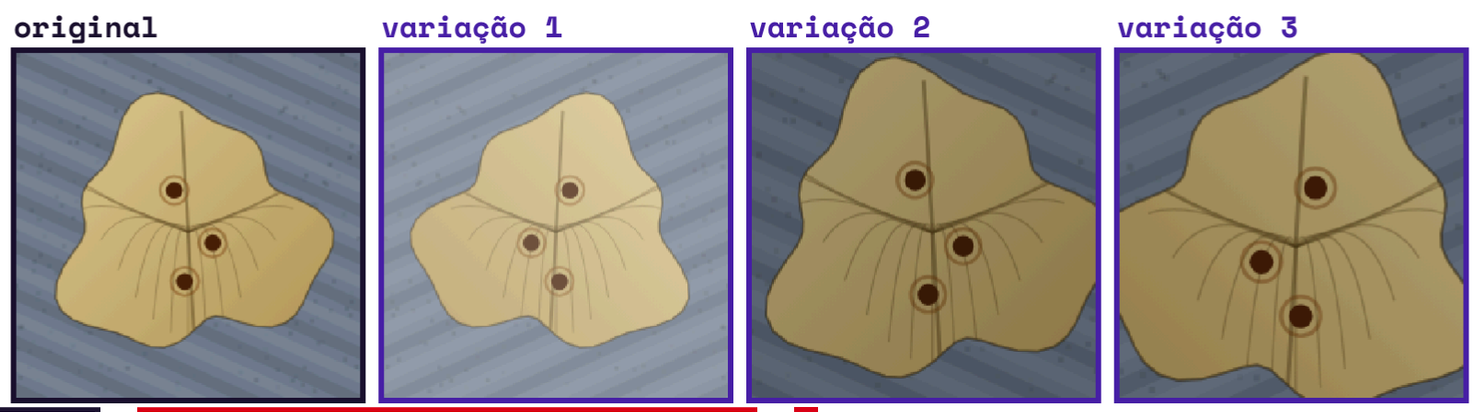
Marquem a caixa vermelha e vejam o estrago.

LAB 14 · AUMENTO FORTE

Recorte agressivo, espelhamento e brilho alterado. A cada época o modelo vê uma versão diferente da mesma foto, e por isso não consegue decorá-la. Cuidado com o exagero: distorção demais ensina o modelo a reconhecer coisas que não existem no campo.

Uma foto vira quatro

Com poucas imagens, mostra-se a mesma foto várias vezes, cada vez um pouco diferente.



recorte 70%

brilho 140%

espelhar na horizontal

⚠️ aplicar o aumento também na validação

× A REGRA DE OURO FOI QUEBRADA

O aumento também foi aplicado na **validação**. Agora a medição acontece sobre fotos distorcidas, que não existem no mundo. A nota que aparece deixa de dizer qualquer coisa — e **nenhuma mensagem de erro aparece**.

Pista: avaliem o mesmo modelo duas vezes. Se a nota **mudar**, o aumento vazou.

LAB 14 · O ERRO: AUMENTO NA VALIDAÇÃO

A regra de ouro desta trilha, e o erro mais comum nela: o aumento vai **só** no treino. Aplicado também na validação, ele distorce as fotos com que vocês medem, e a nota passa a depender do sorteio de cada rodada. O teste é simples — avaliem o mesmo modelo duas vezes. Se a nota mudar, o aumento vazou.

02

BLOCO 2

Configurar

- Três modelos que **já sabem enxergar**
- O treino acontece em **dois estágios**
- A camada de saída tem nome diferente em cada um
- Quatro botões, e só quatro

ABERTURA DO BLOCO 2

Daqui em diante, as quatro decisões do bloco de configuração: qual modelo, como dividir o treino em estágios, qual o nome da camada de saída, e onde ficam os controles de taxa, lote, épocas e aumento.

BLOCO 2 · ESCOLHA ★ TROQUE DE FICHA

Três modelos que já sabem enxergar

Nenhum deles começa do zero. Todos chegam com milhões de pesos já treinados.

EfficientNet-B0 | ResNet-50 | ViT-B/16

ECONÔMICA

Rede de convolução desenhada para dar o melhor resultado com o menor custo.

Escolham quando vocês querem rodar muitos experimentos no mesmo tempo. É a mais barata das três, de longe.

camada de saída	`modelo.classifier[1]`
entram nela	**1280**
taxa do estágio 2	**1e-4**
lote	**32**

A PEGADINHA DESTE

A camada de saída **não** se chama igual à da ResNet. Código copiado de uma para a outra não quebra: cria um atributo novo e treina a saída errada.

O QUE ESPERAR

Cinco vezes menor que a ResNet. No relógio a vantagem encolhe para perto do dobro, porque as contas dela aproveitam mal a placa. Resultado parecido com o da ResNet.

OS TRÊS MODELOS · EFFICIENTNET-B0

O mais leve dos três e o de entrada mais fácil: treina rápido, cabe com folga na GPU gratuita e costuma dar um resultado honesto sem ajuste fino. É a escolha sensata para a equipe que quer uma primeira volta completa antes de arriscar.

Três modelos que já sabem enxergar

Nenhum deles começa do zero. Todos chegam com milhões de pesos já treinados.

EfficientNet-B0 | **ResNet-50** | ViT-B/16

REFERÊNCIA

A rede de convolução mais conhecida da área, com atalhos internos que deixam o treino estável.

Escolham quando vocês querem previsibilidade e tempo sobrando para interpretar em vez de depurar.

camada de saída	`modelo.fc`
entram nela	**2048**
taxa do estágio 2	**1e-4**
lote	**32**

A PEGADINHA DESTE

Existem duas versões de pesos, e elas preparam a imagem de forma diferente. Usem sempre a preparação que vem junto com os pesos, nunca uma escrita à mão.

O QUE ESPERAR

Resultado sólido e sem surpresa. É a escolha segura da trilha, e o mapa de atenção funciona muito bem nela.

OS TRÊS MODELOS · RESNET-50

O intermediário, e o mais documentado dos três — qualquer dúvida que vocês tiverem, alguém já escreveu a respeito. Pesa mais que o anterior e costuma recompensar quem souber mexer nos dois estágios de treino.

Três modelos que já sabem enxergar

Nenhum deles começa do zero. Todos chegam com milhões de pesos já treinados.

EfficientNet-B0 | ResNet-50 | **ViT-B/16**

AVANÇADA

Corta a imagem em quadradinhos de 16 pixels e trata cada um como se fosse uma palavra.

Escolham quando vocês querem testar uma ideia diferente e têm orçamento de placa para isso.

camada de saída	`modelo.heads.head`
entram nela	**768**
taxa do estágio 2	**1e-5**
lote	**16**

A PEGADINHA DESTE

A entrada precisa ser **exatamente** 224 por 224. Qualquer outro tamanho quebra — diferente das duas de cima, que aceitam variação.

O QUE ESPERAR

Umas quatro vezes mais cara que a EfficientNet em tempo de placa, e pode **perder** para as CNNs. Se perder, isso vira um bom parágrafo de relatório.

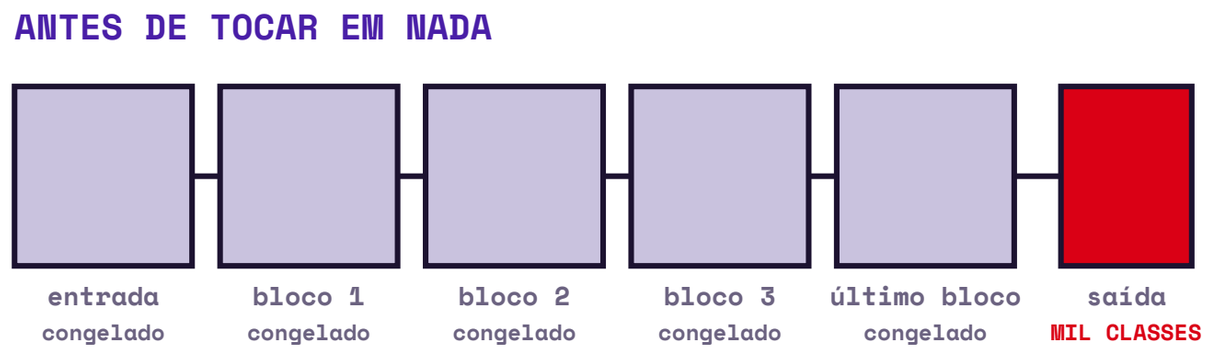
OS TRÊS MODELOS · ViT-B/16

O mais pesado e o mais exigente. Divide a foto em quadradinhos e trata cada um como um pedaço de informação, o que muda bastante o comportamento no treino: pede taxa de aprendizado menor que a dos outros dois e perdoa menos erro de configuração. Escolham com os olhos abertos.

