

AT1 — Criação e Análise de Conjunto de Dados

Classificação de Posturas Humanas com o Dataset MPII

Manuel Lucala Zengo

25 de maio de 2025

Sumário

| | | |
|----------|--|-----------|
| 1 | Introdução | 3 |
| 2 | Metodologia | 3 |
| 2.1 | Conversão das Anotações | 3 |
| 2.2 | Validação Estrutural | 3 |
| 2.3 | Análise Estatística Descritiva | 3 |
| 2.4 | Classificação Automática das Posturas | 4 |
| 2.4.1 | Justificativa da Técnica | 4 |
| 2.4.2 | Regras de Classificação | 4 |
| 2.4.3 | Resultados da Classificação | 5 |
| 2.5 | Seleção e Geração de Exemplos Visuais | 5 |
| 2.5.1 | Estrutura das Pastas Geradas | 5 |
| 3 | Resultados | 5 |
| 3.1 | Distribuição das Classes | 5 |
| 3.2 | Exemplos Visuais | 6 |
| 3.3 | Análise Estatística Final | 6 |
| 4 | Análise Gráfica dos Dados | 6 |
| 4.1 | Distribuição do Número de Joints | 6 |
| 4.2 | Boxplot do Número de Joints | 7 |
| 4.3 | Histograma do Número de Joints | 8 |
| 4.4 | Presença ou Ausência de Anotação de Atividade | 8 |
| 5 | Discussão | 9 |
| 5.1 | Limitações | 9 |
| 5.2 | Possíveis Melhorias Futuras | 9 |
| 6 | Como Executar o Projeto | 9 |
| 7 | Referências | 9 |
| 8 | Estrutura do Diretório do Projeto | 10 |
| 8.1 | Organização do Projeto | 10 |
| 8.2 | Descrição dos Principais Diretórios e Arquivos | 10 |

1 Introdução

O objetivo desta atividade foi selecionar, organizar e analisar um conjunto de dados contendo ao menos três classes distintas de posturas humanas: *em pé*, *sentado* e *em movimento*.

O dataset utilizado foi o **MPII Human Pose Dataset**, um dos principais benchmarks para avaliação de modelos de *pose estimation*, contendo mais de 40 mil pessoas anotadas com pontos articulares.

Devido à ausência de rótulos explícitos de postura no MPII, adotamos técnicas de classificação automática baseadas na geometria dos pontos articulares (*joints*), extraídos das anotações do dataset.

2 Metodologia

2.1 Conversão das Anotações

O dataset original fornece anotações no formato `.mat` (estrutura MATLAB). Para facilitar o processamento, desenvolvemos o script `convert_mpii_mat_to_json.py`, que converteu todas as anotações para o formato `.json`.

Cada imagem passou a possuir um arquivo `.json` com as seguintes informações estruturadas:

- **Nome da imagem**
- **Lista de pontos articulares (*joints*):** coordenadas (x, y), identificador e visibilidade.
- **Atividade (quando disponível):** nome e categoria.

Essa conversão foi fundamental para permitir a manipulação e análise com ferramentas baseadas em Python.

2.2 Validação Estrutural

Após a conversão, realizamos uma **validação estrutural** com o script `validate_json.py`, garantindo que:

- Todos os arquivos contivessem os campos essenciais (`image`, `joints`).
- A estrutura fosse consistente com o esperado.

Foram processados **49.978 arquivos JSON**, com **100% de validade**, o que atesta a qualidade do processo de conversão.

2.3 Análise Estatística Descritiva

Utilizamos **Python** e bibliotecas como `pandas`, `matplotlib` e `seaborn` para realizar uma análise descritiva completa do dataset:

- Distribuição do número de pontos articulares (*joints*) por imagem.

- Estatísticas descritivas: média, mediana, desvio padrão.
- Análise da presença ou ausência de anotações de atividades.

As análises evidenciaram que a maioria das imagens possui cerca de 16 joints e que **100% das imagens não possuem anotações de atividade**, conforme ilustrado na seção de gráficos.

2.4 Classificação Automática das Posturas

2.4.1 Justificativa da Técnica

Dada a ausência de rótulos explícitos de postura, optamos por uma abordagem baseada na **geometria dos pontos articulares**, técnica amplamente utilizada em tarefas de *pose-based classification*.

