Universidade da Beira Interior
Departamento de Informática

Using Vision Language Models for Fashion Characterization in Crowds

Elaborado por:

Matilde Silva
49582

Orientador:

Professor Doutor Hugo Proença

25 de junho de 2025

Resumo

Este projeto propõe o desenvolvimento de um Modelo de Visão-Linguagem, *Vision-Language Model* (VLM), capaz de analisar e classificar, focando-se no tipo, as roupas utilizadas por indivíduos numa multidão em imagens ou fluxos de vídeos.

Para isto, será adotada uma abordagem de aprendizagem profunda multimodal para o desenvolvimento de um modelo que integre técnicas de visão por computador com processamento de linguagem natural.

O modelo será treinado com diversos conjuntos de dados que contém atributos de moda rotulados, de modo a garantir um desempenho robusto em cenários com diferentes condições de iluminação, perspetivas e variações culturais.

Entre as diversas potenciais aplicações para este modelo, destacam-se o aperfeiçoamento de sistemas de vigilância, melhorar a acessibilidade para as pessoas com deficiência visual, e a implementação de mecanismos para processamento em tempo real que permitem garantir uma operação precisa e eficiente em ambientes dinâmicos.

Agradecimentos

A realização deste projeto não teria sido possível sem a colaboração de diversas pessoas, às quais queria deixar o meu agradecimento por todo o apoio oferecido durante a elaboração do mesmo.

Em primeiro lugar, gostaria de agradecer ao orientador deste projeto Professor Doutor Hugo Proença, pela oportunidade e confiança depositada na realização deste trabalho. Estendo ainda o meu agradecimento pela orientação oferecida, ao longo de todo o semestre, e pela transmissão de conhecimentos relevantes no desenvolvimento das várias atividades do projeto e nas contribuições que levaram ao enriquecimento deste relatório.

Aos meus colegas de curso e amigos, deixo o agradecimentos por todo o apoio e troca de conhecimentos que permitiram tornar esta fase mais leve e produtiva.

Por fim, agradeço à minha família, por todo o apoio incondicional e incentivo oferecido em todos os momentos.

Conteúdo

Lista de Figuras

Lista de Tabelas

Acrónimos

VLM *Vision-Language Model*

HPE *Human Pose Estimation*

IA Inteligência Artificial

CNN *Convolucional Neural Network*

ROI *Region Of Interest*

VSC *Visual Studio Code*

PAFs *Part Affinity Fields*

MS COCO *Microsoft Common Objects in Context*

ECCV European Conference on Computer Vision

CVPR *Computer Vision and Pattern Recognition*

Capítulo

1

Introdução

1.1 Enquadramento

Com os avanços atuais na área da inteligência artificial, principalmente no que diz respeito a uma aprendizagem profunda multimodal, os *Vision-Language Model* (VLM) têm surgido como ferramentas valiosas de análise e descrição de imagens com base no seu conteúdo visual. Estes modelos integram técnicas de visão por computador com processamento de linguagem natural, permitindo gerar descrições textuais coerentes e contextualizadas.

Esta capacidade de classificação demonstra utilidade nas mais diversas áreas e domínios, desde segurança e vigilância, a acessibilidade para pessoas com deficiências visual e na implementação de mecanismos para processamento em tempo real em ambientes mais dinâmicos.

Neste contexto, este projeto visa o desenvolvimento de um VLM capaz de descrever a roupa de um indivíduo, focando-se no seu tipo, através de uma abordagem que permita garantir robustez face a diferentes condições ambientais e culturais.

1.2 Motivação

A capacidade de analisar o tipo de roupa de um indivíduo, como já foi referido anteriormente, pode representar um valor significativo nas mais diversas aplicações.

Entre elas, o aperfeiçoamento de sistemas de vigilância pode verificar-se como sendo essencial na identificação e rastreamento de indivíduos com base no seu vestuário. Para pessoas com deficiência visual, poderá representar uma forma de suporte visual que providencie auxílio em termos de aces-

sibilidade. Este modelo poderá ainda verificar-se útil em outros tipos de ambientes que possam ser influenciados por análises baseadas nas tendências e preferências de consumidores.

Com base no atual desenvolvimento significativo em modelos VLM, a realização deste projeto motiva a exploração das soluções mais robustas e adaptáveis, através da elaboração de uma comparação, em termos do desempenho obtido a partir da utilização de modelos desta última geração.

1.3 Objetivos

Este projeto tem como principal objetivo desenvolver um VLM capaz de analisar e classificar o tipo de peça de vestuário de um indivíduo numa multidão em imagens ou vídeos.

Os objetivos principais incluem:

- Integrar técnicas de visão computacional com modelos de linguagem natural;
- Explorar e comparar diferentes formas de interação com modelos pré-treinados;
- Avaliar o desempenho das diferentes abordagens em múltiplos cenários, considerando fatores como iluminação, ângulo de visão e diversidade cultural no vestuário;
- Implementar soluções que permitam o processamento em tempo real, otimizando a aplicação em sistemas dinâmicos.

1.4 Estrutura do relatório

De modo a refletir o trabalho que foi feito, este documento encontra-se estruturado da seguinte forma:

1. O primeiro capítulo – **Introdução** – apresenta o projeto, a motivação para a sua escolha, o enquadramento para o mesmo, os seus objetivos e a respetiva organização do documento.
2. O segundo capítulo – **Estado da arte** – descreve os principais conceitos essenciais para a compreensão do contexto do projeto.
3. O terceiro capítulo – **Implementação e Testes** – descreve a abordagem e decisões tomadas neste projeto, incluindo ferramentas utilizadas, arquitetura do sistema, métodos selecionados e técnicas utilizadas.

4. O quarto capítulo – **Resultados experimentais** – apresenta as análises finais dos modelos implementados, com base nos resultados obtidos durante o desenvolvimento do projeto.

5. O quinto capítulo – **Conclusões e Trabalho Futuro** – apresenta um balanço crítico do trabalho realizado, destacando os resultados alcançados, as limitações encontradas e possíveis implementações futuras.

Capítulo

2

Estado da Arte

2.1 Introdução

Para uma compreensão mais aprofundada deste projeto é importante começar por entender alguns conceitos relacionados com a Inteligência Artificial e as suas subáreas mais relevantes.

Neste capítulo, irão ser explorados os principais conceitos envolvidos na construção e aplicação dos modelos de VLM baseados em aprendizagem profunda multimodal, visão por computadores e processamento de linguagem natural. Para além disto, serão abordados também temas relacionados com *Human Pose Estimation* e Redes Neuronais.

2.2 Inteligência Artificial (IA)

A IA [4] é uma tecnologia que permite replicar funções cognitivas humanas, tais como aprendizagem, compreensão, resolução, tomada de decisões, criatividade e autonomia. Esta tecnologia tem vindo cada vez mais a ser utilizada, provocando avanços significativos nas mais diversas áreas de trabalho, entre estas, saúde, descobertas científicas, setores financeiros, proteção ambiental, cibersegurança, educação, entre outras.

Para melhor entender como é que se comportam os sistemas IA é importante conhecer as diferentes técnicas com que estes podem operar.

2.2.1 Aprendizagem Supervisionada

A Aprendizagem Supervisionada é uma técnica fundamental para que o sistema consiga reconhecer e prever padrões nos dados, permitindo desta forma

mapear corretamente um entrada para a saída correspondente. Para isto, este tipo de sistemas aprendem a partir de um conjunto de dados rotulados, em que a cada ponto está associado a um resultado conhecido.

2.2.2 Aprendizagem Não-Supervisionada

A Aprendizagem Não-Supervisionada consiste numa técnica em que o sistema, contrariamente à descrita anteriormente, aprende com dados não rotulados e que não estão associadas a um resultado previamente definido. Desta forma, o principal objetivo implica a deteção de padrões nos dados sem qualquer conhecimento prévio.

2.2.3 Aprendizagem Por Reforço

Neste tipo de aprendizagem, o agente aprende a tomar decisões através da interação com o ambiente, recebendo recompensas ou penalizações com base nas ações que executa, sempre com o objetivo de maximizar a recompensa acumulada ao longo do tempo.

2.2.4 Aprendizagem Profunda

A Aprendizagem Profunda é um subconjunto da aprendizagem automática baseado em redes neuronais em múltiplas camadas, em que o agente aprende padrões e representações complexas a partir dos dados. Quanto maior for o número de camadas ocultas, de onde origina o termo "profunda", maior será a capacidade de representar estas relações complexas.

2.2.4.1 Aprendizagem Profunda Multimodal

No contexto de Aprendizagem Profunda, a Aprendizagem Profunda multimodal distingue-se pelo tipo e quantidade de dados que são processados em simultâneo. Modelos baseados em Aprendizagem Profunda Multimodal, ao contrário dos convencionais, têm a capacidade de processar e integrar múltiplos tipos de dados ao mesmo tempo, através da combinação de diferentes fontes de informação, como imagem, texto, vídeo, áudio, entre outros, com o intuito de obter resultados mais ricos e contextualizados do fenómeno em análise.

2.2.5 Processamento de Linguagem Natural

O Processamento de Linguagem Natural permite ao sistema a capacidade de processar, analisar, e responder a um texto ou discurso de uma forma que se assemelha à linguagem humana.

2.2.6 Visão Computacional

Visão Computacional [5] é um campo da IA, que utiliza Aprendizagem Automática e Redes Neuronais para treinar uma máquina ou um sistema de forma a gerar informação essencial a partir de uma imagem, vídeo ou outros tipos de recursos visuais, permitindo ainda detetar possíveis erros ou falhas nas imagens.

De uma forma resumida, permite realizar funções humanas, como por exemplo distinguir objetos independentemente da distância a que se encontram, do tipo de movimento que descrevem, entre outros, no entanto através de uma abordagem muito mais rápida e que supera na grande maioria dos casos a capacidade humana.

É a partir da integração de Visão Computacional e do processamento de linguagem natural que irão ser criados os VLM propostos.

2.3 *Human Pose Estimation* (HPE)

HPE é uma das áreas de pesquisa inseridas na IA mais interessantes e que tem vindo a ser cada vez mais utilizada devido à sua versatilidade e utilidade, para além das diferentes aplicações que pode tomar nas mais diversas áreas e domínios. [1]

De uma forma generalizada, esta é uma abordagem que permite identificar e classificar articulações do corpo humano. Isto torna-se possível através da deteção das coordenadas de cada articulação em pontos-chave. Quando a conexão é significativa o suficiente, estes pontos formam pares, até assumirem a forma de um esqueleto do corpo humano.