O treino acontece em dois estágios

Descongelar o modelo inteiro com poucas imagens é receita de decoreba.

como ele chega | estágio 1 · sondagem | estágio 2 · ajuste fino



25 milhões de pesos — e a saída ainda é a do pré-treino

O QUE VEM NA CAIXA

O modelo chega sabendo enxergar, mas a última camada responde **mil classes** que não são as de vocês. Trocar essa camada é a primeira coisa a fazer — e é a lacuna mais traiçoeira do caderno.

DOIS ESTÁGIOS · ESTÁGIO 0

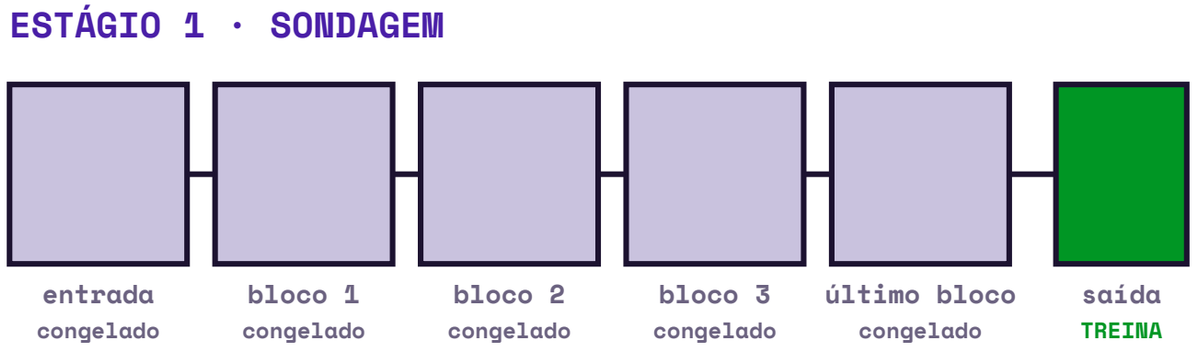
O modelo chega com o tronco já treinado em milhões de fotos genéricas: ele sabe enxergar borda, textura e forma. O que ele não sabe é o nome das classes de vocês. Este é o ponto de partida, antes de qualquer treino de vocês.

BLOCO 2 · ESTRATÉGIA ★ AVANCE OS ESTÁGIOS

O treino acontece em dois estágios

Descongelar o modelo inteiro com poucas imagens é receita de decoreba.

como ele chega | **estágio 1 · sondagem** | estágio 2 · ajuste fino



POR QUE COMEÇAR ASSIM

Congela tudo o que o modelo já sabia e treina **só a camada nova**. É rápido, não estraga nada, e já dá um resultado decente. Serve de piso para o estágio 2.

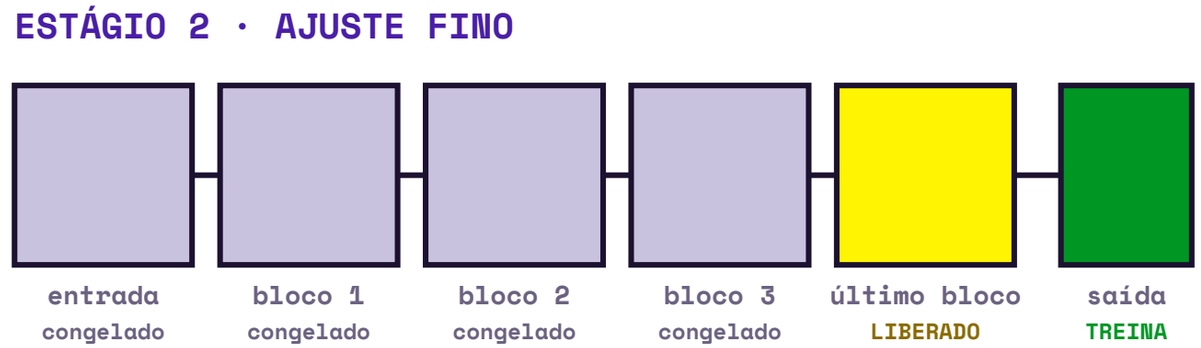
DOIS ESTÁGIOS · ESTÁGIO 1 · SÓ A CABEÇA

Primeiro estágio: o tronco fica congelado e só a camada de saída aprende. São poucos milhares de pesos se movendo, então é rápido e seguro — e serve para a cabeça nova sair do aleatório antes de o tronco ser liberado.

O treino acontece em dois estágios

Descongelar o modelo inteiro com poucas imagens é receita de decoreba.

como ele chega | estágio 1 · sondagem | estágio 2 · ajuste fino



POR QUE NÃO SOLTAR TUDO

Libera **só o último bloco**, com passo dez vezes menor. Descongelar o modelo inteiro com poucas imagens é receita de decoreba: ele apaga o que sabia e decora o que você mostrou.

DOIS ESTÁGIOS · ESTÁGIO 2 · O ÚLTIMO BLOCO

Segundo estágio: libera-se o último bloco do tronco, que é onde moram as características mais específicas. Agora são milhões de pesos se movendo, e por isso o passo tem de ser menor. Liberar tudo de uma vez, com passo grande, apaga o que o modelo já sabia — é o erro que a curva de "passo grande demais" mostra mais adiante.

BLOCO 2 · LACUNA CURTA E TRAIÇOEIRA ★ ESCOLHA A LINHA

A saída tem nome diferente em cada modelo

Vocês escolheram a **ResNet-50**. Qual destas linhas troca a camada de saída dela?

ResNet-50 | EfficientNet-B0 | ViT-B/16

```
modelo.fc = nn.Linear(2048, n_classes)
```

```
modelo.classifier[1] = nn.Linear(1280, n_classes)
```

```
modelo.heads.head = nn.Linear(768, n_classes)
```

✔ É ESSA

Na ResNet a camada de saída se chama `fc`, e entram 2048 números nela.

A CAMADA DE SAÍDA · RESNET-50

Cada arquitetura guarda a camada de saída com um nome diferente. Na ResNet ela se chama `fc`. Trocar essa camada é o que transforma um classificador de mil categorias genéricas no classificador das classes de vocês.

A saída tem nome diferente em cada modelo

Vocês escolheram a **EfficientNet-B0**. Qual destas linhas troca a camada de saída dela?

ResNet-50 | **EfficientNet-B0** | ViT-B/16

```
modelo.fc = nn.Linear(2048, n_classes)
```

```
modelo.classifier[1] = nn.Linear(1280, n_classes)
```

```
modelo.heads.head = nn.Linear(768, n_classes)
```

× ESSE É O NOME DA RESNET-50

Na ResNet a camada de saída se chama `fc`, e entram 2048 números nela. Usada na EfficientNet-B0, essa linha **não quebra** — e é por isso que ela é perigosa.

A CAMADA DE SAÍDA · A ARMADILHA

Aqui está a razão de este laboratório existir. A equipe escolheu a EfficientNet e copiou a linha da ResNet. Em Python, atribuir a um nome que não existe **não dá erro**: cria um atributo novo que o modelo ignora. O treino roda inteiro, a saída de mil classes do pré-treino continua lá, e o placar sai péssimo sem nenhuma mensagem de erro para explicar por quê.

BLOCO 2 · LACUNA CURTA E TRAIÇOEIRA ★ ESCOLHA A LINHA

A saída tem nome diferente em cada modelo

Vocês escolheram a **ViT-B/16**. Qual destas linhas troca a camada de saída dela?

ResNet-50 | EfficientNet-B0 | ViT-B/16

```
modelo.fc = nn.Linear(2048, n_classes)
```

```
modelo.classifier[1] = nn.Linear(1280, n_classes)
```

```
modelo.heads.head = nn.Linear(768, n_classes)
```

✔ É ESSA

No ViT a saída fica **dentro** de heads, e se chama head. Entram 768.

A CAMADA DE SAÍDA · VIT-B/16

E no ViT o nome é outro ainda. Três modelos, três nomes, nenhum aviso quando se erra. Confiram esta linha olhando a documentação do modelo que vocês escolheram, não a de outro.