As variáveis analisadas foram:

- **Altura**: distância vertical entre cabeça e pés.
- **Largura**: distância horizontal entre os ombros.

A razão **Altura/Largura** foi utilizada como critério para inferir a postura, baseado nas seguintes premissas:

- Pessoas em **pé** tendem a ter uma razão maior.
- Pessoas **sentadas** apresentam uma razão intermediária.
- Pessoas em **movimento** podem ter uma razão menor, devido a posturas dinâmicas ou inclinadas.

Essa abordagem é preferível em datasets sem rótulos explícitos, sendo uma solução eficiente para gerar pseudo-rótulos.

2.4.2 Regras de Classificação

A classificação automática foi realizada conforme as seguintes regras:

- $\frac{Altura}{Largura} > 1.5$: **em pé**
- $0.8 \leq \frac{Altura}{Largura} \leq 1.5$: **sentado**
- $\frac{Altura}{Largura} < 0.8$: **em movimento**

Essa lógica foi implementada no script `classify_postures.py`.

2.4.3 Resultados da Classificação

O resultado da classificação automática foi o seguinte:

- **Em pé:** 23.452 imagens
- **Sentado:** 448 imagens
- **Em movimento:** 386 imagens
- **Não classificado:** 25.692 imagens

Observamos um claro **desbalanceamento entre as classes**, o que é uma limitação importante a ser considerada em futuras etapas de modelagem.

2.5 Seleção e Geração de Exemplos Visuais

Para criar subconjuntos representativos, selecionamos automaticamente **100 imagens por classe**.

A seleção priorizou imagens com **maior número de joints**, sob a hipótese de que imagens mais completas oferecem maior representatividade da postura.

O processo foi realizado com o script `draw_samples_by_class.py`, que:

- Desenhou os pontos articulares sobre as imagens.
- Salvou os exemplos organizados por classe.

2.5.1 Estrutura das Pastas Geradas

```
mpii_dataset/sample_visuals/  
    em_pe/  
    sentado/  
    movimento/
```

Essa organização visa facilitar o treinamento e a validação de modelos de classificação de posturas.

3 Resultados

3.1 Distribuição das Classes

Após a classificação e seleção, obtivemos:

- **100 exemplos** da classe *em pé*
- **100 exemplos** da classe *sentado*
- **100 exemplos** da classe *em movimento*

3.2 Exemplos Visuais



Figura 1: Exemplos representativos das classes: em pé, sentado e movimento.

Esses exemplos confirmam visualmente a adequação da regra de classificação geométrica adotada.

3.3 Análise Estatística Final

- Média de joints: **17.3**
- Máximo: **256 joints**
- Mínimo: **0 joints**
- Desvio padrão: **20.43**

4 Análise Gráfica dos Dados

4.1 Distribuição do Número de Joints

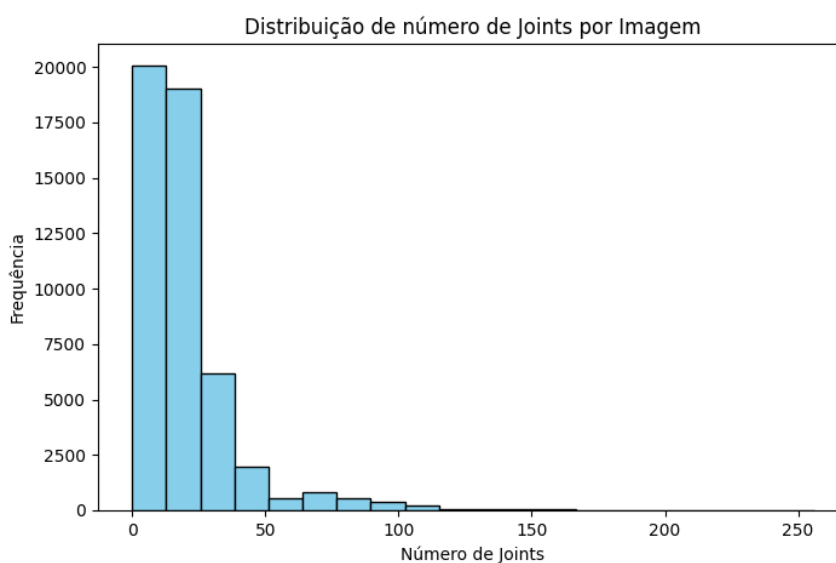


Figura 2: Distribuição do número de joints por imagem.

