

Guia Prático: Lab 07 — Visão Computacional com Transfer Learning

Inteligência Artificial II (IA II) — CEFET-MG

Prof. Aléssio Miranda Júnior

2026-01-01

Índice

1	Objetivo da Atividade	1
2	Dinâmica de Sala de Aula (60 minutos)	2
3	Passo a Passo do Laboratório	2
3.1	Etapa 1: A Dor de Aprender do Zero (Cenário 0)	2
3.2	Etapa 2: A Mágica do Transfer Learning (Cenário A)	2
3.3	Etapa 3: A Armadilha do Descongelamento (Cenário B)	3
3.4	Etapa 4: A Explosão Dimensional (Cenário C)	3
4	O Painel de Rastreamento Acumulado	3
5	Etapa 5: Conclusão de Engenharia	4
6	Validação e Submissão (GitLab CI/CD)	4

1 Objetivo da Atividade

Neste laboratório prático, você aplicará **Transfer Learning** utilizando a rede profunda MobileNetV2 (pré-treinada pelo Google no dataset ImageNet com 1,4 milhão de imagens).

O código inicial que você recebeu no arquivo `src/transfer_learning_starter.py` contém uma CNN básica (treinada do zero). O seu objetivo é **apagar esse modelo ruim** e substituí-lo por uma arquitetura avançada de Transfer Learning.

Além disso, você aplicará a prática profissional de **Experiment Tracking (MLOps)**, registrando automaticamente todas as suas hipóteses no arquivo `historico_experimentos.csv` e emitindo sua conclusão de engenharia no terminal.

2 Dinâmica de Sala de Aula (60 minutos)

Etapa	Duração	Atividade Principal
Etapa 1	05 min	Rodar o modelo inicial básico (Cenário 0) e registrar o resultado ruim.
Etapa 2	15 min	Implementar a MobileNetV2 congelada (Cenário A - Padrão Ouro).
Etapa 3	20 min	Executar os Cenários de armadilha (Descongelamento e Flatten).
Etapa 4	15 min	Preencher a Conclusão de Engenharia com base no CSV acumulado.
Etapa 5	05 min	Submissão final via <code>git commit</code> e <code>git push</code> no GitLab.

3 Passo a Passo do Laboratório

3.1 Etapa 1: A Dor de Aprender do Zero (Cenário 0)

Abra o arquivo `src/transfer_learning_starter.py` e execute-o do jeito que está:

```
python src/transfer_learning_starter.py
```

Observe o CSV gerado no terminal. A acurácia será ruim e a loss não convergirá direito. Os testes no `pytest tests/test_transfer.py` vão reprovar seu modelo.

3.2 Etapa 2: A Mágica do Transfer Learning (Cenário A)

Localize a função `criar_modelo_transfer()` no código. Apague a CNN básica e implemente a verdadeira anatomia do Transfer Learning, conforme o quadro cirúrgico abaixo:

```
def criar_modelo_transfer():
    # 1. O BACKBONE (O "Cérebro" pré-treinado no ImageNet)
    base_model = MobileNetV2(
        weights='imagenet',
        include_top=False,          # <-- Remove a cabeça de 1000 classes originais
        input_shape=(96, 96, 3)
    )

    # 2. O CONGELAMENTO (Garante que os 2.2M pesos não sejam corrompidos)
    base_model.trainable = False
```

```
# 3. A CABEÇA ADAPTADORA (A "Especialização" para Avião vs Automóvel)
modelo = Sequential([
    base_model,
    GlobalAveragePooling2D(), # <-- Reduz os mapas de convolução para um vetor plano
    Dense(16, activation='relu'),
    Dropout(0.3),             # <-- Previne memorização (Overfitting)
    Dense(1, activation='sigmoid') # <-- Probabilidade final (Avião=0, Auto=1)
])

modelo.compile(optimizer='adam', loss='binary_crossentropy', metrics=['accuracy'])
return modelo, base_model
```

Altere a variável no topo do arquivo para DESCRICAO_EXPERIMENTO = "Cenário A (Padrão GAP)", execute novamente e celebre: **> 98% de acurácia treinando apenas 20.000 parâmetros!**

3.3 Etapa 3: A Armadilha do Descongelamento (Cenário B)

- O que acontece se treinarmos todos os 2,2 milhões de pesos com o gradiente maluco da nova cabeça densa?
- Altere DESCRICAO_EXPERIMENTO = "Cenário B (Descongelado)".
- Mude `base_model.trainable = True`.
- Execute e veja o resultado no CSV (Atenção ao tempo de treinamento e à queda na generalização!).

3.4 Etapa 4: A Explosão Dimensional (Cenário C)

- Restaure para o Cenário A (congele a base).
- Altere DESCRICAO_EXPERIMENTO = "Cenário C (Flatten)".
- Troque `GlobalAveragePooling2D()` por `Flatten()`.
- Execute e observe na tabela CSV a diferença bizarra de **Parâmetros Treináveis**. O modelo fica enorme sem necessidade.

4 O Painel de Rastreamento Acumulado

Ao final da Etapa 4, o seu terminal exibirá o painel completo:

```
=====
PAINEL DE EXPERIMENTOS ACUMULADOS (historico_experimentos.csv)
=====
```

#	Descrição	Treináveis	Épocas	Loss	Acurácia	Status
1	Cenário 0 (Rede Básica)	553,233	5	0.6931	55.20%	Reprovado
2	Cenário A (Padrão GAP)	20,513	5	0.0481	98.10%	Nível Ouro
3	Cenário B (Descongel.)	2,278,497	3	0.1241	91.20%	Aprovado
4	Cenário C (Flatten)	184,336	4	0.0825	95.50%	Nível Ouro

```
=====
```

5 Etapa 5: Conclusão de Engenharia

Com a tabela `historico_experimentos.csv` pronta, vá até o final do arquivo `src/transfer_learning_starter.py` e escreva sua conclusão na variável `CONCLUSAO_ALUNO`. Você deve responder: 1. Qual o impacto de congelar a base (Cenário A vs B)? 2. Qual a importância de usar `GlobalAveragePooling2D` em vez de `Flatten()`? 3. Qual experimento obteve a engenharia mais otimizada?

6 Validação e Submissão (GitLab CI/CD)

Restaure o código para o **Cenário A (Padrão Ouro)** e execute a telemetria do laboratório:

```
# 1. Avalie as suas métricas e o congelamento
pytest tests/test_transfer.py -v -s

# 2. Submeta o trabalho com o histórico preenchido
git add src/transfer_learning.py historico_experimentos.csv transfer_learning.png
git commit -m "Lab 07 finalizado com telemetria MLOps"
git push origin main
```