Quatro botões, e só quatro

Não existem mil coisas para ajustar. Existem estas — e a curva responde na hora.

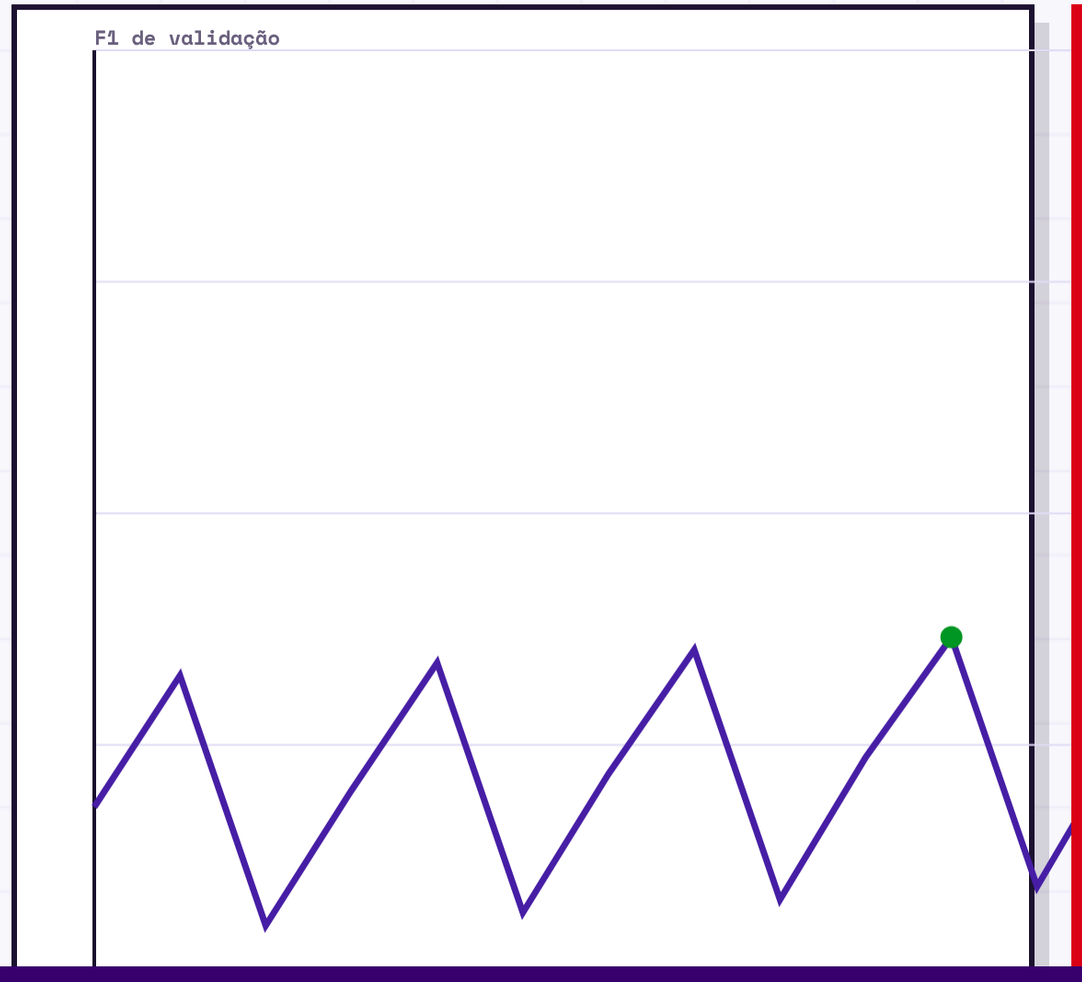
taxa 1e-2

lote 32

épocas 12

aumento leve

MELHOR F1 NA PLACA



PASSO GRANDE DEMAIS

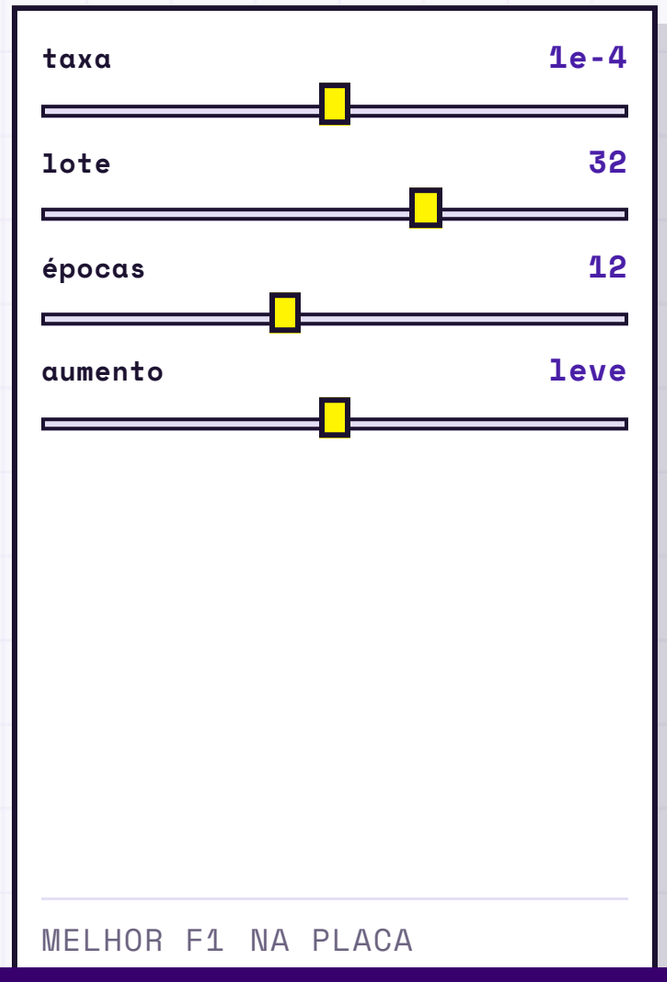
A curva serrilha e não sobe. Com pesos pré-treinados, um passo desse tamanho **apaga o que o modelo já sabia** nas primeiras épocas.