O gráfico demonstra que a maioria das imagens possui até 20 pontos articulares anotados, com distribuição altamente assimétrica.

4.2 Boxplot do Número de Joints

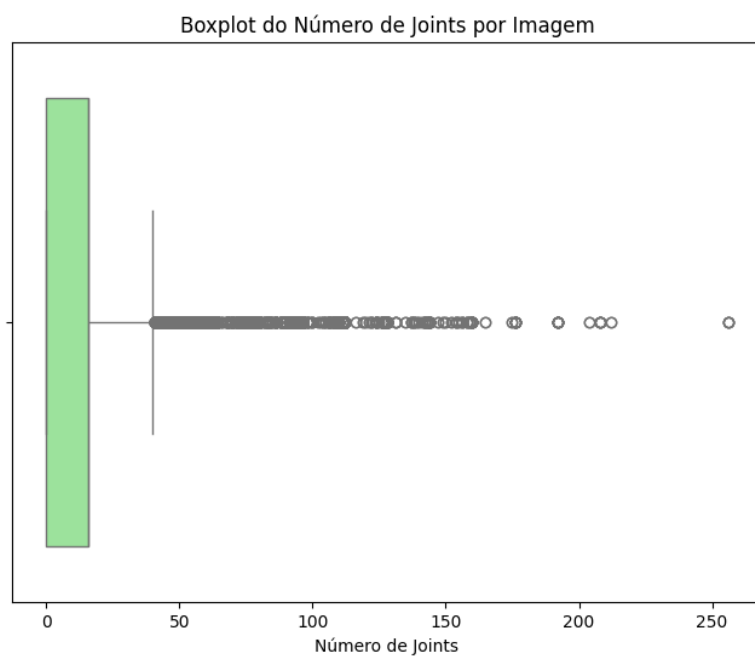


Figura 3: Boxplot da distribuição do número de joints.

O boxplot evidencia uma concentração de imagens com número reduzido de joints, além da presença de outliers significativos, chegando a 256 joints.

4.3 Histograma do Número de Joints

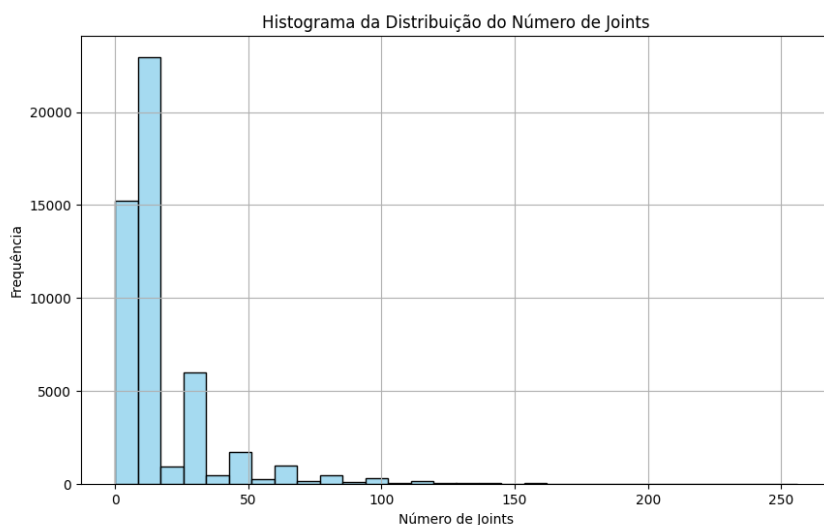


Figura 4: Histograma da distribuição do número de joints por imagem.

A distribuição apresenta cauda longa, com a maioria das imagens apresentando poucos joints e raros casos com valores extremos.