Existem três tipos principais de abordagens para modelos do corpo humano:

1. ***Skeleton-based model*:** esta é a abordagem utilizada neste projeto, e permite detetar um conjunto de pontos-chave conectados por segmentos;

2. ***Countour-based model*:** usa o contorno externo do corpo para formar a silhueta;

Figura 2.1: *Human Pose Estimation* (fonte: [1])

3. ***Volume-based model*:** representa o corpo como um volume tridimensional.

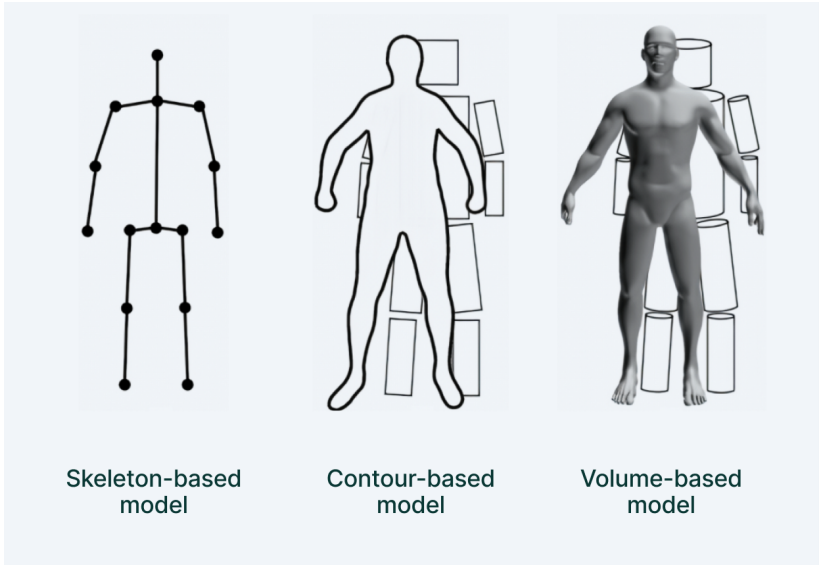

Figura 2.2: Modelos do Corpo Humano (fonte: [1])

No contexto de Visão Computacional, a HPE é usado como forma de extrair informação do corpo humano e aprender a sua geometria, através de duas abordagens principais:

- ***Abordagem Clássica*:** refere-se a técnicas e métodos que envolvem geralmente algoritmo tradicionais de aprendizagem automática, detetando

as partes do corpo com base em modelações de distribuição de probabilidades sobre possíveis poses.

- ***Abordagem por Aprendizagem Profunda*:** ao contrário das abordagens clássicas, onde as características eram definidas manualmente, neste tipo de abordagem são utilizadas Redes Neuronais para extrair padrões e representações de imagens de forma a conseguir aprender automaticamente características complexas, desde que lhe seja fornecida um conjunto de dados suficientemente grande e bem estruturados para treino, validação e teste. No contexo deste projeto, esta abordagem verifica-se como especialmente útil em tarefas de classificação, deteção e segmentação.

2.4 Redes Neuronais

Redes Neuronais [6] consistem em modelos de Aprendizagem Automática que procuram simular funções complexas do cérebro humano, através da interligação de nós, denominados de neurónios. Os neurónios permitem a aprendizagem de padrões nos dados processados e permitem a execução de tarefas baseadas no reconhecimento de padrões e tomada de decisões. No contexto de Aprendizagem Profunda, o reconhecimento destes padrões é executado sem regras pré-definidas.

As Redes Neuronais são construídas a partir dos seguinte componente:

1. **Neurónios:** unidades básicas que recebem, entradas, cada uma destas acompanhadas de um limiar e de uma função de ativação;
2. **Conexões:** ligações entre os neurónios que carregam informação definida por pesos e *biases*;
3. **Pesos e *Biases*:** parâmetros que definem a força e direção das conexões;
4. **Funções de Propagação:** mecanismo de processamento e transferência de dados entre camadas de neurónios;
5. **Regra de Aprendizagem:** método de ajustes dos pesos e *biases* ao longo do tempo com o objetivo de melhorar a precisão do modelo.

2.4.1 Camadas numa Arquitetura de Redes Neuronais

- **Camada de Entrada:** camada onde são recebidas as entradas ma rede. Cada neurónio corresponde a uma característica dos dados de entrada;

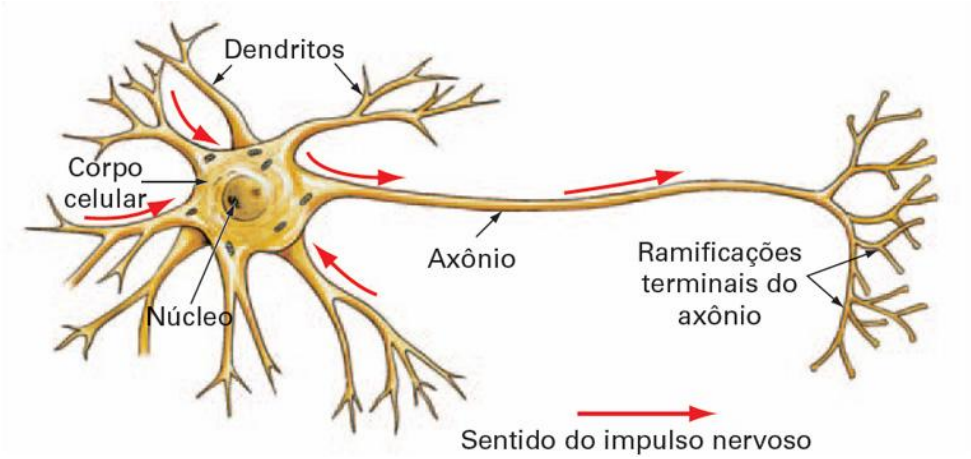

Figura 2.3: Rede Neuronal Artificial (fonte: [2])

- **Camadas Ocultas:** a rede pode ser constituídas por uma ou várias camadas ocultas, cada uma delas correspondente a um conjunto de neurónios responsáveis por transformar entradas em representações úteis na camada de saída;
- **Camada de Saída:** camada que produz o resultado do modelo.

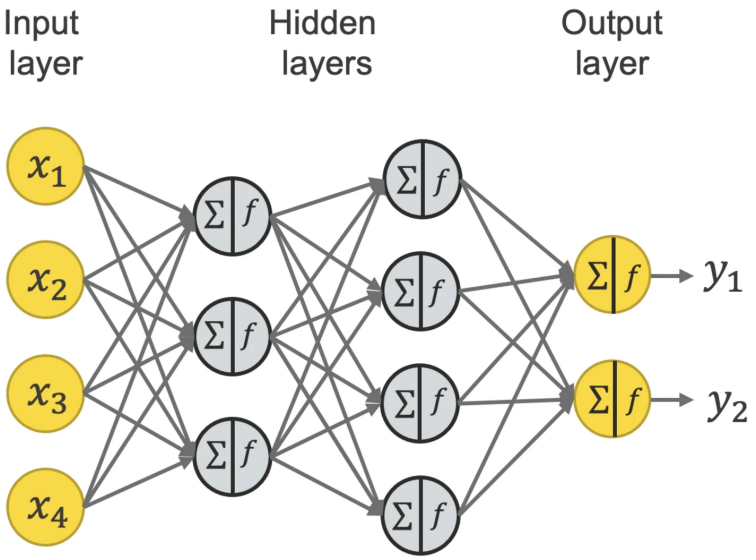

Figura 2.4: Camadas numa Arquitetura de Redes Neuronais (fonte: [3])

2.4.2 Funcionamento de Redes Neuronais

1. ***Forward Propagation*:** quando uma entrada é introduzida na rede, atravessa sequencialmente a camada de entrada, seguida das camadas ocultas, até alcançar a camada de saída. Nesta fase, cada entrada passa pelo seguinte processo:

   a) **Transformação Linear:** as entradas são multiplicadas pelos pesos e associadas a conexões. Estes produtos são de seguida somados entre si e é adicionada a *bias* ao resultado desta soma. Este processo é geralmente representado a partir da fórmula representada abaixo,

   $$z = w_1 x_1 + w_2 x_2 + \cdots + w_n x_n + b \tag{2.1}$$

   onde w representa os pesos, x as entradas e b as *bias*.

   b) **Ativação:** É aplicada uma função de ativação ao resultado da Transformação Linear, representado por z. Alguma das funções de ativação mais utilizadas incluem *ReLU, sigmoid, and tanh*.

2. ***Backpropagation*:** o desempenho da rede é avaliada através do cálculo de uma função de perda que mede a diferença entre o resultado verdadeiro e o output previsto. Esta fase implica:

   a) **Cálculo da Perda:** Algumas escolhas comuns para o cálculo desta perda incluem o Erro Quadrático Médio e Entropia Cruzada. Neste projeto é utilizada a Entropia Cruzada descrita na fórmula abaixo,

   $$\mathscr{L}_{\text{CE}} = -\sum_{c=1}^{C} y_c \log(\hat{y}_c) \tag{2.2}$$

   onde C representa o número de classes, y_c o valor real (1 se a classe correta é c, 0 caso contrário) e $\hat{y}_c$ a probabilidade prevista para a classe c, depois da aplicação de *Softmax*.

   b) **Cálculo do Gradiente:** A rede mapeia os gradientes da função de perda com os respetivos pesos e *bias*. Se a saída do modelo for passada por uma função *Softmax*,

   $$\hat{y}_i = \frac{e^{z_i}}{\sum_{j=1}^{C} e^{z_j}} \tag{2.3}$$

   onde $\hat{y}_i$ é a probabilidade prevista para a classe i, z_i é o *logit* (saída não normalizada) da classe i e C é o número total de classes. Então, o gradiente da perda em relação à entrada z_k (*logit* da classe k) é:

   $$\frac{\partial \mathscr{L}_{\text{CE}}}{\partial z_k} = \hat{y}_k - y_k \tag{2.4}$$

c) **Atualização dos Pesos:** Depois de calculados os gradientes, os pesos e as *bias* são ajustados utilizando um algoritmo de otimização. Os pesos são ajustados na direção contrária dos gradientes de modo a reduzir a perda;

3. **Iteração:** o processo de *Forward Propagation*, cálculo da perda, *Back-propagation* e atualização dos pesos é repetido em várias iterações com o intuito de reduzir, ao longo de cada iteração, a perda e melhorar a precisão das previsões.