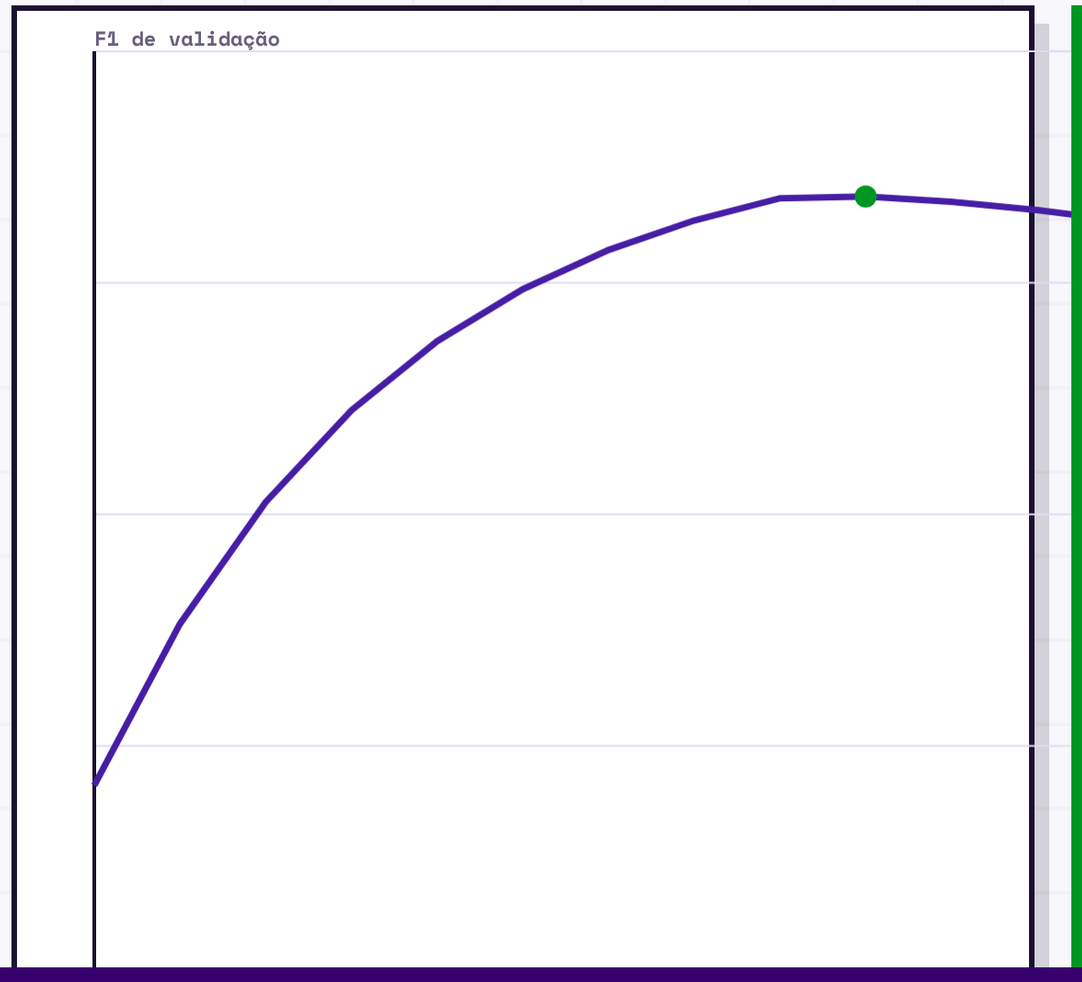
HIPERPARÂMETROS · O PASSO GRANDE DEMAIS

Taxa de aprendizado no extremo de cima. Num modelo que já vem treinado, um passo desse tamanho não ajusta: atropela. Os pesos que levaram milhões de fotos para ficar bons são destruídos nas primeiras épocas, e a perda serrilha sem descer.

Quatro botões, e só quatro

Não existem mil coisas para ajustar. Existem estas — e a curva responde na hora.





FAIXA DE TRABALHO

São só estes quatro botões. Mexam **um por vez** e anotem: é isso que vira a tabela de experimentos no relatório.

HIPERPARÂMETROS · O PASSO DE TRABALHO

O valor padrão da trilha para EfficientNet e ResNet. Com o tronco pré-treinado, o ajuste tem de ser pequeno — a ordem de grandeza aqui é a diferença entre afinar e apagar. Para o ViT, menor ainda.

Quatro botões, e só quatro

Não existem mil coisas para ajustar. Existem estas — e a curva responde na hora.

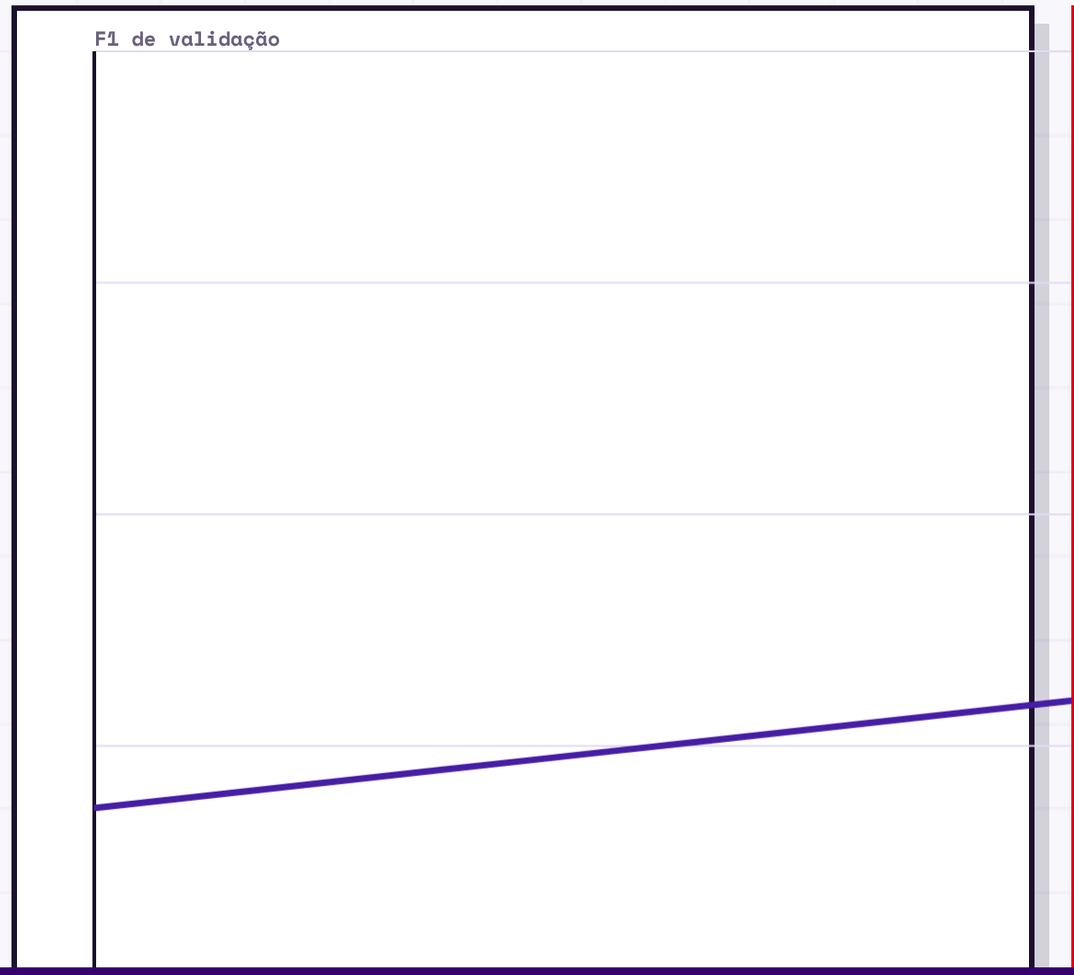
taxa 1e-6

lote 32

épocas 12

aumento leve

MELHOR F1 NA PLACA



PASSO PEQUENO DEMAIS

Quase nada se move. Vocês vão gastar o orçamento de placa para ver uma reta.

HIPERPARÂMETROS · O PASSO PEQUENO DEMAIS

O extremo de baixo. Nada quebra, e é justamente por isso que engana: o modelo aprende tão devagar que, nas épocas que cabem no tempo de vocês, ele mal sai do lugar. Uma curva que desce muito pouco e muito devagar costuma ser isto, não falta de capacidade.

03

BLOCO 3

Modelar

- As **mesmas três células**: A prepara, B treina, C avalia
- A forma nunca muda de trilha para trilha
- **É aqui que ficam as lacunas do desafio.** Todas elas.

ABERTURA DO BLOCO 3

Este é o bloco que tem código para escrever. As duas páginas seguintes mostram as atividades de completar linha resolvidas, para vocês conferirem o raciocínio depois de terem tentado — não antes.

BLOCO 3 · ATIVIDADE 3.1 ★ COMPLETEM O CÓDIGO

Construa o modelo que vocês escolheram

Escolham o modelo, cliquem num espaço amarelo e respondam no painel da direita.

ResNet-50 | EfficientNet-B0 | ViT-B/16

8 de 8 certas · 0 em branco

```
# ═══ CÉLULA A · preparo ═══
# a preparação vem junto com os pesos, nunca escrita à mão
preparo_val = PESOS.transforms()
# no treino entra o aumento: a mesma foto, sempre diferente
preparo_tr = Compose([RandomResizedCrop(224), RandomHorizontalFlip() ,
                      ToTensor(), Normalize(med, dp)])
# o aumento vai SÓ no treino; a validação é o mundo como ele é
validacao = Dataset(pasta_val, transform= preparo_val )

# ═══ CÉLULA B · treino ═══
modelo = resnet50(weights=PESOS)
# cada modelo guarda a saída com um nome diferente
modelo. fc  = nn.Linear(2048, n_classes)
otimizador =  AdamW (modelo.parameters(), lr= 1e-4 )
# ═══ CÉLULA C · avaliação ═══
modelo. eval ()
print( f1_score (y_va, previsoes, average= "macro" ))
```

LACUNA 8 DE 8 · JÁ RESPONDIDA

responder de novo

✔ **CÉLULA COMPLETA**

As 8 lacunas estão certas. Assim esta célula roda sem erro — e, mais importante, sem o erro silencioso que não dá mensagem nenhuma.

COMPLETE O CÓDIGO · RESNET-50 (RESOLVIDO)

A célula de preparo e a de treino, com todas as lacunas preenchidas. Reparem na ordem das decisões: a preparação da validação vem dos próprios pesos, o aumento entra só no treino, e a camada de saída recebe o nome da arquitetura escolhida. Cada uma dessas linhas é uma lacuna no caderno do desafio.

