UNIVERSITÀ DEGLI STUDI DI SALERNO

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE ED
ELETTRICA E MATEMATICA APPLICATA

CORSO DI LAUREA MAGISTRALE IN INGEGNERIA INFORMATICA

ELABORATO FINALE

Adapting Vision Language Models via parameter-efficient fine-tuning for Multitask Classification of Age, Gender, and Emotion

Relatore

Prof. Mario Vento

Prof. Antonio Greco

Candidato

Antonio Sessa

Matr. 0622702305

Anno Accademico 2024-2025

Abstract

Description of the problem addressed

Recognizing facial attributes such as age, gender, and emotion is an inherently difficult computer vision task due to high intra-class variability and challenging real-world conditions. In this field, large-scale Vision Language Models (VLM) can offer a powerful, generalized visual representations from large scale image-text pre-training, but their direct application to this specialized domain can be inefficient, due to the unnecessary computational overhead of their full architectures, which are not optimized for a pure classification objective. Therefore the central challenge of this work is to develop an efficient and effective adaptation framework to leverage these powerful, pre-trained vision encoders for a unified, multi-task classification objective.

Thesis framework in the contemporary technical scenario

In the current landscape of computer vision, Vision Language Models represent a major shift in the field, demonstrating exceptional zero-shot capabilities through massive-scale pre-training on billions of image-text pairs, in fact, state-of-the-art models like CLIP, SigLIP, and the recent Perception Encoders have shown that joint vision-language training yields powerful, transferable visual representations. Concurrently, the field has witnessed the rise of Parameter-Efficient Fine-Tuning techniques, particularly LoRA and its variants, which enable adaptation of this large pre-trained models with minimal trainable parameters and computational cost. This thesis positions itself at the intersection of these two trends, proposing a framework that leverages the rich visual representations learned by state-of-the-art VLMs while employing PEFT methodologies to enable efficient, specialized adaptation for multi-task facial analysis to address both the performance and efficiency demands of real-world scenarios.

Personal contribution of the candidate to the solution of the problem described

This thesis contributes a comprehensive framework for the efficient multi-task adaptation of a VLM's vision encoder, encompassing its design, implementation, and rigorous evaluation, with a systematic comparison of multiple adaptation techniques such as linear probing, attention probing, partial fine-tuning, and Parameter-Efficient Fine-Tuning (PEFT). The final result of this work is a unified multi-task model that achieves strong accuracy and generalization across all three facial analysis tasks, while also being computationally efficient by discarding the VLM's text encoder to halve inference GFLOPs.

Description of the experimental contents of the work

The experimental work provides a rigorous empirical evaluation of the proposed framework. The PE-Core-L vision encoder is adapted for the three tasks using a composite dataset (FairFace, Lagenda, RAF-DB, CelebA-HQ), with generalization tested on unseen benchmarks (UTKFace, VggFace2). A comprehensive comparison is conducted between multiple adaptation strategies: a zero-shot baseline, linear and attention probing, partial fine-tuning of the final blocks, and PEFT (LoRA+/DoRA). These methods are evaluated in both single-task and multi-task settings, with the latter employing uncertainty-weighting to balance the loss functions, masked labeling to handle partially annotated data and balanced sampling to address the unbalanced datasets. Performance is measured by accuracy, balanced accuracy and also by computational efficiency, using GFLOPs and the number of trainable parameters to quantify the final model's efficiency.