2.5 Avaliação dos Modelos

Quando concluído o treino dos modelos, estes devem ser avaliados de acordo com o seu desempenho e capacidade de classificação. Neste projeto, é utilizada uma Arquitetura Neuronal Convolucional *Inception ResNet v2*, que combina os princípios das arquiteturas *Inception* com ligações residuais.

Rede Neuronais Convolucionais, *Convolucional Neural Network* (CNN), no contexto de Redes Neuronais, distinguem-se pelo uso de filtros locais e partilha de pesos, de modo a extrair automaticamente padrões espaciais das imagens, partindo da utilização de menos parâmetros e de uma melhor eficiência em comparação a redes tradicionais.

Mais detalhes acerca da arquitetura utilizada na avaliação destes modelos podem ser consultados na secção 3.4.2, onde é feita uma descrição mais aprofundada do modelo utilizado neste projeto.

2.6 Conclusões

Este capítulo, visa contribuir com o enquadramento necessário para melhor compreender os principais desafios, qualidades e potenciais aplicações do tipo de sistemas explorados neste projeto.

Foram abordados diversos pontos-chave, que sustentam os VLM, desde os princípios fundamentais da IA e as suas áreas de pesquisa, às abordagens mais recentes baseadas em aprendizagem profunda multimodal e a técnicas que implicam Visão Computacional, processamentos de linguagem natural, HPE e Redes Neuronais.

Este conhecimento base, foi explorado de forma a justificar possíveis decisões de metodologias tomadas ao longo do projeto.

Capítulo

3

Implementação e Testes

3.1 Introdução

Ao longo deste capítulo são apresentadas as principais ferramentas utilizadas, bem como os métodos e técnicas selecionados de modo a sustentar a implementação deste projeto. Além disso, é também descrita a arquitetura do sistema, detalhando as várias fases do processo e a forma como estas interagem entre si.

Com isto, o principal objetivo do capítulo reside numa contextualização de todas as decisões técnicas tomadas ao longo do desenvolvimento do trabalho, como o intuito de atingir os objetivos propostos.

3.2 Ferramentas Utilizadas

3.2.1 *Hardware*

Na tabela 3.1 é possível visualizar as especificações relativos ao *Hardware* utilizado no desenvolvimento deste trabalho.

Tabela 3.1: Específicações do Hardware Utilizado

Nome	Descrição
RAM	12GB
CPU	11th Gen Intel(R) Core(TM) i7-116567 @ 2.80GHz
GPU	NVIDEA GeFORCE MX350

3.2.2 Ambiente de Desenvolvimento

Este projeto foi realizado no Sistema Operativo *Windows 11 HOME 64bit*, versão 23H2.

3.2.3 *Visual Studio Code* (VSC)

O VSC, ambiente de desnvolvimento utilizado neste projeto, consiste num editor de código-fonte e num *software* livre de código aberto desenvolvido pela *Miscrosoft* para *Windows, Linux e macOS*. Esta ferramenta, conhecida pela sua versatilidade, desempenho e extensibilidade, inclui funcionalidades tais como suporte de depuração, controlo de versões GIT, realce de sintaxe, auto-completar inteligente e uma vasta biblioteca de extensões que permitem a personalização do ambiente de desenvolvimento de acordo com as necessidade do projeto.

3.2.4 Linguagem de Programação

Neste Projeto, foi utilizada a linguagem de programação ***Python***, amplamente utilizada no âmbito de ciências de dados, IA, aprendizagem automática e desenvolvimento de *Software*. A utilização desta linguagem deve-se ao facto da sua simples sintaxe e de esta se encontrar aliada a um vasto conjunto de bibliotecas como *Pytorch*, *Numpy*, *Pandas*, entre outras.

Para além da linguagem identificada anteriormente, foi ainda utilizada a linguagem ***Bash***, para sistemas *Unix/Linux*, na preparação e organização dos dados e na execução das tarefas repetitivas no sistema de ficheiros.

3.2.5 *Pytorch*

Pytorch é uma *framework* amplamente utilizada em aplicações de aprendizagem automática, mais especificamente, de aprendizagem profunda, selecionada devido à sua facilidade de desenvolvimento, experimentação e depuração de modelos complexos. Esta é caracterizada pela sua interface intuitiva, baseada em *Python* e pelo uso de grafos computacionais dinâmicos que permitem acelerar significativamente o treino dos modelos. O *Pytorch* foi a ferramenta utilizada pelo método *Realtime MultiPerson Pose Estimation*, descrito na secção 3.4.1, e na construção, treino e avaliação dos modelos VLM criados a partir do método de Redes Neuronais Convolucionais *Inception-ResNet Pytorch Implementation*, descrito na secção 3.4.2.

3.2.6 Ambientes Virtuais

Os Ambientes Virtuais representam uma solução essencial a utilizar neste projeto, pois permitem editar, sem ao mesmo tempo danificar os arquivos do ambiente de desenvolvimento principal. No desenvolvimento deste trabalho, foram criados dois ambientes virtuais:

- ***pose-env-mod1*:** para o método *Realtime MultiPerson Pose Estimation*
- ***test_env*:** para o método *Inception-ResNet Pytorch Implementation*

Através das instruções do guia disponibilizado na bibliografia [7], é possível criar e configurar um ambiente virtual.

3.2.7 Bibliotecas

As bibliotecas apresentadas na Tabela 3.2 correspondem às principais dependências utilizadas diretamente pelo método *Realtime Multi-Person Pose Estimation*, ainda que tenham sido também consultados outros exemplos complementares.

Pacote	**Versão**	**Utilização**
matplotlib	3.10.1	Visualização de dados (gráficos 2D)
numpy	1.23.5	Cálculo numérico com *arrays* multidimensionais
opencv-python-headless	4.11.0.86	Processamento de imagem (sem GUI)
pillow	11.1.0	Processamento de imagens (*PIL fork*)
progress	1.6	Barras de progresso no terminal
pycocotools	2.0.8	Ferramentas COCO para anotação de imagem
python-dateutil	2.9.0.post0	Manipulação avançada de datas
scipy	1.15.2	Computação científica e estatística
torch	2.0.1+cu117	*Framework* de aprendizagem profunda *(PyTorch)*
torchvision	0.15.2+cu117	Conjuntos de dados e modelos visuais para *PyTorch*
yacs	0.1.8	Gestão de ficheiros de configuração

Tabela 3.2: Principais bibliotecas utilizadas pelo método *Realtime MultiPerson Pose Estimation*

Na tabela 3.3 é possível observar a lista de principais bibliotecas utilizadas diretamente pelo método *Inception-ResNet Pytorch Implementation.*

Bilblioteca	Versão	Utilização
matplotlib	3.10.1	Criação de gráficos e visualizações em 2D
numpy	2.2.4	Computação científica e manipulação de *arrays*
pandas	2.2.3	Análise e manipulação de dados em formato tabular
python-dateutil	2.9.0.post0	Manipulação flexível de datas e horas
scikit-learn	1.6.1	Aprendizagem automática clássica
torch	2.6.0	*Framework* para aprendizagem profunda com aceleração em GPU
torchvision	0.21.0	Conjuntos de dados, transformações e modelos para visão computacional com *PyTorch*

Tabela 3.3: Principais bibliotecas utilizadas pelo método *Inception-ResNet Pytorch Implementation*

3.3 Técnicas Utilizadas

3.3.1 *Data Augmentation*

Aumentar o número de dados de treino é uma das abordagens mais eficientes a tomar quando o objetivo passa por melhorar o desempenho do modelo sem ter que recolher dados adicionais.

Data Augmentation [8] [9] consiste numa técnica que envolve a criação de novos dados de treino através da aplicação de transformações nos dados já existentes. Desta forma, é possível obter um modelo treinado com um conjunto de dados ampliado e mais diversificado, promovendo uma maior capacidade de generalização e contribuindo para a mitigação do problema de sobreajuste.

De seguida irão ser exploradas algumas das técnicas utilizadas com mais frequência, implementadas em *Pytorch*.

1. ***Random Horizontal Flip*:** vira a imagem horizontalmente;
2. ***Random Rotation*:** roda a imagem em ângulos aleatórios;
3. ***Color Jitter*:** permite alterar o brilho, contraste, saturação e tom da imagem;
4. ***Random Crop*:** corta uma parte aleatória da imagem;
5. ***Random Gray Scale*:** converte a imagem para tons de cinzento aleatoriamente com base numa dada probabilidade;

6. ***Random Perspective*:** aplica uma perspetiva aleatória baseada no valor de uma dada probabilidade;
7. ***Scale*:** aumentar ou diminuir o tamanho da imagem, sem alterarção do valor do raio;
8. ***Resize*:** redimensionar o tamanho da imagem de acordo com o tamanho requerido. Esta transformação pode implicar distorções na imagem, dependendo dos valores dados;
9. ***ToTensor*:** converte imagens nos formatos PIL.image e NumPy.ndarray num *tensor* em *Pytorch*, essencial para imagens de treino de modelos em *Pytorch*;
10. ***Normalize*:** aplicada em imagens já convertidas em *tensor*, tendo como objetivo normalizar os valores dos *píxeis*, ajustando-os com base numa dada média e desvio padrão. Fornece auxílio ao modelo, de modo a que este treine mais rapidamente.

Neste projeto, de modo a expandir o conjunto de dados de treino e ao mesmo tempo não comprometer ou distorcer o conteúdo representado nas imagens, foram selecionadas técnicas como o *Random Horizontal Flip*, *Color Jitter* e *Random Gray Scale*, aplicadas conjuntamente nas imagens de forma aleatória. Para além das técnicas identificadas anteriormente, foram ainda utilizadas as técnicas *Normalize* e *ToTensor* no processo de treino de cada um dos modelos propostos.

3.3.2 *Region Of Interest* (ROI)

Uma ROI [10] refere-se a uma área específica de uma imagem ou vídeo, assumindo geralmente a forma de um retângulo ou polígono, que é selecionada para análise ou processamento. Esta área é delimitada, manualmente pelo utilizador ou automaticamente por um algoritmo de processamento de imagem, com base na informação relevante para uma determinada aplicação.

3.4 Métodos selecionados

3.4.1 *Keras Realtime MultiPerson Pose Estimation*

Este método apresenta uma versão *Keras* de *Realtime MultiPerson Pose Estimation* [11]. No entanto, para a realização deste projeto, foi selecionada a primeira versão em *Pytorch* que se encontrava disponibilizada pelos respetivos autores no repositório.