BLOCO 3 · ATIVIDADE 3.1 ★ COMPLETEM O CÓDIGO

Construa o modelo que vocês escolheram

Escolham o modelo, cliquem num espaço amarelo e respondam no painel da direita.

ResNet-50 | **EfficientNet-B0** | ViT-B/16

6 de 6 certas · 0 em branco

```
# ═══ CÉLULA A · preparo ═══ (igual ao da ResNet)
preparo_val = PESOS.transforms()
preparo_tr  = Compose([RandomResizedCrop(224), RandomHorizontalFlip(),
                       ColorJitter(brightness=0.2), ToTensor(), Normalize(med, dp)

# ═══ CÉLULA B · treino ═══
modelo = efficientnet_b0(weights=PESOS)
# aqui a saída NÃO se chama fc
modelo.[classifier[1]] = nn.Linear(1280, n_classes)
otimizador = [AdamW](modelo.parameters(), lr=1e-4)
# modelo pequeno: cabe um lote maior na memória
lote = [32]

# ═══ CÉLULA C · avaliação ═══
modelo.[eval]()
print([f1_score](y_va, previsoes, average=[macro]))
```

LACUNA 6 DE 6 · JÁ RESPONDIDA

responder de novo

✔ **CÉLULA COMPLETA**

As 6 lacunas estão certas. Assim esta célula roda sem erro — e, mais importante, sem o erro silencioso que não dá mensagem nenhuma.

COMPLETE O CÓDIGO · EFFICIENTNET-B0 (RESOLVIDO)

A mesma atividade para a EfficientNet. Compare com a página anterior: o esqueleto é idêntico e três coisas mudam — o construtor do modelo, o nome da camada de saída e o número de entradas dela. É exatamente esse conjunto que a equipe precisa saber localizar na documentação do modelo que escolheu.

BLOCO 3 · ATIVIDADE 3.1 ★ COMPLETEM O CÓDIGO

Construa o modelo que vocês escolheram

Escolham o modelo, cliquem num espaço amarelo e respondam no painel da direita.

ResNet-50 | EfficientNet-B0 | **ViT-B/16**

6 de 6 certas · 0 em branco

```
# ═══ CÉLULA A · preparo ═══
# o ViT corta a imagem em quadradinhos de 16 pixels
# por isso a entrada precisa ser EXATAMENTE 224 por 224
preparo_val = PESOS.transforms()

# ═══ CÉLULA B · treino ═══
modelo = vit_b_16(weights=PESOS)
modelo.[heads.head] = nn.Linear(768, n_classes)
# o mais sensível dos três: passo dez vezes menor que o das CNNs
otimizador = [AdamW](modelo.parameters(), lr=[1e-5])
# modelo grande: o lote precisa ser menor para caber na memória
lote = 16

# ═══ CÉLULA C · avaliação ═══
modelo.[eval]()
print([f1_score](y_va, previsoes, average=["macro"]))
```

LACUNA 6 DE 6 · JÁ RESPONDIDA

responder de novo

✔ **CÉLULA COMPLETA**

As 6 lacunas estão certas. Assim esta célula roda sem erro — e, mais importante, sem o erro silencioso que não dá mensagem nenhuma.

COMPLETE O CÓDIGO · VIT-B/16 (RESOLVIDO)

E a terceira variação. Três modelos, o mesmo caderno, e a diferença concentrada em meia dúzia de nomes. Quem entendeu o padrão das três páginas consegue trocar de modelo no meio do desafio sem reescrever a célula inteira.

BLOCO 3 · ATIVIDADE 3.2 ★ ACHEM A LINHA

Este código roda e entrega menos do que deveria

Sem mensagem nenhuma. Cliquem na linha culpada.

```
treino = Dataset(pasta_treino, transform=preparo_treino)

validacao = Dataset(pasta_val, transform=preparo_treino)

modelo = resnet50(weights=PESOS)

modelo.fc = nn.Linear(2048, n_classes)

for epoca in range(15): treinar_uma_epoca(modelo, treino)

print(avaliar(modelo, validacao))
```

✔ É ESSA A LINHA

Aqui está. A validação recebeu o **preparo do treino**, com recorte ao acaso e variação de cor. A nota que aparece deixa de dizer qualquer coisa sobre fotos de verdade — e ninguém recebe mensagem de erro.

E NO CADERNO DE VOCÊS

A célula de verificação passa a mesma foto duas vezes pela fila da validação. Se sair diferente, ela **para o caderno** e diz o que aconteceu — em vez de deixar a nota mentir.

CAÇA AO ERRO · A LINHA QUE ESTÁ ERRADA

Sete linhas plausíveis, uma defeituosa. A errada é a da validação, que recebeu o preparo do treino — com o aumento de dados dentro. É o mesmo erro do Laboratório 14, agora escrito em código: o aumento distorcendo justamente as fotos com que se mede. Nenhuma das sete dá mensagem de erro.

04

BLOCO 4

Avaliar

- O que as **duas curvas** dizem numa trilha de imagem
- O modelo olhou para o objeto, ou para o fundo?
- A matriz de confusão é o mapa do próximo teste

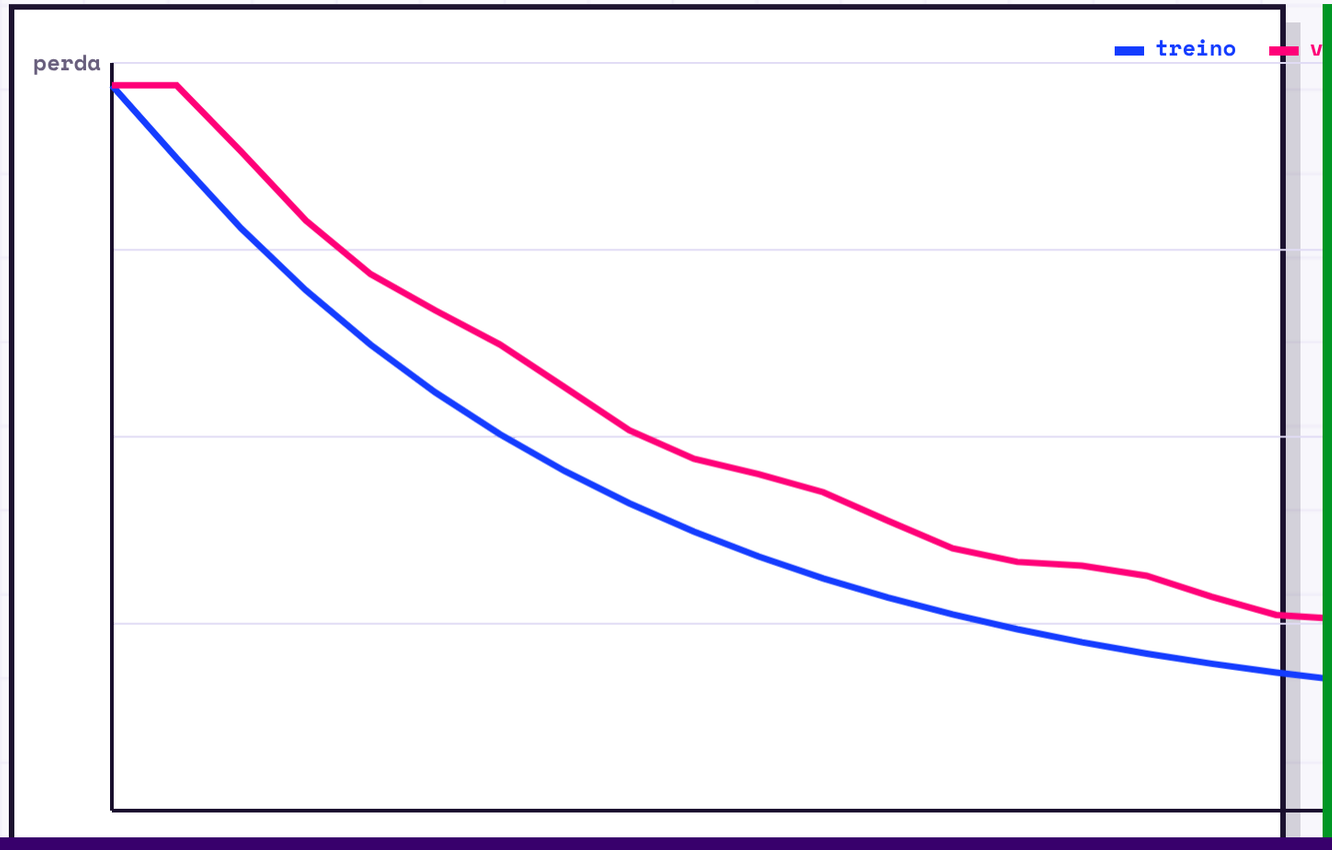
ABERTURA DO BLOCO 4

O último bloco, e o que mais pesa no relatório. Três leituras: o que as curvas dizem, onde o modelo olhou, e com quem cada classe é confundida.