4.4 Presença ou Ausência de Anotação de Atividade

Distribuição: Presença de Anotação de Atividade

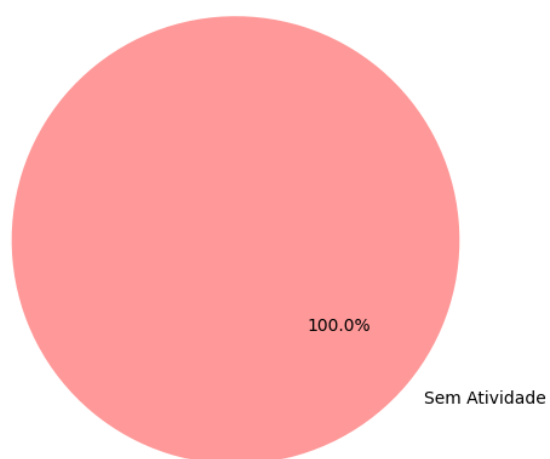


Figura 5: Distribuição da presença ou ausência de anotação de atividade.

Como podemos observar, a totalidade das imagens **não possui anotações de atividades**. Esse aspecto reforça a importância da classificação automática de posturas como abordagem alternativa.

5 Discussão

5.1 Limitações

- O dataset não possui rótulos explícitos de postura — a classificação inferida pode conter ruídos e ambiguidades.
- Alto número de imagens com pontos incompletos, resultando em uma grande quantidade de imagens não classificadas.
- Desequilíbrio entre classes: muito mais imagens classificadas como *em pé* do que como *sentado* ou *em movimento*.

5.2 Possíveis Melhorias Futuras

- Aplicar técnicas de **semi-supervised learning** para refinar a classificação automática.
- Explorar o uso de **modelos de aprendizado profundo baseados em grafos** (*Graph Convolutional Networks*) para inferência direta de posturas.
- Aumentar a base de dados com imagens rotuladas de forma supervisionada para reduzir ruídos.

6 Como Executar o Projeto

1. Instale as dependências: `numpy`, `opencv-python`, `matplotlib`, `seaborn`, `pandas`.
2. Execute os scripts na seguinte ordem:
 - `python convert_mpii_mat_to_json.py`
 - `python validate_json.py`
 - `python classify_postures.py`
 - `python draw_samples_by_class.py`
 - `python generate_graphs.py`
3. Alternativamente, utilize o notebook `AT1_analysis.ipynb` para uma execução interativa e visual.

7 Referências

- MPII Human Pose Dataset: <http://human-pose.mpi-inf.mpg.de/>
- Andriluka et al., 2014: *2D Human Pose Estimation: New Benchmark and State of the Art Analysis*.

8 Estrutura do Diretório do Projeto

8.1 Organização do Projeto

A seguir, apresentamos a estrutura do projeto com os principais arquivos e diretórios gerados ao longo da execução da AT1:

```
.
AT1_analysis.ipynb
convert_mpii_mat_to_json.py
classify_postures.py
draw_samples_by_class.py
extract_mpii.py
generate_graphs.py
validate_json.py
README.md
mpii_dataset
  annotations
    bsd.txt
    mpii_human_pose_v1_u12_1.mat
    README.md
  annotations_json
  dataset_annotations.json
  images
  posture_classes
    em_pe.txt
    movimento.txt
    nao_classificado.txt
    sentado.txt
  sample_visuals
    em_pe
    movimento
    sentado
data
  extraction_log.txt
  mpii_human_pose_v1.tar.gz
  mpii_human_pose_v1_u12_2.zip
```

8.2 Descrição dos Principais Diretórios e Arquivos

- **AT1_analysis.ipynb**: Notebook contendo toda a análise estatística e visual.
- **Scripts Python**: automação de todas as etapas da atividade (conversão, validação, classificação, visualização).
- **data/**: Arquivos brutos do dataset, incluindo o pacote original (`.tar.gz`) e anotações adicionais.
- **mpii_dataset/**:
 - **annotations/**: Arquivos de anotação originais em formato MATLAB.

- `annotations_json/`: Anotações convertidas para formato JSON.
- `dataset_annotations.json`: Anotações consolidadas.
- `images/`: Imagens do dataset (aproximadamente 12GB).
- `posture_classes/`: Arquivos `.txt` com as imagens classificadas por classe.
- `sample_visuals/`: Exemplos visuais de cada classe com *joints* desenhados.