Contents

Chapter 1

Introduction

1.1 Background

Recognizing soft biometric attributes from facial images is an increasingly critical research area with concrete applications across numerous sectors: in social robotics, it enables more natural and personalized interactions, in marketing, it allows for refined audience segmentation and dynamic content personalization. Furthermore, it can support behavioral analysis in security and surveillance, and be applied to psychological well-being monitoring in healthcare. Despite its broad utility, facial attribute recognition is inherently difficult, as unlike generic classification, it must contend with high intra-class variability: individuals of the same age, gender, or emotion display significant differences due to genetic, ethnic, and morphological factors. The task is further complicated by environmental variables such as poor lighting, varied head poses, complex facial expressions, and occlusions, which make recognition in uncontrolled scenarios exceptionally challenging. In this context, Vision Language Models present a promising opportunity. These models, pre-trained on massive quantities of image-text pairs from the web, acquire robust semantic representations that facilitate transfer to downstream tasks. For facial attribute analysis, a VLM's ability to leverage this vast pre-trained knowledge may enable competitive performance even with limited data, bypassing the need for extensive, task-specific annotations required by traditional deep neural networks.

1.2 Problem statement

The primary challenge this thesis addresses is the effective adaptation of large-scale, pre-trained vision-language models for the specialized domain of multi-task facial attribute classification. While these models possess powerful, generalized visual encoders from being trained on vast and diverse image-text datasets, their direct application to specific tasks such as recognizing gender, age, and emotion from faces is not straightforward. VLMs with auto-regressive decoder architectures with hundreds of millions of parameters incur unnecessary computational overhead for discriminative objectives. On the other hand, dual-encoder models used for zero-shot classification via cosine similarity often fail to achieve the satisfactory accuracy and robustness required for detailed facial analysis. Given these limitations, this thesis aims to develop an efficient and effective adaptation framework. Rather than using the entire VLM architecture, our approach is to isolate the pre-trained vision encoder, as it is the core component from which all of the model's visual understanding capabilities stem. The fundamental hypothesis is that this encoder, as a result of its vast pre-training, has learned a highly adaptable visual representation, that while not yet specialized for facial attributes, provides a powerful and feature-rich foundation for efficient fine-tuning, with the goal of producing an unified multi-task model for the following three key facial analysis tasks:

1.2.1 Facial Expression Recognition

Facial expression recognition (FER) is the task of detecting human emotions from static images or videos, and it is of particular interest for field such as customer behavior analysis, advertising and sociable robotics. The task is typically formulated as a 6+1 class classification task, with the classes taken from Paul Ekman's work that compiled the "universal" human expressions: wrath (anger), grossness (disgust), fear, joy (happiness), loneliness (sadness), shock (surprise) and the plus one for neutral, as a lack of emotion.

1.2.2 Age group classification

Age estimation from facial images is a computer vision task that involves predicting a person's age or age range from a digital image or video. This task has a wide array of applications, including targeted advertising, access control for age-restricted content, and enhancing human-computer interaction. Considering the inherent uncertainty of the task, as the age labels

themselves are often derived from noisy data, we will formulate the problem as an age range classification task. Given an image of a face, we will classify it into one of nine possible groups: 0-2, 3-9, 10-19, 20-29, 30-39, 40-49, 50-59, 60-69, and 70+. Moreover, many practical applications do not require a precise age but rather a general age category, making classification a suitable and efficient solution.

1.2.3 Gender Recognition

Gender recognition from facial images is a computer vision task aimed at classifying a person's perceived gender based on their facial features. This capability is often used in conjunction with other facial analysis tasks to gather demographic statistics, for applications in areas such as human-computer interaction, retail analytics, targeted advertising, and content personalization. The task is formulated as a binary classification problem, where the model is trained to predict one of two labels: 'male' or 'female'.

1.3 Objectives

The primary objective of this thesis is to design, implement, and evaluate an efficient multi-task learning model for the simultaneous classification of gender, age, and emotion from facial images using state-of-the-art vision encoder pre-trained at a massive scale by aligning billions of image-text pairs. To achieve this goal, this work will:

- **Investigate and compare adaptation strategies:** Systematically evaluate different methods for adapting the vision encoder, ranging from linear probing to parameter-efficient fine-tuning (PEFT) methodologies.
- **Evaluate state-of-the-art vision encoders:** Conduct a literature-based analysis to identify and select the vision encoders best suited for the target multi-task classification problem.
- **Develop a robust and balanced multi-task framework**: design and implement a system for the multi-task classifications problem, explicitly addressing challenges of inter-task imbalance (where emotion is underrepresented in the data), class imbalance (especially for negative emotions and extreme age groups), and management of partially annotated datasets (where each dataset provides labels for only a subset of tasks).