O que as curvas dizem numa trilha de imagem

A leitura é a mesma do capítulo anterior. As causas é que mudam.

Como deve ser | Decoreba | Validação fácil demais | Passo grande demais



TREINO E VALIDAÇÃO CAINDO JUNTAS

As duas descem e a distância entre elas fica pequena. É o que vocês querem ver. A validação assenta num patamar — e é a melhor época dela que vai para a entrega, não a última.

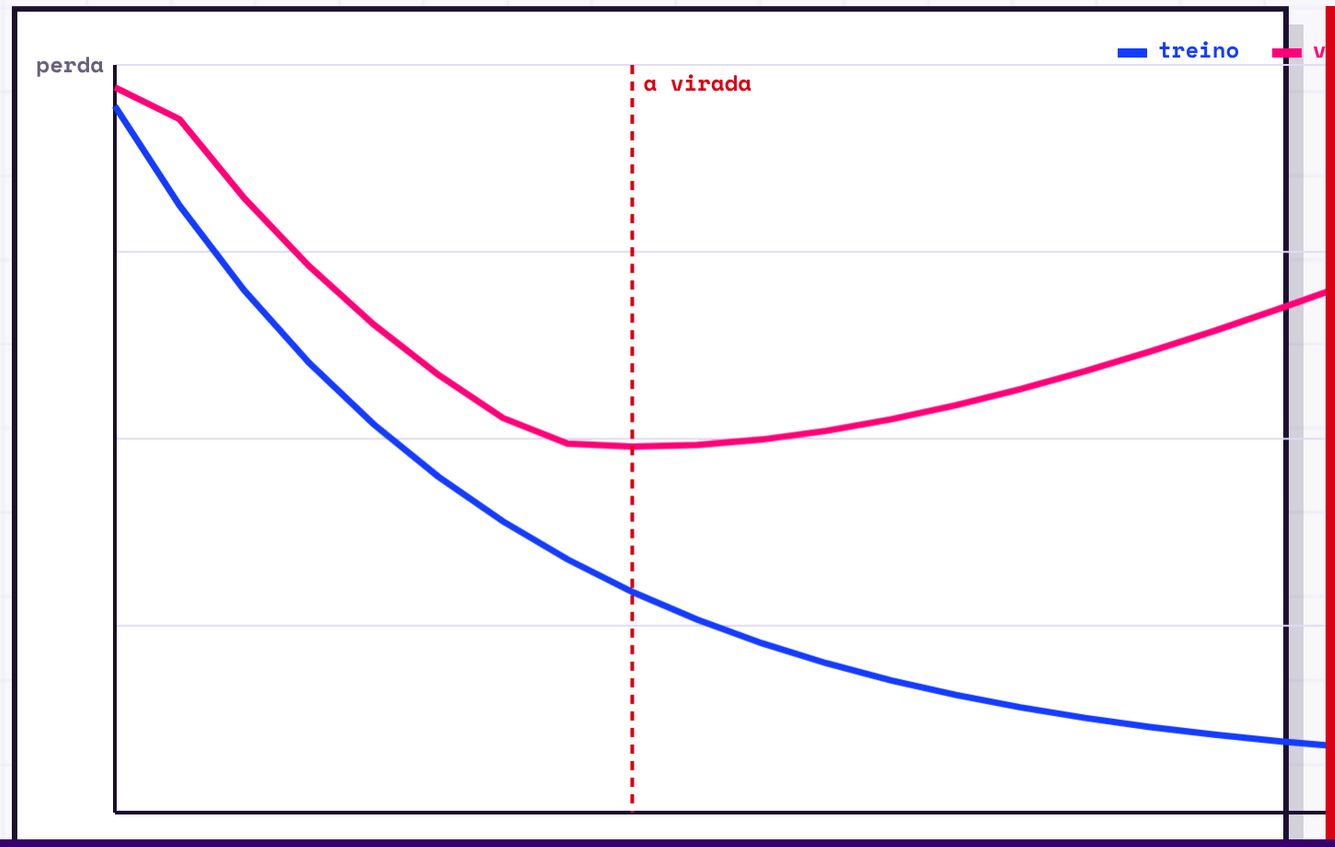
AS CURVAS · COMO DEVE SER

As duas descem e a distância entre elas fica pequena. A validação assenta num patamar — e é a melhor época dela que vai para a entrega, não a última. Guardem esta página como referência: as três seguintes são o que dá errado.

O que as curvas dizem numa trilha de imagem

A leitura é a mesma do capítulo anterior. As causas é que mudam.

Como deve ser | **Decoreba** | Validação fácil demais | Passo grande demais



TREINO DESCE, VALIDAÇÃO SOBE

Depois da virada, o modelo está guardando as fotos do treino em vez de aprender o que elas têm em comum. Com poucas imagens e milhões de pesos, **isso é a regra, não a exceção** — por isso a trilha tem aumento de dados, dois estágios e parada antecipada.

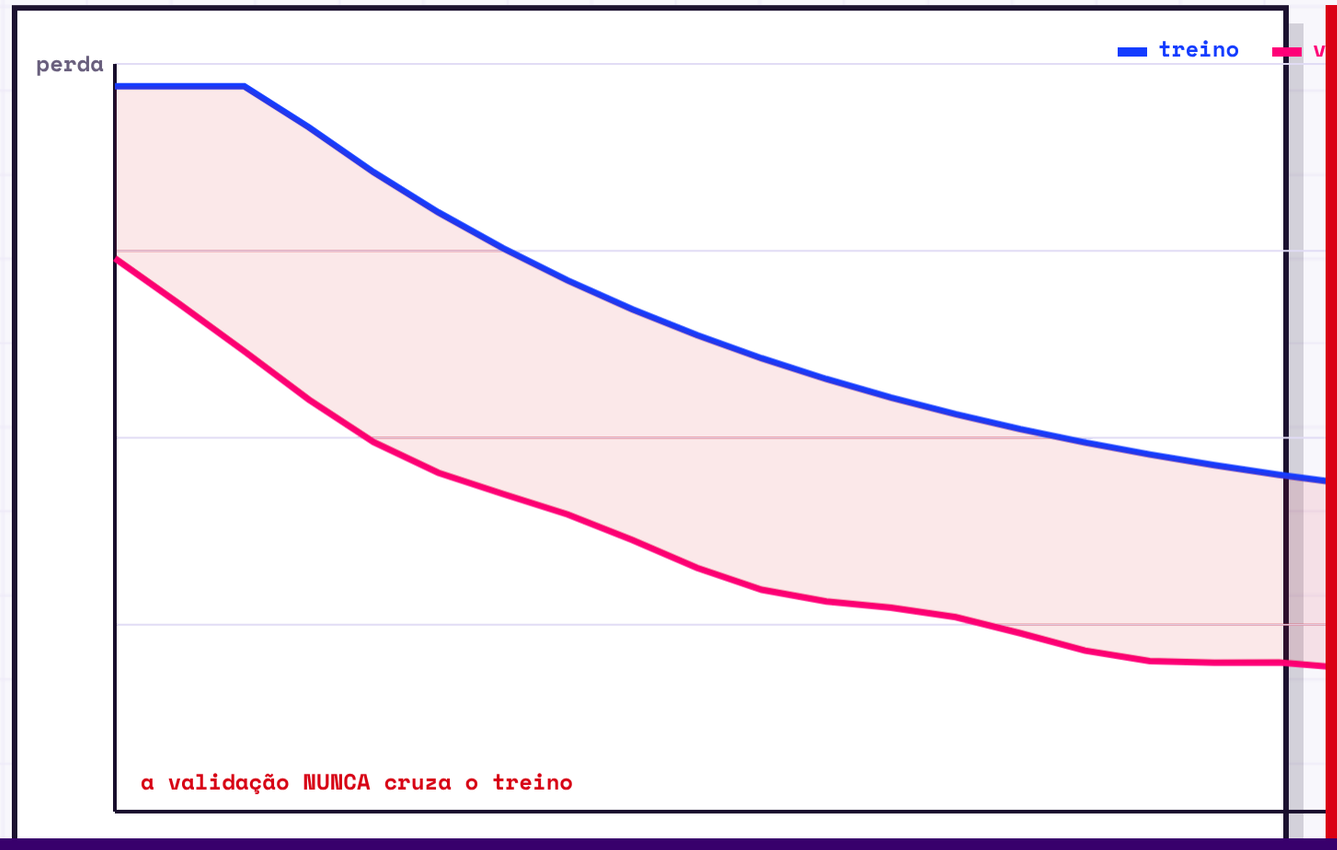
AS CURVAS · DECOREBA

O treino desce, a validação vira para cima. Depois da virada, o modelo está guardando as fotos do treino em vez de aprender o que elas têm em comum. Com poucas imagens e milhões de pesos isso é a regra, não a exceção — por isso a trilha tem aumento de dados, dois estágios e parada antecipada.