Este consiste numa abordagem eficiente de estimar a pose em 2D de múltiplas pessoas numa imagem, através de uma representação não paramétrica denominada *Part Affinity Fields* (PAFs). PAFs consistem em campos vetoriais que indicam a conexão entre partes do corpo, sendo utilizados, neste contexto, para aprender a ligar corretamente esses pontos de modo a formar esqueletos humanos completos.

Em termos gerais, existem 4 métodos de HPE:

- ***Generative and Discriminative (3D Single Person)*:** *Generative* modela o corpo e gera poses possíveis, enquanto *Discriminative* aprende diretamente dos dados para prever a pose.
- ***Top Down and Bottom Up (Multi-Person)*:** *Top-Down* deteta primeiro pessoas e depois estima as suas poses, enquanto *Bottom-Up* deteta articulações e depois agrupa-as por pessoa.
- ***Regression and Detection Based (Single Person)*:** *Regression* prevê diretamente as coordenadas dos pontos do corpo, enquanto *Detection-Based* gera mapas de calor para localizar articulações.
- ***One-Stage and Multi-Stage*:** *One-Stage* estima a pose num único passo, enquanto *Multi-Stage* ajusta a pose em várias etapas sucessivas.

A estimativa, neste método em específico, é efetuada através de uma abordagem *Top Down and Bottom Up* e tem o objetivo de aprender a detetar em tempo real as localizações das partes do corpo, uni-las para criar o esqueleto do corpo humano e conjuntamente aprender a associar essas partes a indivíduos numa imagem, independentemente do número de pessoas representadas.

Este método destacou-se principalmente por alcançar o primeiro lugar no *Microsoft Common Objects in Context* (MS COCO) *Keypoints Challenge* de 2016, e para além disso, ainda conquistou o European Conference on Computer Vision (ECCV) *Best Demo Award* de 2016 e o artigo oral do *Computer Vision and Pattern Recognition* (CVPR) de 2017.

3.4.2 *Inception, Inception-ResNet Pytorch Implementation*

Esta é uma implementação em *Pytorch* de *Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning* [12] que apresenta dois modelos, um modelo *Inception-v4* e um modelo *Inception-ResNet-v2*.

Este explora como é que ao combinar a arquitetura *Inception* com as ligações residuais, características de Redes Residuais (*Resnets*), poderemos alcançar maiores avanços no desempenho do reconhecimento de imagens, com

um custo computacional relativamente baixo. São obtidos indícios de que este tipo de treino permite acelerar significativamente o treino das redes *Inception*, superando até mesmo, numa pequena margem, as redes *Inception* igualmente dispendiosas e sem ligações residuais. Além disto, fornece ainda uma maior estabilização no treino de redes residuais de *Inception* muito amplas.

Esta implementação atingiu 3,08% de erro top-5 no conjunto de teste do desafio de classificação *ImageNet* (CLS).

3.5 Arquitetura do Sistema

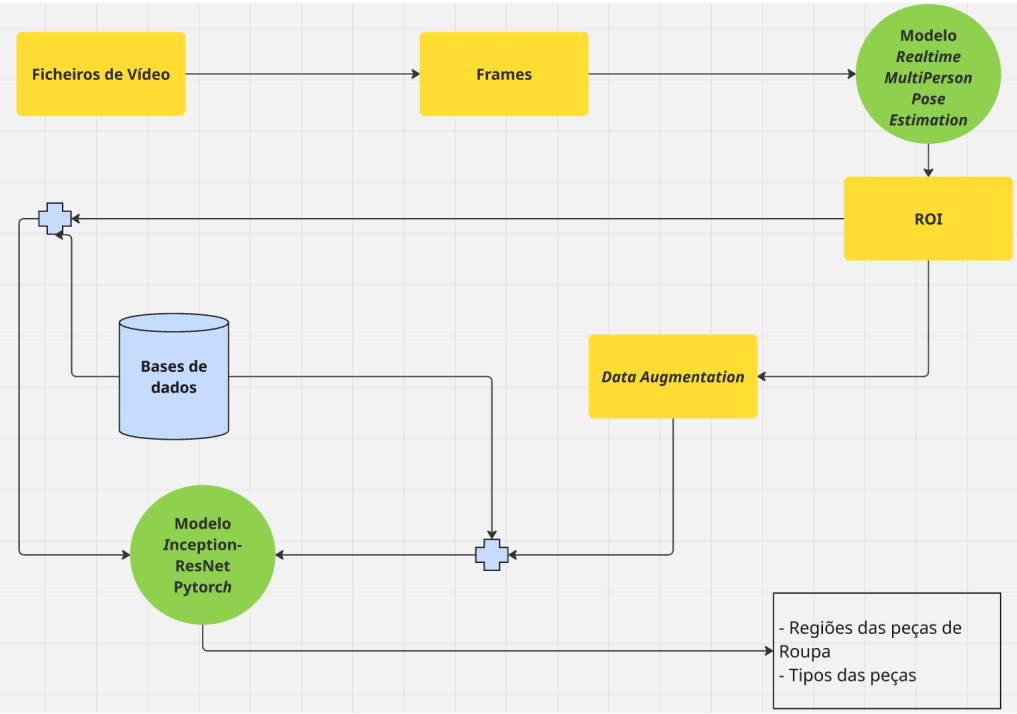

Figura 3.1: Arquitetura do Sistema

A partir da figura 3.1 é possível observar a arquitetura que suporta o funcionamento do sistema desenvolvido neste projeto. Em primeiro lugar, os ficheiros de vídeo são convertidos em *frames* que são introduzidas no modelo *Realtime MultiPerson Pose Estimation*, de modo a detetar as coordenadas dos pontos-chave dos esqueletos do corpo humano nos indivíduos representados nas imagens.

De seguida, as coordenadas desses pontos-chaves são utilizados para criar ROIs, em cada uma das imagens, representando a informação essencial ao modelo que se pretende implementar.

No caso dos modelos treinados neste projeto, são criados dois conjuntos de dados diferentes, um com ROIs relativos à parte de baixo do esqueleto , partindo da cintura, e outro com ROIs relativos à parte de cima do esqueleto.

Estes conjuntos de dados são divididos com base nas suas classes rotuladas e divididos em dados de treino, validação e teste, de forma a serem posteriormente enviados para o modelo *Inception-ResNet Pytorch Implementation*, onde os modelos são treinados, a partir destes dados, para detetar os padrões e onde são ainda avaliados, devolvendo uma previsão de classificação do tipo de peça de roupa utilizado pelo indivíduo representado em cada uma das imagens.

Neste projeto, o objetivo final implica o treino de quatro modelos diferentes:

1. ***Lower Body Clothes*:** classificação de peças de roupa da parte de baixo do corpo, com aplicação de *Data Augmentation* nos dados de treino.
2. ***Upper Body Clothes*:** classificação de peças de roupa da parte de cima do corpo, com aplicação de *Data Augmentation* nos dados de treino.
3. ***Lower Body Clothes With Unbalanced Classes*:** classificação de peças de roupa da parte de baixo do corpo, sem aplicação de *Data Augmentation* nos dados de treino.
4. ***Upper Body Clothes With Unbalanced Classes*:** classificação de peças de roupa da parte de cima do corpo, sem aplicação de *Data Augmentation* nos dados de treino.

Aos primeiros dois modelos são aplicadas técnicas de *Data Augmentation*, enumeradas anteriormente na secção 3.3.1, de modo a expandir o conjunto de dados de treino e a que todas as classes contenham o mesmo número de de imagens.

Os dois modelos seguintes são treinados a partir dos mesmos conjuntos de dados, no entanto, sem a utilização de técnicas de *Data Augmentation*. Assim, as classes deixam de se encontrar balanceadas, o que se verifica como sendo uma valiosa oportunidade de comparar o desempenho destes modelos em relação à capacidade de classificação com os respetivos modelos mencionados anteriormente. A tomada desta abordagem tem como principal objetivo, analisar de que forma a utilização de técnicas como o *Data Augmentation*, permite influenciar o desempenho final dos modelos treinados na tarefa de classificação.

3.6 Conclusões

A análise das decisões tomadas ao longo deste projeto, na seleção das ferramentas, técnicas e métodos aplicados, juntamente com a descrição da arquitetura do sistema implementado, permite definir a base tecnológica sobre a qual o projeto foi construído.

A interação conjunta entre todos estes componentes serve de suporte para os resultados obtidos e consequentemente para as análises que serão conduzidas nos capítulos seguintes.

Capítulo

4

Resultados Experimentais

4.1 Introdução

Neste capítulo, são apresentados, para cada um dos modelos criados ao longo deste projeto, as respetivas análises objetivas e subjetivas.

As análises objetivas ou quantitativas baseiam-se nas métricas obtidas, como *accuracy*, perda, precisão, entre outras, permitindo a elaboração de uma avaliação rigorosa do desempenho destes modelos. A análise subjetiva ou qualitativa já diz respeito a uma avaliação baseada especialmente num apreciação visual e interpretação dos resultados obtidos.

Assim sendo, esta fase permite adquirir uma avaliação mais abrangente e fundamentada dos modelos criados, através da identificação das suas limitações práticas, bem como na perceção da sua aplicação em cenários reais.

4.2 Análise Objetiva

4.2.1 Modelo *Realtime MultiPerson Pose Estimation*

O modelo *rtpose* de HPE foi previamente treinado e avaliado a partir do conjunto de dados *COCO* 2017. Na tabela 4.1 encontram-se representados os parâmetros de treino do modelo e na tabela 4.2, é possível observar os resultados obtidos na avaliação do mesmo.

Model Name	**mAP**	***Inference Time***
[original rtpose]	0.653	-

Tabela 4.2: Resultados da avaliação do modelo *rtpose*

Tabela 4.1: Hiperparâmetros de treino utilizados

Hiperparâmetro	**Valor**
Épocas totais	75
Épocas com pesos congelados	5
Learning Rate inicial	1.0
Learning Rate (pré-treinamento)	1×10^{-4}
Momentum	0.9
Weight Decay	0.0
Batch size	4
Tamanho da imagem	368
Otimizador	SGD com Nesterov
Scheduler	ReduceLROnPlateau (factor 0.8, patience 5)
Dispositivo	GPU

4.2.2 Modelos *Inception-ResNet Pytorch Implementation*

Cada um dos modelos é treinado com base nos seguintes parâmetros:

Parâmetro	**Valor**
Modelo	Inception-ResNet-v2
Encoder auxiliar	ResNet34
Épocas	10
Otimizador	Adam
Learning rate	0.0001
Função de perda	CrossEntropyLoss

Tabela 4.3: Principais parâmetros de treino

Os conjuntos de dados utilizados contém informação acerca dos atributos e *labels* correspondentes, relativos às peças de roupa inferiores, *Lower Body Clothes*, e às peças de roupa superiores, *Upper Body Clothes*. A partir da tabela 4.4 podem ser observados os valores específicos atribuídos a cada um destes atributos, tal como imagens de exemplos para cada uma destas *labels*, sendo que para as peças de roupa correspondentes às *Upper Body Clothes* não foram encontrados exemplos rotulados com as *labels* 4, 5, 6, 7 e 8.