- **Analyze performance and efficiency trade-offs:** Assess the proposed solutions by comparing their predictive performance (using metrics such as accuracy and balanced accuracy) against their computational efficiency (measured by the number of trainable parameters and inference latency).

1.4 Thesis structure

This thesis is organized into the following chapters:

- **Chapter 2 - Literature Review:** Provides a comprehensive background on the core technologies, including Vision Transformers and Vision Language Models (VLMs). It examines the Perception Encoder VLM, reviews state-of-the-art methods for facial attribute analysis, details the Parameter-Efficient Fine-Tuning technique LoRA, and describes the datasets used for training and evaluation.

- **Chapter 3 - Methodology:** Details the experimental design and implementation. This chapter covers the shared training configurations, data augmentation pipelines, and balanced sampling strategies. It then describes the specific adaptation methods evaluated linear probing, attention probing, full fine-tuning, and PEFT and elaborates on the multi-task learning framework designed to jointly train the model on all three tasks.

- **Chapter 4 - Experimental Results:** Presents the empirical findings from the experiments. It defines the evaluation metrics and provides a detailed comparative analysis of each model's performance and efficiency, contrasting single-task and multi-task approaches across the different adaptation strategies.

- **Chapter 5 - Conclusion:** Summarizes the main contributions and findings of this work. It discusses the implications of the results in relation to the initial objectives and proposes potential avenues for future research.

Chapter 2

Literature Review

This chapter reviews the foundational concepts, technologies, and methods central to this thesis. It first covers the foundational architecture, the Vision Transformer, and the Vision Language Model paradigm that leverages it for large-scale pre-training. This is followed by a review of Perception Encoders, the specific, state-of-the-art model chosen for this work. With the tool established, the review addresses the problem domain by surveying current literature on Facial Expression Recognition and Age Estimation. Finally, the chapter details the key adaptation methodology, Parameter-Efficient Fine-Tuning with LoRA, and the Datasets used in our experiments.

2.1 Vision Transformers

Vision Transformers (ViT)[1] are an adaptation of the transformers encoder architecture [2] for computer vision task. The main innovations lie in how the input images are converted from the 2D spatial representation to a 1D embedding sequence called *patch embeddings*: given an input image $x \in \mathbb{R}^{H \times W \times C}$, it gets split into $N = HW/P^2$ patches of dimension (P, P). These N patches of resolution (P, P) get flattened and get mapped to D_{model} dimension with a trainable linear projection. These process can be efficiently done with a single convolution, by setting the kernel dimension equal to the desired patch size P and stride equal to P. In the original ViT article, at this point the (N, D_{model}) gets summed with a 1D learnable positional embedding (differently from the original transformers, where positional information is encoded trough sinusoidal positional encoding), and gets prepended with a $[CLS]$ token. Alternatively, Rotary Position Embedding [3] and attention pooling heads have become a popular [4, 5, 6]

alternative to respectively encode positional information and obtain a single comprehensive embedding.

```
# A convolutional layer is efficiently used in ViTs
# to pass from a 2D image
# to a sequence of patch embeddings
...
class VisionTransformer(nn.Module):

  def __init__(self, d_model=768, patch_size=16, ...):
    ...
    self.d_model = d_model # embedding dimensions of image patches
    self.patch_size = patch_size # height and width of the image have
    to be divisible by the patch_size

    # no overlap between patches, as stride=kernel_size
    self.conv1 = nn.Conv2d(
      in_channels=3,
      out_channels=d_model,
      kernel_size=patch_size,
      stride=patch_size,
      bias=False,
    )
    ...
  def forward(self,x):
    # x is shaped as such: [B, 3, h, w]
    batch, channels, h, w = x.shape

    # Applying the convolutional layer to create patch embeddings
    x = self.conv1(x) # [B, 3, h, w]->[B, d_model, h//ps, w//ps]

    # Reshaping to obtain a sequence [B, (h//ps)*(w//ps), d_model]
    xseq = x.permute(0, 2, 3, 1).reshape(batch, -1, d_model)
    ...