O que as curvas dizem numa trilha de imagem

A leitura é a mesma do capítulo anterior. As causas é que mudam.

Como deve ser | Decoreba | **Validação fácil demais** | Passo grande demais



A VALIDAÇÃO MUITO ABAIXO DO TREINO O TEMPO TODO

Um pouco abaixo é **normal**: o treino é feito com fotos distorcidas pelo aumento de dados e com dropout ligado, e a validação não. Uma distância grande assim é suspeita: quase sempre a validação está fácil demais, com **fotos repetidas ou quase iguais** entre treino e validação. Procurem duplicatas entre as pastas antes de comemorar.

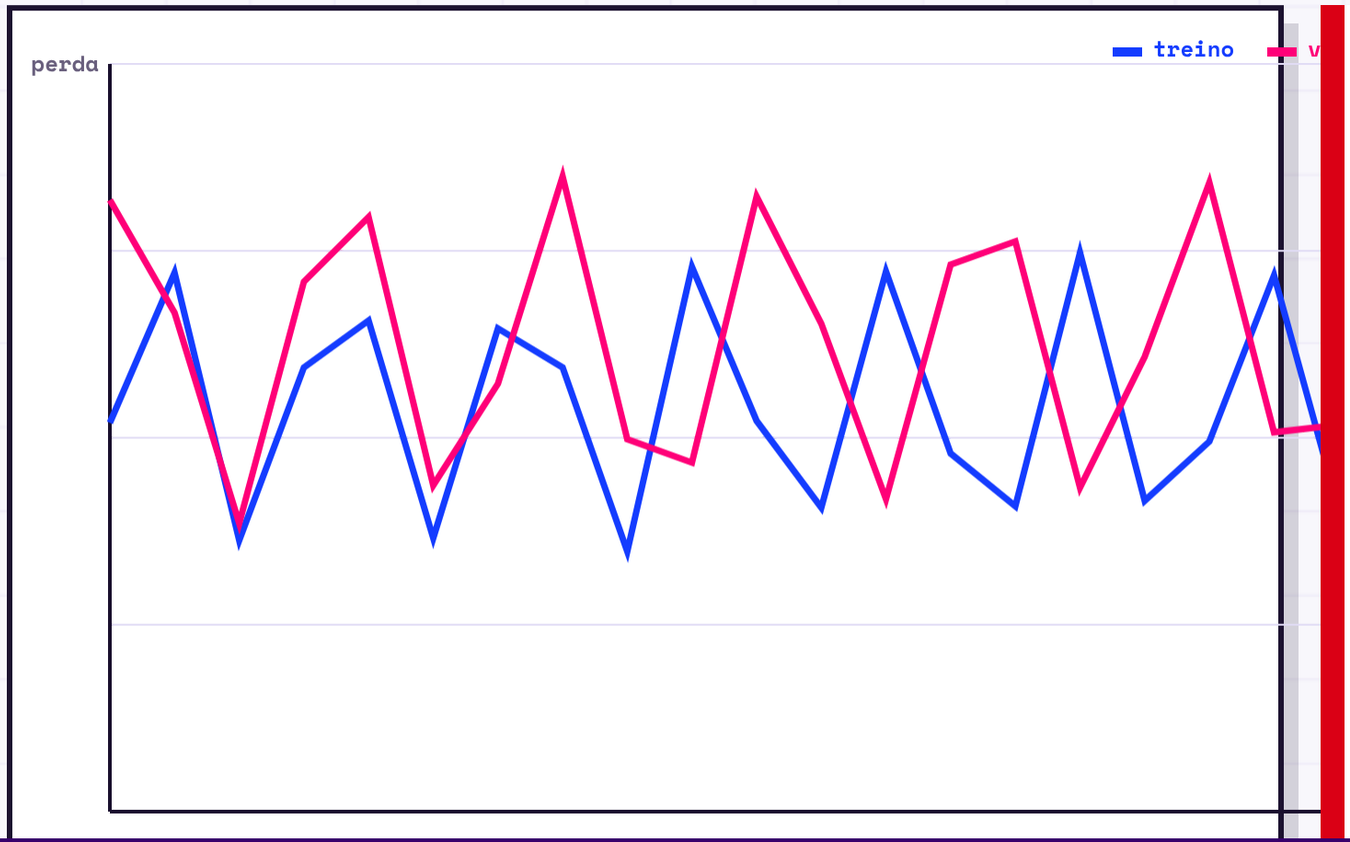
AS CURVAS · VALIDAÇÃO FÁCIL DEMAIS

A validação muito abaixo do treino o tempo todo. Um pouco abaixo é normal: o treino roda com fotos distorcidas pelo aumento e com dropout ligado, e a validação não. Uma distância grande é suspeita — quase sempre há fotos repetidas ou quase iguais entre as duas pastas. Procurem duplicatas antes de comemorar.

O que as curvas dizem numa trilha de imagem

A leitura é a mesma do capítulo anterior. As causas é que mudam.

Como deve ser | Decoreba | Validação fácil demais | **Passo grande demais**



AS DUAS SERRILHAM E NÃO DESCEM

A perda pula de uma época para a outra e não assenta. Num modelo pré-treinado isso costuma significar que o passo apagou o que ele já sabia. Diminuam a taxa antes de mexer em qualquer outra coisa.

AS CURVAS · PASSO GRANDE DEMAIS

As duas serrilham e não descem. Num modelo pré-treinado isso costuma significar que o passo apagou o que ele já sabia — é a mesma história da página dos hiperparâmetros, vista agora pelo gráfico. Diminuam a taxa antes de mexer em qualquer outra coisa.

O modelo olhou para o objeto ou para o fundo?

Esta pergunta não aparece em nenhuma métrica. Dois modelos, a mesma acurácia.

Modelo A · 91% | **Modelo B · 91%**

MODELO A · 91% DE ACERTO

O calor está **sobre o objeto**, e principalmente sobre os detalhes que decidem a classe. Este modelo vai continuar acertando em outro lugar, com outra luz e outra câmera.