A avaliação da *accuracy* foi conduzida com base em dois critérios distintos, fundamentados na correspondência entre os rótulos previstos e os rótulos reais:

- **Top-1:** considera-se correta a previsão em que a classe com maior probabilidade atribuída pelo modelo coincide exatamente com o rótulo

Figura 4.1: Exemplos de imagens *Lower Body Clothes* rotuladas com a *Label* 0 - *Jeans*

Figura 4.2: Exemplos de imagens *Lower Body Clothes* rotuladas com a *Label* 1 - *Leggings*

Figura 4.3: Exemplos de imagens *Lower Body Clothes* rotuladas com a *Label* 2 - *Pants*

Figura 4.4: Exemplos de imagens *Lower Body Clothes* rotuladas com a *Label* 3 - *Shorts*

Figura 4.5: Exemplos de imagens *Lower Body Clothes* rotuladas com a *Label* 4 - *Skirt*

Figura 4.6: Exemplo de imagem *Lower Body Clothes* rotulada com a *Label* 5 - *Dress*

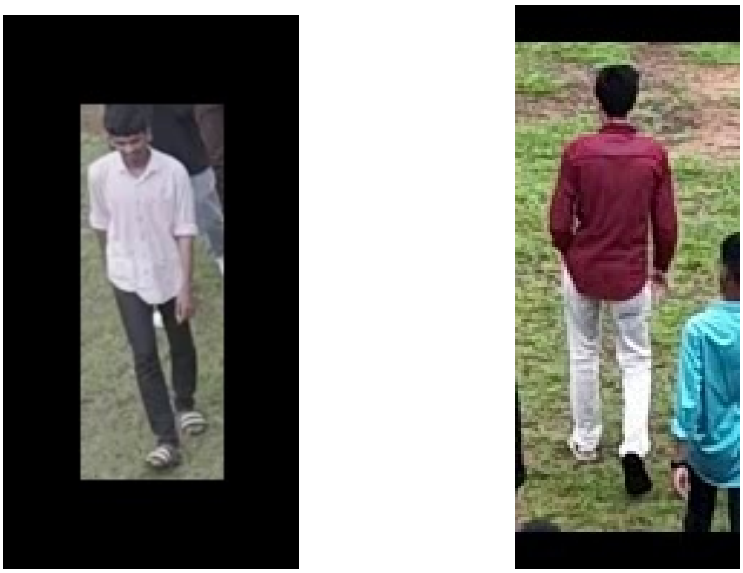

Figura 4.7: Exemplos de imagens *Lower Body Clothes* rotuladas com a *Label* 6 - *Uniform*

Figura 4.8: Exemplos de imagens *Lower Body Clothes* rotuladas com a *Label* 7 - *Suit*

Figura 4.9: Exemplos de imagens *Lower Body Clothes* rotuladas com a *Label* -1 - *Unknown*

Figura 4.10: Exemplos de imagens *Upper Body Clothes* rotuladas com a *Label* 0 - *T-Shirt*

Figura 4.11: Exemplos de imagens *Upper Body Clothes* rotuladas com a *Label* 1 - *Blouse*

Figura 4.12: Exemplos de imagens *Upper Body Clothes* rotuladas com a *Label* 2 - *Sweater*

Figura 4.13: Exemplos de imagens *Upper Body Clothes* rotuladas com a *Label* 3 - *Coat*

Figura 4.14: Exemplos de imagens *Upper Body Clothes* rotuladas com a *Label* -1 - *Unknown*

Label	**Descrição (*Lower Body CLothes)***	**Descrição (*Upper Body CLothes)***
0	*Jeans*	*TShirt*
1	*Leggings*	*Blouse*
2	*Pants*	*Sweater*
3	*Shorts*	*Coat*
4	*Skirt*	*Dress*
5	*Dress*	*Uniform*
6	*Uniform*	*Shirt*
7	*Suit*	*Suit*
8	-	*Hoodie*
-1	*Unknown*	*Unknown*

Tabela 4.4: Classes, *labels* e respetivas descrições

verdadeiro;

- **Top-5:** considera-se correta a previsão em que o rótulo verdadeiro está entre as cinco classes mais prováveis indicadas pelo modelo.

4.2.2.1 Modelo *Lower Body Clothes*

Classe	**Amostras**	**Top-1 Acc**	**Top-5 Acc**
-1	84	0.9881	1.0
0	1563	0.9123	1.0
1	529	0.9112	1.0
2	146	0.9795	1.0
4	522	0.8870	1.0
6	1393	0.9325	1.0

Tabela 4.5: Valores de *accuracy* obtidos por classe no Modelo *Lower Body Clothes*

Na tabela 4.5 encontram-se representados os valores de *accuracy* obtidos por classe na avaliação deste modelo. Como podemos observar devido à presença de uma grande descrepância na quantidade de dados de treino nas classes 3, 5 e 7, estas foram eliminadas e não foram incluídas no conjuntos de dados de treino deste modelo.

Top-1 Acc	**Top-5 Acc**
0.92	1.0

Tabela 4.6: Valor de *accuracy* do Modelo *Lower Body Clothes*

Na tabela 4.6 é possível verificar que este modelo obteve um desempenho com uma *accuracy* de 0.92, segundo o critério *Top-1*, e de 1.0, segundo o critério *Top-5*.

4.2.2.2 Modelo *Upper Body Clothes*

Classe	Amostras	Top-1 Acc	Top-5 Acc
-1	4160	0.9661	1.0
0	245	0.4939	1.0
1	119	0.8319	1.0
2	23	0.9130	1.0
3	592	0.9679	1.0

Tabela 4.7: Valores de *accuracy* obtidos por classe no Modelo *Upper Body Clothes*

Na tabela 4.7 encontram-se representados os valores de *accuracy* obtidos por classe na avaliação do Modelo *Upper Body Clothes*. Neste modelo, foram eliminadas as classes 4, 5, 6, 7 e 8 devido à inexistência de imagens rotuladas com essas classes.

Top-1 Acc	Top-5 Acc
0.94	1.0

Tabela 4.8: Valor de *accuracy* do Modelo *Upper Body Clothes*

Na tabela 4.8 é possível verificar que este modelo obteve um desempenho com uma *accuracy* de 0.94, segundo o critério *Top-1*, e de 1.0, segundo o critério *Top-5*.

4.2.2.3 Modelo *Lower Body Clothes With Unbalanced Classes*

Classe	Amostras	Top-1 Acc	Top-5 Acc
-1	84	0.9524	1.0
0	1563	0.8503	1.0
1	529	0.9282	1.0
2	146	0.9589	1.0
4	522	0.8448	0.9981
6	1393	0.9253	1.0

Tabela 4.9: Valores de *accuracy* obtidos por classe no Modelo *Lower Body Clothes With Unbalanced Classes*

Na tabela 4.9 encontram-se representados os valores de *accuracy* obtidos por classe na avaliação do Modelo *Lower Body Clothes With Unbalanced Classes*.

Top-1 Acc	Top-5 Acc
0.89	0.99

Tabela 4.10: Valor de *accuracy* do Modelo *Lower Body Clothes With Unbalanced Classes*

Na tabela 4.10 é possível observar que este modelo obteve um desempenho com uma *accuracy* de 0.89, segundo o critério *Top-1*, e de 0.99, segundo o critério *Top-5*. Em comparação com os valores obtidos na avaliação do Modelo *Lower Body Clothes*, descrito na secção 4.2.2.1, verifica-se que este modelo apresenta um desempenho inferior. Deste modo, é possível concluir que a utilização de técnicas de *data augmentation* têm um impacto positivo no desempenho final do classificador.

4.2.2.4 Modelo *Upper Body Clothes With Unbalanced Classes*

Classe	Amostras	Top-1 Acc	Top-5 Acc
-1	4160	0.9502	1.0
0	245	0.7102	1.0
1	119	0.8824	1.0
2	23	1.0000	1.0
3	592	0.8632	1.0

Tabela 4.11: Valores de *accuracy* obtidos por classe no Modelo *Upper Body Clothes With Unbalanced Classes*

Na tabela 4.11 encontram-se representados os valores de *accuracy* obtidos por classe na avaliação do Modelo *Upper Body Clothes With Unbalanced Classes*.

Top-1 Acc	Top-5 Acc
0.93	1.0

Tabela 4.12: Valor de *accuracy* do Modelo *Upper Body Clothes With Unbalanced Classes*

Na tabela 4.12 é possível verificar que este modelo obteve um desempenho com uma *accuracy* de 0.93, segundo o critério *Top-1*, e de 1.0, segundo o critério *Top-5*. Com base no desempenho obtido neste modelo em relação ao Modelo *Upper Body Clothes*, descrito na secção 4.2.2.2, em que se notou novamente uma redução desse mesmo valor, é possível mais uma vez comprovar o impacto positivo provocado quando aplicadas técnicas de *data augmentation* no conjunto de dados de treino.

4.3 Analise Subjetiva

4.3.1 Modelo *Realtime MultiPerson Pose Estimation*

Após o envio, para o Modelo de *Realtime MultiPerson Pose Estimation*, do conjuntos de dados composto pelas ROIs criadas anteriormente, foram apontadas algumas situações de falha na sua capacidade de deteção.

Dentro destas situações, destacam-se maioritariamente cenários em que o modelo não detetou pontos-chave representados nas imagens ou em que o modelo deteta incorretamente a sua posição.

A ocorrência deste tipo de situações pode estar associada a várias causas. Geralmente estas são as principais razões apontadas:

1. **Características Visuais da Imagem:** características como distância, resolução, ângulos, perspetivas, iluminação, entre outras, podem influenciar o desempenho do modelo.

2. **Ambiguidade Visual:** fundos complexos ou padrões que se assemelhem a partes do corpo;

3. **Oclusões parciais ou totais:** quando partes do corpo estão tapadas, o modelo pode inventar pontos-chave onde julga provável;

4. **Poses incomuns:** o modelo é treinado com poses típicas e pode falhar com poses estranhas em relação às do seu treino.