```

Listing 2.1: Spatial to sequence and embedding in ViT, Pytorch code snippet

The core component of ViT is the transformer block, that implements multi-head self-attention: given the input patch embeddings $X \in \mathbb{R}^{N \times d_{model}}$, it is projected into queries (Q), keys (K), and values (V) for each of the h attention heads:

$$Q_i, K_i, V_i = XW_i^Q, XW_i^K, XW_i^V$$

where $W_i^Q, W_i^K, W_i^V \in \mathbb{R}^{d_{model} \times d_k}$ are learnable weight matrices for the i-th head and k usually equal to $d_{model}//h$.

The attention for each head is then computed using scaled dot-product attention:

$$\text{head}_i = \text{Attention}(Q_i, K_i, V_i) = \text{softmax}\left(\frac{Q_i K_i^T}{\sqrt{d_k}}\right) V_i$$

The outputs of the attention heads are concatenated and then linearly projected to produce the output:

$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, \ldots, \text{head}_h) W^O$$

where $W^O \in \mathbb{R}^{hd_v \times d_{model}}$ is another learnable weight matrix. This process transform each patch embedding, transforming them in contextualized representations by aggregating information from all other patches based on learned attention patterns. These process is both one of the main strength and weakness of the transformer block. As each patch has to attend to every other patch, it can capture long-range relationship, but the computational complexity of the multi-head self-attention operation is $O(N^2 \times d_{model})$ with respect to the image size. In fact, to address this issue, alternative architectures such as Swin Transformers [7] achieve linear complexity respect to image size by introducing a window-based self-attention mechanism that computes attention locally within non-overlapping windows. After the multi-head attention operation, each patch embedding pass trough a feed forward network (FFN). Algorithmically, the transformer block can be describe like this, taking also in account layer normalization and skip connections typically found in most implementations:

$$\begin{aligned}
X_{norm_1} &= \text{LayerNorm}(X) \\
Z_0 &= \text{MultiHeadAttention}(X_{norm_1}) \\
X' &= X + Z_0 \\
X'_{norm} &= \text{LayerNorm}(X') \\
Z_1 &= \text{FFN}(X'_{norm}) \\
X_{out} &= X' + Z_1
\end{aligned}$$

As X_{out} is shaped exactly as X, transformer blocks can be easily stacked, to obtain deep networks.

A major difference of ViT respect convolutional neural networks (CNN) is the lack of image-specific inductive biases: convolution is a process that explicitly encodes 2D spatial locality and translation equivariance, whereas the transformer architecture treats the input as a sequence of

patches without assuming such priors. This causes ViTs to under-perform, compared to CNN, when trained from scratch on mid-sized dataset[1], but to excel when pre-trained on large datasets and then adapted to down-stream tasks [1]. Pre-training of ViTs can be divided in three major category:

- **Supervised pre-training:** the ViT is pre-trained on big human-annotated datasets, like JFT-300M and ImageNet-21k, for an image classification tasks. The original ViT [1] is an example of Vision Transformer that underwent this kind of training.
- **Self-supervised pre-training:** the ViT is pre-trained on a dataset of un-labeled images. There are many possible training objectives [9], among theme there are: masked patch prediction, where part of the patch embeddings are masked and the model is trained to predict the original content of the masked patches; contrastive learning, which aims to learn representations that are invariant to data augmentations. This is achieved by creating different augmented "views" of an image and training the model to maximize the similarity between representations of the same image while minimizing the similarity with representations of different images. Popular ViT trained in such manner are BEiT, BERT Pre-Training of Image Transformers [10] and the DINO family of models [11, 12, 5]
- **Natural Language Supervision:** the ViT is pre-trained on huge dataset of image-text pairs. As this approach is at the base of vision language models, that are of particular interest for this thesis, we will go on more detail in the following paragraph.