Os dois acertam 91%. Um vai continuar acertando em outro lugar, com outra luz. O

LAB 15 · O MODELO OLHOU A PLANTA

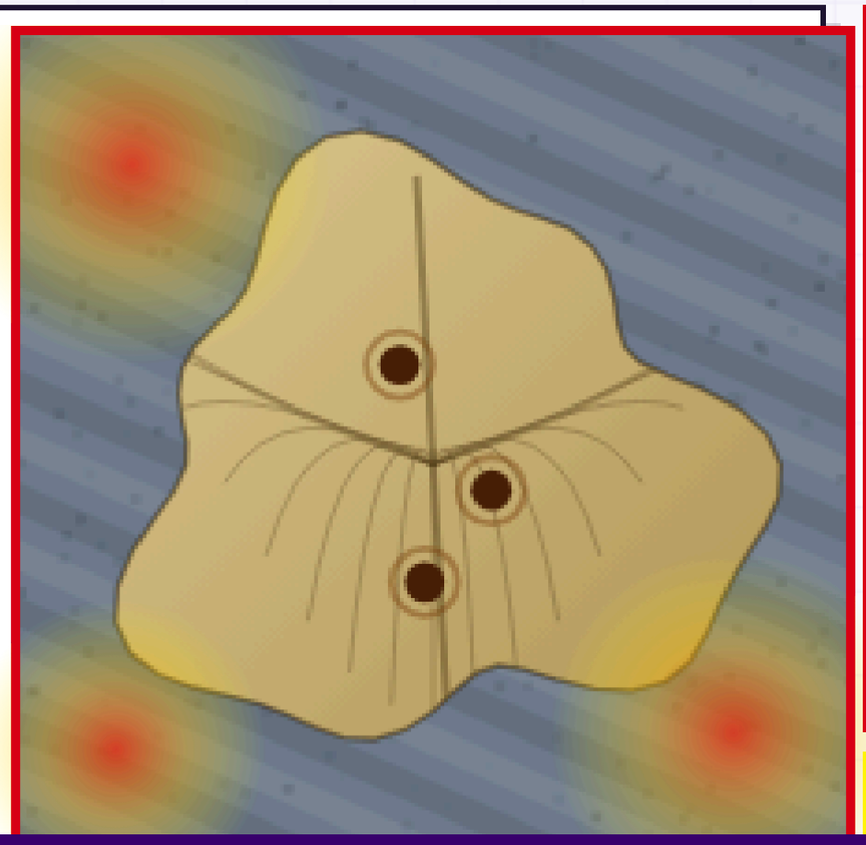
O mapa de calor mostra onde o modelo prestou atenção para decidir. Aqui ele olhou o objeto — é o que vocês querem ver, e é o que permite dizer à cliente que o acerto tem motivo.

BLOCO 4 · LABORATÓRIO 15 ★ ALTERNE OS DOIS CASOS

O modelo olhou para o objeto ou para o fundo?

Esta pergunta não aparece em nenhuma métrica. Dois modelos, a mesma acurácia.

Modelo A · 91% | Modelo B · 91%



MODELO B · 91% DE ACERTO

O mesmo número — e o calor está **nos cantos**, no fundo. Ele aprendeu uma pista do cenário, não o objeto. Troque o fundo e ele desaba.

Nenhuma métrica separa os dois. Só olhar separa.

Os dois acertam 91%. Um vai continuar acertando em outro lugar, com outra luz. O

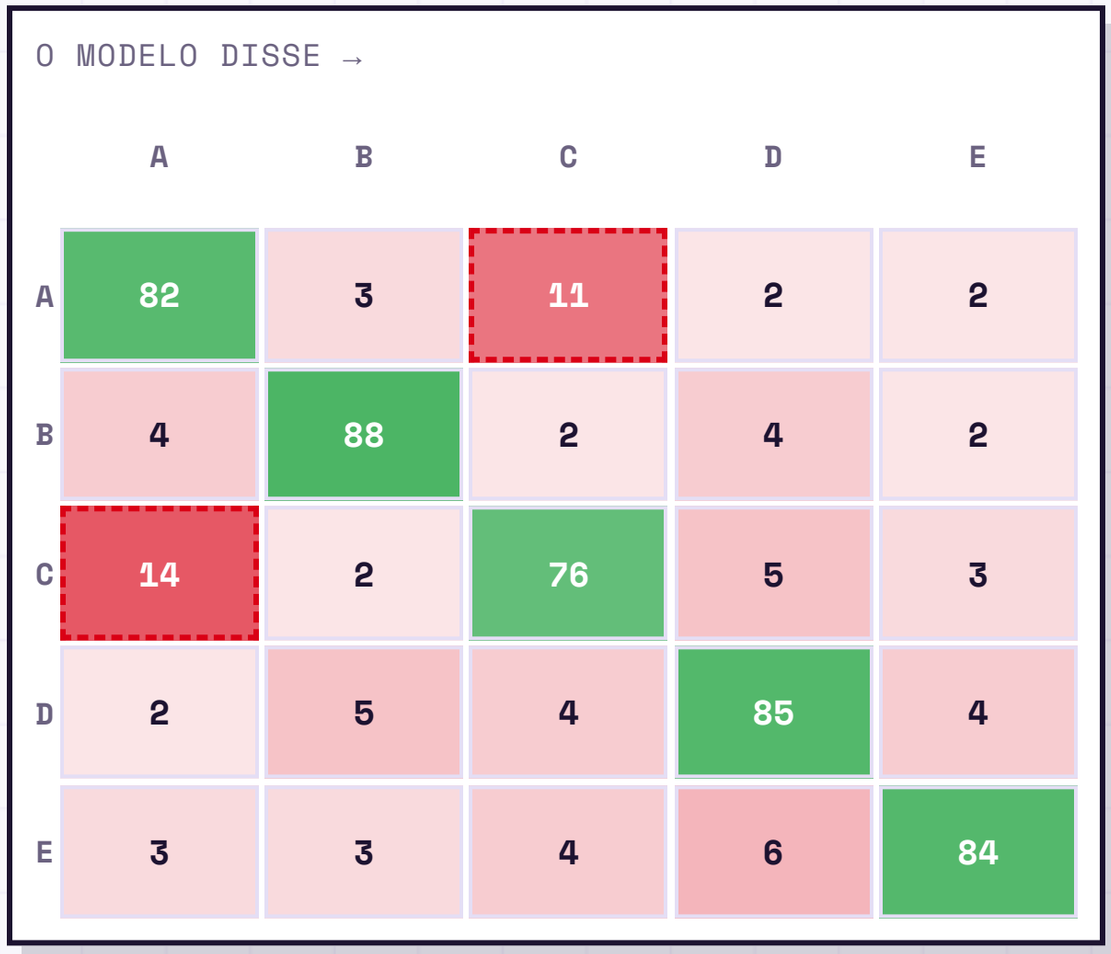
LAB 15 · O MODELO OLHOU O FUNDO

O mesmo acerto, pelo motivo errado. O modelo decidiu olhando os cantos da foto — o fundo, não a planta. Num conjunto onde cada classe foi fotografada num lugar diferente, isso acerta no teste e falha no campo, onde o fundo é outro. É por isso que esta página existe, e é por isso que a nota do desafio não é só o placar.

BLOCO 4 · MATRIZ ★ CLIQUEM FORA DA DIAGONAL

O modelo não erra ao acaso

Cinco classes, cinco linhas. Exemplo ilustrativo — a matriz de vocês sai do próprio treino.



A DIAGONAL É O QUE DEU CERTO

82 das 100 fotos da **classe A** foram para o lugar certo. A diagonal não ensina nada de novo — o que interessa está **fora** dela.

A MATRIZ · A DIAGONAL

A matriz de confusão lida de uma vez: cada linha é a classe verdadeira, cada coluna o palpite do modelo. A diagonal é o que deu certo e não ensina nada de novo. O que interessa está fora dela.

O modelo não erra ao acaso

Cinco classes, cinco linhas. Exemplo ilustrativo — a matriz de vocês sai do próprio treino.

O MODELO DISSE →

	A	B	C	D	E
A	82	3	11	2	2
B	4	88	2	4	2
C	14	2	76	5	3
D	2	5	4	85	4
E	3	3	4	6	84

CLASSE C → CLASSE A

O mesmo par, no sentido contrário. Quando a confusão é **simétrica**, o problema é das classes serem parecidas. Quando só um lado erra, costuma ser desequilíbrio no número de fotos.

A MATRIZ · O PAR QUE TROCA

O maior erro da matriz, e é recíproco: parte da classe C vai para a A e parte da A volta para a C. Um par assim quase nunca é defeito do modelo — é um par que se parece de verdade, e a pessoa que entende do assunto reconhece o erro na hora. É este tipo de leitura que a cliente da narrativa está pedindo, e não a média geral.

O QUE LEVAR DAQUI

Cinco frases e um endereço

O modelo chega sabendo enxergar — vocês aproveitam.

O aumento vai só no treino. Na validação, nunca.

Cada modelo tem um nome para a saída, e errar não dá erro.

Quem vai para a entrega é a melhor época, não a última.

Número alto olhando para o fundo não é resultado.

Agora abram o Capítulo 3 do material de imersão e refaçam os laboratórios com calma.

olimpiadadeia.ceia.digital

olimpiada@ceia.ufg.br

FECHAMENTO

O fim do aulão e o endereço do material vivo. Tudo o que está neste PDF pode ser mexido lá: os controles, as lacunas, os dois estágios, as curvas. Quem quiser realmente entender, abre o material e refaz com calma — este documento serve para lembrar o que foi visto, não para substituir a mexida.