Figura 4.15: Pontos-chave não detetados

Figura 4.16: Pontos-chave detetados incorretamente

As figuras 4.15, onde não são detetados todos os pontos-chave representados na imagem, e 4.16, onde é possível constatar que os pontos-chaves relativos à parte inferior do corpo humano foram detetados incorretamente, são dois exemplos de cenários de falha na deteção do modelo. Uma possível justificação para esta falha pode residir, por exemplo, em relação à posição dos indivíduos na imagem, nomeadamente quando estes se encontram de costas, o que pode dificultar o processo de deteção de pontos-chave como joelhos, tornozelos, entre outros.

4.3.2 Modelos *Inception-ResNet Pytorch Implementation*

Durante a fase de avaliação destes modelos, foram identificadas situações em que este não conseguiu realizar uma classificação correta. A partir da análise visual e da interpretação dos casos de insucesso, foi possível identificar potenciais fatores que poderão estar na origem dessas falhas de classificação.

4.3.2.1 Modelo *Lower Body Clothes*

A partir dos resultados obtidos na avaliação deste modelo, entre os diversos fatores identificados na tentativa de justificar estas falhas, destacam-se:

1. **Vestuários em tons escuros:** uma característica comum em diversos casos consiste na dificuldade de classificação de peças de vestuário em que predominam os tons mais escuros, nomeadamente o preto, como é o caso da peça de roupa representada na figura 4.17, cuja classe verdadeira era 0 (*Jeans*), no entanto, foi classificada como -1 (*Unknown*).

Figura 4.17: Vestuário em tons escuros

2. **Confusão entre classes:** Como é possível observar a partir da matriz de confusão 4.26, existe uma dificuldade na classificação entre a classe 0, correspondente a *Jeans*, e a classe 6, correspondente a *Uniforms*. Isto pode verificar-se devido ao facto de o rótulo Uniforme ser muito genérico, podendo incluir camisas, casacos, calças, entre outras, o que pode influenciar a falha de classificação entre estas classes em cenários como o das imagens 4.18 e 4.19, seja devido à similaridade das calças do uniforme às calças comuns, seja devido à falta de conteúdo relativo às restantes parte do uniforme na imagem.

3. **Pontos-chave não detetados ou detetados na posição incorreta:** isto resultou, subsequentemente, na geração inadequada ROIs nas imagens. Em muitos destes casos, as ROIs não incluíam a representação correspondente à secção inferior do esqueleto, a área de maior relevância para o modelo em análise, comprometendo assim a eficácia da classificação. Exemplos da ocorrência deste tipo de situações são elaborados de forma mais aprofundada na subsecção 4.3.1.

Figura 4.18: Previsão de classe 6 em vez de 0 (ROI)

Figura 4.19: Previsão de classe 6 em vez de 0 (Frame completa)

4. **Pontos-chave insuficientes:** A existência de imagens em que foi detetado um número reduzido de pontos-chave, poderia levar à criação de ROIs que não representassem informação suficiente para que o modelo fornecesse uma classificação precisa da classe correta.

   No cenário representado nas imagens 4.20 e 4.21, como na imagem original foram detetados os pontos-chaves relativos à anca direita e à anca esquerda, a ROI irá ser criada com base apenas nesses dois pontos. Assim sendo, a ROI não irá representar informação suficiente correspondente à parte inferior do esqueleto para que o modelo posso efetuar uma classificação correta.

5. **Múltiplos indivíduos:** Em imagens que representam mais do que uma pessoa, nalguns casos, o indivíduo identificado para classificação não corresponde ao individuo real cujo vestuário está etiquetado na imagem. Uma forma de tentar contornar esta situação foi, na fase de criação das ROIs, realizar uma tentativa de identificação da pessoa principal na imagem, no entanto, é possível observar que nalgumas imagens, este processo não se verificou como sendo suficiente ou poderá não ter sido identificada a pessoa principal correta.

   A imagem 4.23, onde podem ser observadas a representação de duas pessoas, encontra-se rotulada com a *label* 1 - *Leggins*, correspondente

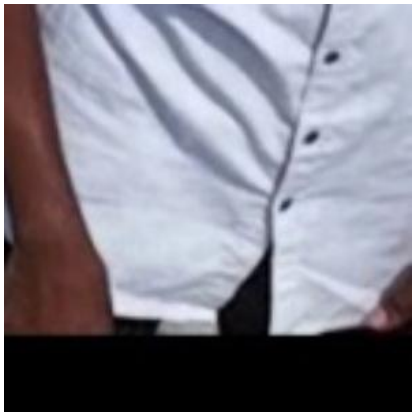

Figura 4.20: Número insuficiente de pontos-chave (ROI)

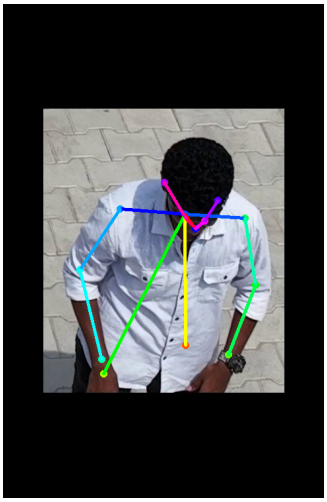

Figura 4.21: Número insuficiente de pontos-chave (Esqueleto)

à peça de vestuário do indivíduo da direita. No entanto, o indivíduo identificado pelo modelo, na imagem 4.22, para a classificação foi o da esquerda, o que levou o modelo a classificar a peça de roupa em análise como sendo classe 4 - *Skirt*.

Figura 4.22: Classificação no indivíduo incorreto (ROI)

Figura 4.23: Classificação no indivíduo incorreto (Original)

6. **Origem cultural do vestuário:** Com base na tabela 4.5, torna-se possível identificar a classe em que foram obtidos valores de precisão mais reduzidos, neste caso, a classe 4, correspondente à classe *Skirt*. Esta questão pode ter sido influenciada pelo facto de em muitos exemplos em que a classificação deveria ser *Skirt*, verificou-se que os indivíduos são do sexo feminino e de origem do Sul da Ásia e Ásia Central. Uma prática comum em trajes tradicionais com esta origem cultural, como o *Salwar Kameez* ou o que se encontra representado nas figuras 4.24 e 4.25, implica o uso de calças por baixo das saias e dos vestidos. Esta sobreposição de peças pode contribuir para a incorreta classificação da classe nestes exemplos, que são confundidos maioritariamente, por exemplos da classe 6, *Uniforms*.

Figura 4.24: Vestuário de origem cultural asiática (ROI)

Figura 4.25: Vestuário de origem cultural asiática (Original)

4.3.2.2 Modelo *Upper Body Clothes*

Da mesma forma que no modelo destacado anteriormente foram identificadas situações em que este não conseguiu realizar uma classificação correta, foram também identificadas, neste modelo, possíveis fatores que pudessem justificar a ocorrência deste tipo de situações.

À semelhança do anterior, estas também poderiam ser originadas a partir da incorreta deteção das coordenadas dos pontos-chave do esqueleto, tal como na insuficiência da quantidade de pontos-chaves detetados e na identificação do indivíduo incorreto nas imagens. No entanto são identificados ainda potenciais fatores como:

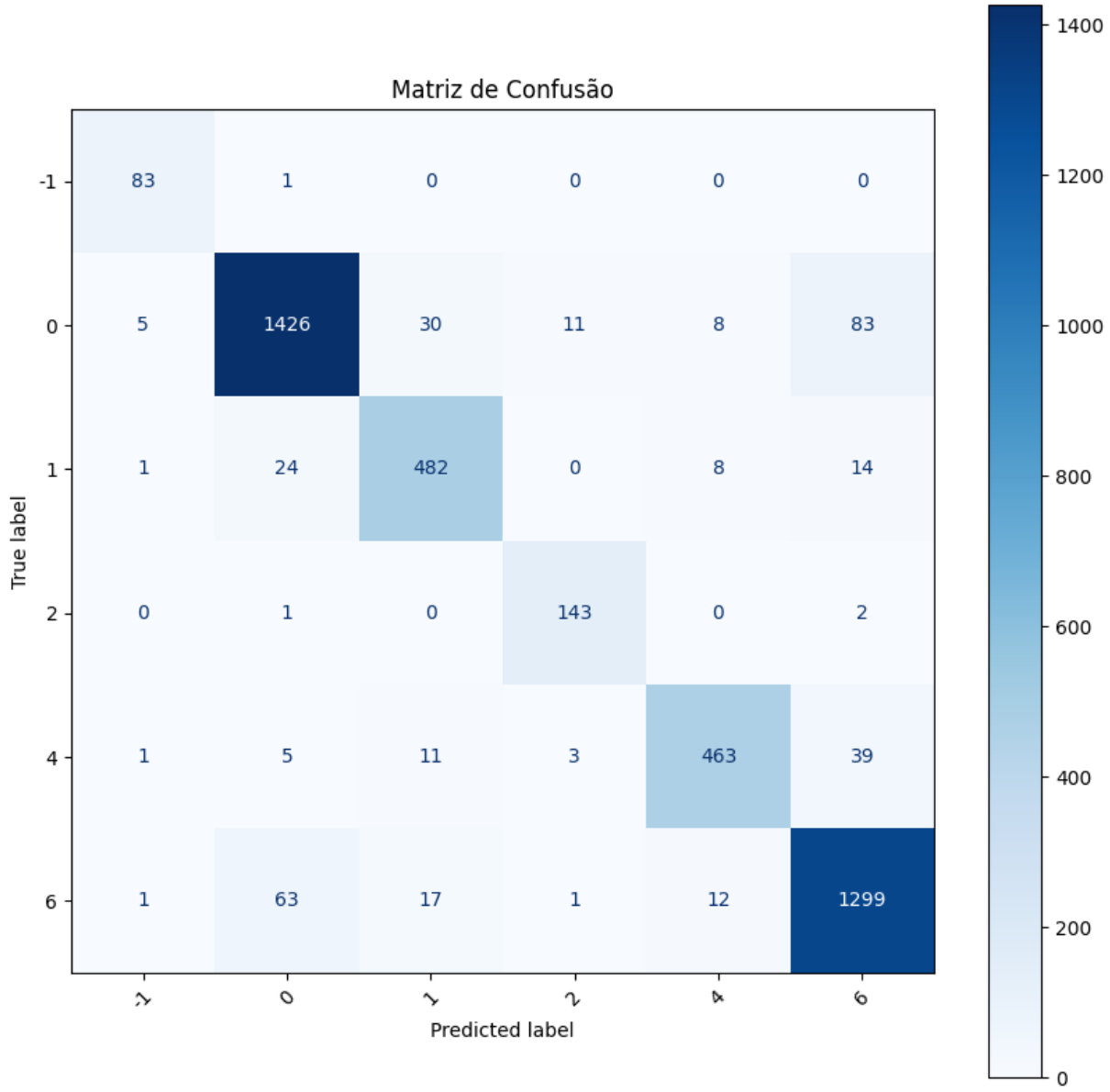

Figura 4.26: Matriz de confusão do Modelo *Lower Body Clothes*

1. **Classificação de imagens da classe 0:** Na matriz de confusão 4.30, é possível observar uma dificuldade de classificação de peças correspondentes à classe 0 de *T-Shirts*, que apresenta o valor mais reduzido de precisão na classificação das imagens em relação aos restantes valores apresentados na tabela 4.7. Na maior parte dos casos ele acaba por classificar estas peças, como a exibida no cenário da imagem 4.27, com a *label* -1 *Unknow*, ou seja, desconhecido. Isto poderá dever-se ao facto de nos dados de treino do modelo, as imagens serem maioritariamente da classe -1, o que levou a que nas outras classes o processo de *data augmentation* tivesse que ser mais intenso para que estas atingissem o mesmo número de imagens da classe -1.

Figura 4.27: Classe 0 prevista como classe -1

2. **Classificação de imagens da classe -1:** A partir da matriz, verifica-se ainda uma tendência para classificar incorretamente imagens etiquetadas com a *label* -1, correspondente a *Unknown*, classificando-as na grande maioria dos casos, como sendo classe 1 - *Blouse* ou classe 3 - *Coat*.

   Imagens rotuladas com a classe -1, *Unkown*, ou seja, desconhecido, apresentam uma tendência para exibir peças de vestuário com características mais ambíguas que dificultam a sua correta distinção.

   A tendência paara o modelo classificar estas imagens incorretamente pode dever-se ao facto de estas apresentarem certos padrões e características genéricas correspondentes a outras classes, como é o caso da imagem 4.28, em que é possível observar uma peça de roupa com mangas e botões, levando o modelo a classificá-la como *Blouse*, Blusa. O mesmo se verifica na figura 4.29, em que se encontra representada uma

peça com características que se assemelham a um casaco, contribuindo para que a previão do modelo seja *Coat*.

Figura 4.28: Classe -1 prevista como classe 1

Figura 4.29: Classe -1 prevista como classe 3

4.3.2.3 Modelo *Lower Body Clothes With Unbalanced Classes*

A decisão de treinar este modelo surgiu, sobretudo, com o objetivo de investigar de que forma o desempenho do classificador poderia ser influenciado pela aplicação de técnicas de *data augmentation*. Esta abordagem teve como propósito avaliar se a capacidade de classificação apresentaria melhorias em comparação com o modelo *Lower Body Clothes*, descrito na Secção 4.3.2.1.

Contudo, observou-se um desempenho inferior neste modelo, com um aumento na frequência dos casos de falha. Apesar disso, as principais situações que originam essas falhas, bem como os fatores que as justificam, mantêm-se semelhantes aos identificados no modelo Lower Body Clothes, sendo agora apenas mais recorrentes.

4.3.2.4 Modelo *Upper Body Clothes With Unbalanced Classes*

Da mesma forma que se mantém os principais cenários e potenciais fatores que justifiquem a sua ocorrência, quando exploramos os resultados obtidos nos modelos *Lower Body Clothes* e *Lower Body Clothes With Unbalanced Classes*, o mesmo se verifica entre os modelos *Upper Body Clothes* e *Upper Body Clothes With Unbalanced Classes*, no entanto, manifestando-se também em maior frequência.

Uma vez mais, o treino deste modelo surgiu do objetivo de explorar de que forma o desempenho entre estes dois modelos poderia ser influenciada pela

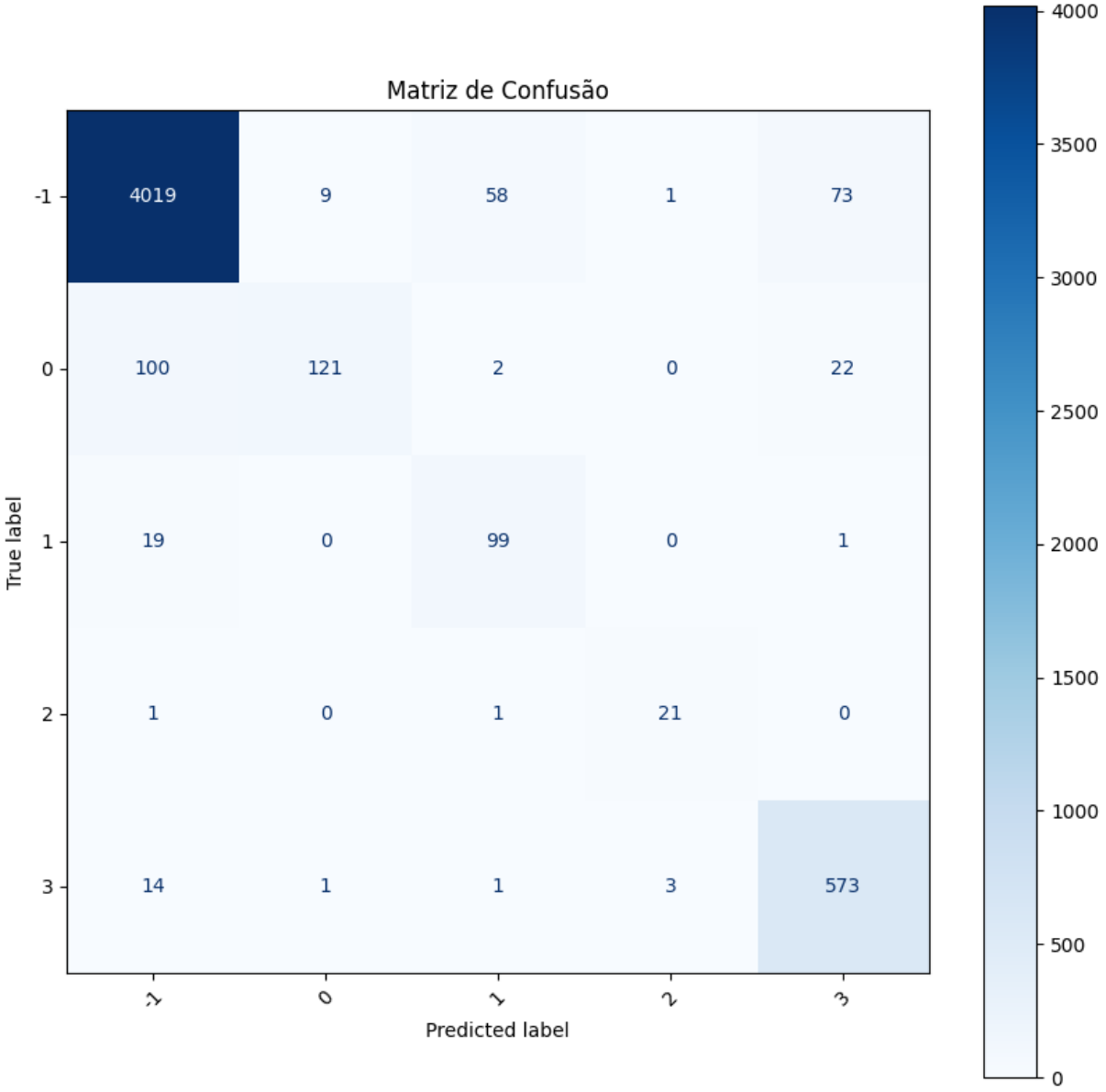

Figura 4.30: Matriz de confusão do Modelo *Upper Body Clothes*

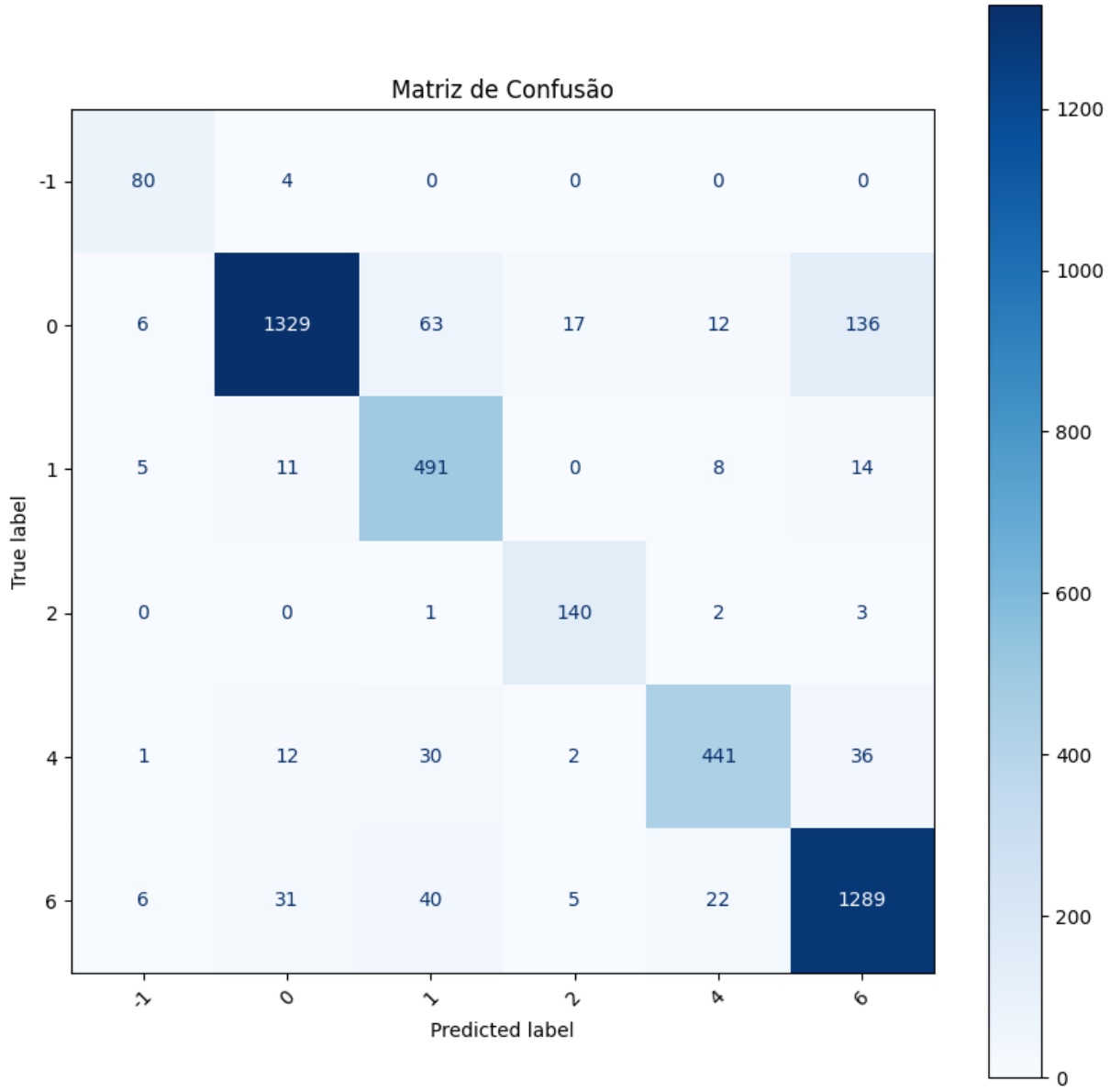

Figura 4.31: Matriz de confusão do Modelo *Lower Body Clothes With Unbalanced Classes*

utilização de técnicas de *data augmentation*, que demonstrou novamente, que a falta da aplicação desta técnica, resultou num desempenho inferior em relação ao *Upper Body Clothes*.

4.4 Conclusões

A análise combinada dos resultados objetivos e subjetivos possibilitou uma avaliação do comportamento dos modelos treinados baseada especialmente no seu desempenho final, destacando as suas capacidades e fragilidades em diferentes situações.

Este capítulo revelou-se determinante no processo de tomada de decisões relativas à implementação de alterações direcionadas à melhoria da capacidade de classificação dos modelos e à minimização de situações de falha, contribuindo assim para o reforço da competência e fiabilidade da solução desenvolvida.

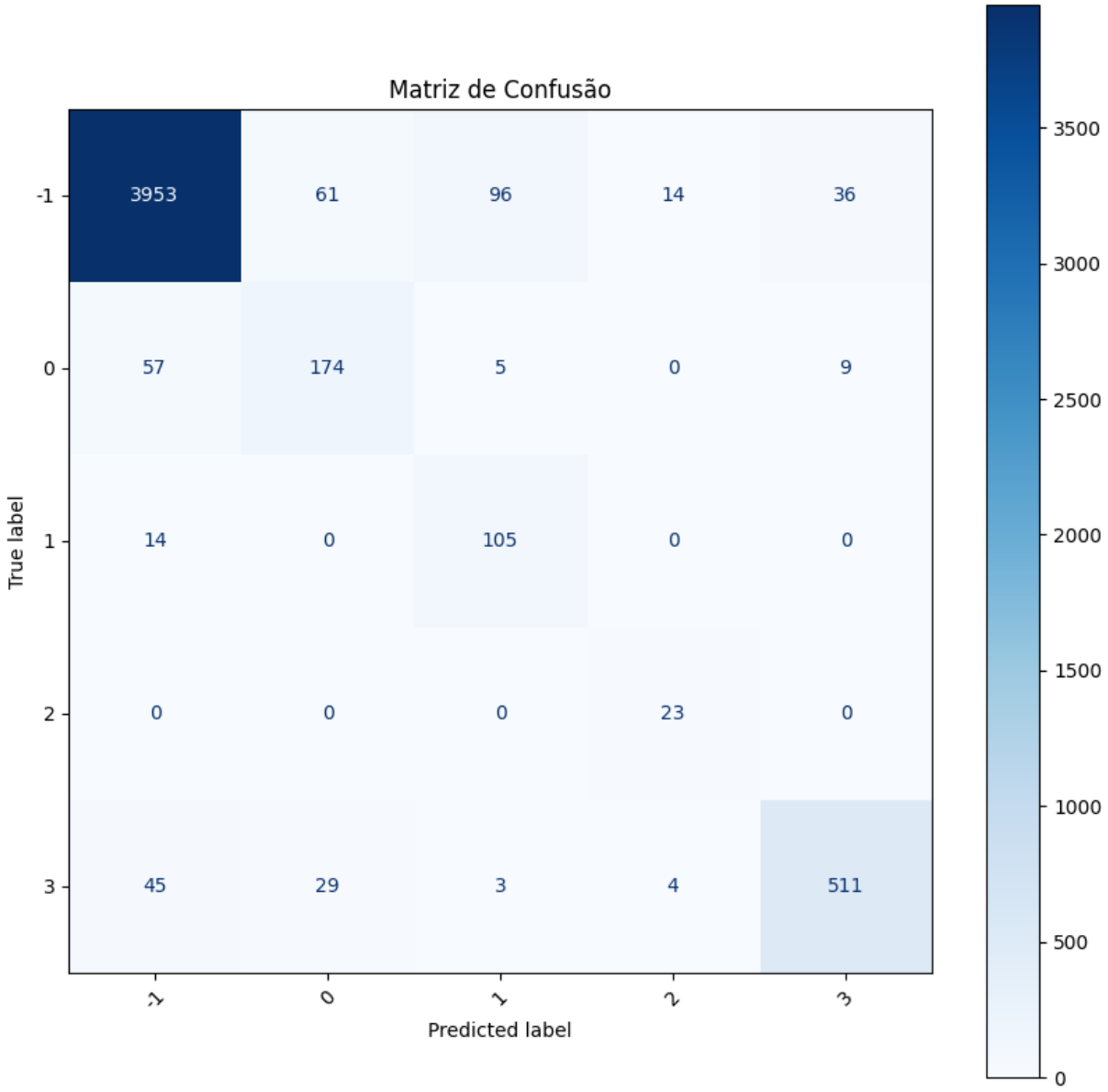

Figura 4.32: Matriz de confusão do Modelo *Upper Body Clothes With Unbalanced Classes*

Capítulo

5

Conclusões e Trabalho Futuro

5.1 Conclusões Principais

Este trabalho apresenta o processo de desenvolvimento, implementação e avaliação de modelos de aprendizagem profunda multimodal para a tarefa de classificação de peças de roupa em indivíduos representados em imagens.

A partir dos modelos criados foram ainda exploradas algumas abordagens que pudessem contribuir para que estes alcançassem uma melhor capacidade de classificação. A exploração destas abordagens resultaram na implementação de novos de modelos, com o intuito de analisar, com base nos resultados obtidos depois de avaliados, de que modo o desempenho destes classificadores era afetado de acordo com as decisões tomadas.

O processo de desenvolvimento permitiu adquirir um conhecimento mais aprofundado acerca das diferentes etapas envolvidas na construção dos modelos e os resultados obtidos demonstram viabilidade e eficácia nas abordagens adotadas.

Para concluir, os objetivos definidos inicialmente foram alcançados e constituem uma base promissora para o desenvolvimento de trabalhos futuros.

5.2 Trabalho Futuro

A partir dos resultados obtidos e dos modelos desenvolvidos ao longo do projeto, existem ainda várias possibilidades de futuras implementações e uma grande variedade de potenciais aplicações para estes modelos em diferentes cenários.

Exemplos de possibilidades, das quais algumas já foram identificadas na fase inicial do desenvolvimento do projeto, incluem:

1. **Expansão para Acessórios e Calçado:** Embora o foco deste projeto tenha sido a parte superior e inferior do corpo, poderiam ainda ser implementados modelos para a classificação de acessórios como chapéus, malas, óculos, entre outros, ou até modelos para classificação de calçado;

2. **Segurança e Vigilância:** A classificação do vestuário de um indivíduo pode verificar-se útil em situações que envolvem segurança e vigilância, como por exemplo, na identificação de pessoas em ambientes restritos;

3. **Análise de tendências de moda:** A partir de peças de roupa identificadas em imagens de, por exemplo, redes sociais, pode ser elaborada uma análise de tendências de moda com base em diferentes regiões, faixas etárias, género, estações, entre outras;

4. **Recomendação automática de roupa:** Estes modelos podem ser integrados em plataformas de comércio online como sistemas de recomendação personalizado, de modo a sugerir itens semelhantes ou complementares aos identificados nas imagens com base no histórico de compras ou de visualização;

5. **Assistência para pessoas com deficiência visual:** Pode representar um mecanismo de suporte a pessoas com deficiência visual, permitindo identificar a roupa que estão a usar, escolher ou até em possíveis combinações entre peças.

Bibliografia

[1] Nilesh Barla. A Comprehensive Guide to Human Pose Estimation, 2021. [Online] https://www.v7labs.com/blog/human-pose-estimation-guide.

[2] Data Science Academy. Capítulo 4 – O Neurônio, Biológico e Matemático, 2025. [Online] https://www.deeplearningbook.com.br/o-neuronio-biologico-e-matematico/.

[3] Nei Grando. Neurônios e Redes Neurais Artificiais, 2022. [Online] https://neigrando.com/2022/03/03/neuronios-e-redes-neurais-artificiais/.

[4] Sienna Roberts. Artificial Intelligence Techniques: Explained in Detail, 2025. [Online] https://www.theknowledgeacademy.com/blog/artificial-intelligence-techniques/.

[5] IBM. What is computer vision?, 2021. [Online] https://www.ibm.com/think/topics/computer-vision.

[6] GeeksForGeeks. What is a Neural Network?, 2025. [Online] https://www.geeksforgeeks.org/neural-networks-a-beginners-guide/.

[7] Daniil Fedorov. Ambiente virtual do Python 3 no Ubuntu 22.04, 2023. [Online] https://serverspace.com.br/support/help/python-3-virtual-environment-on-ubuntu-22-04/.

[8] birdhouses. A Comprehensive Guide To Data Augmentation Techniques In Pytorch, 2024. [Online] https://peerdh.com/blogs/programming-insights/a-comprehensive-guide-to-data-augmentation-techniques-in-pytorch.

[9] Scaler Topics. Data Augmentations in Pytorch, 2023. [Online] https://www.scaler.com/topics/pytorch/data-augmentations-in-python/.

[10] Tracker. O que é : Region of Interest, 2024.

[11] Shih-En Wei Yaser Sheikh Zhe Cao, Tomas Simon. Realtime Multi-Person Pose Estimation, 2016. [Online] https://github.com/ZheC/Realtime_Multi-Person_Pose_Estimation?tab=readme-ov-file.

[12] Vincent Vanhoucke Alex Alemi Christian Szegedy, Sergey Ioffe. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning, 2016. [Online] https://github.com/zhulf0804/Inceptionv4_and_Inception-ResNetv2.PyTorch.