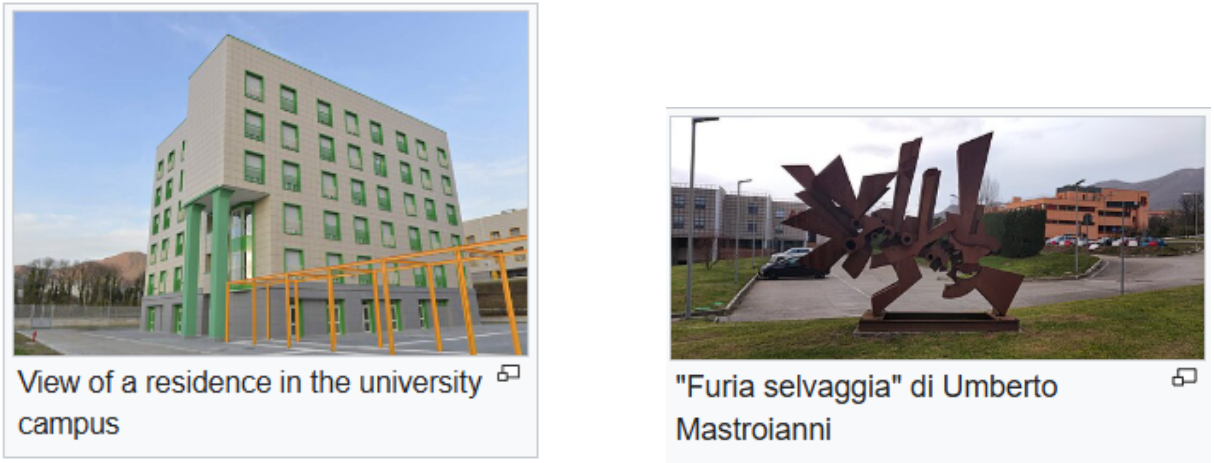
ViTs have been successfully applied in many fields of computer vision with strong performance, such as in image/video recognition, object detection and semantic segmentation, and have also sparked the development a plethora of new architectures [13].

2.2 Vision Language Models

Vision Language Models (VLMs) represent a significant step in AI, creating models that can understand images and text within a single, unified framework. Their development mirrors the trajectory of Large Language Models (LLMs): both rely on new architectures designed to

[1]DeiT [8] addresses this problem with *distillation through attention*, adding a token to the input called the 'distillation token'. This token is dedicated to reproducing the teachers decision by minimizing the cross-entropy loss between the logits predicted on this new token and the label predicted by the teacher model.

process massive datasets and, crucially, new training objectives that can extract a learning signal from vast, unlabelled (or weakly-labelled) data found on the internet. For the case of VLMs, the dataset are composed on vast collections of image-text pairs (e.g. 2.1) that can be scraped from the internet.



(a) Example of an image-text pair

(b) Another, more "noisy" sample

Figure 2.1: Two example image-text pairs that may be used to train a VLM, obtained from University Of Salerno Wikipedia page

With this huge amount of data, VLMs are pre-trained to learn joint vision–language representations in a shared semantic space where images and text referring to the same concept are projected into nearby regions. The first successful VLM trained with these approach is CLIP (Contrastive Language-Image Pre-training), introduced by OpenAI in 2021 [14]. CLIP is based on a "two-tower" architecture, presenting an image encoder such as a ViT[2], to produce an image embedding, and a text encoder, based on transformers blocks. These encoders are trained on scratch to project their respective vector embedding of images and texts containing the same semantic meaning, for example 2.1a, into nearby regions. This is achieved by introducing a contrastive learning objective. The goal is to train the two encoders so that in a batch of N (image, text) pairs, the N correct pairs have a high similarity score, while the $N^2 - N$ incorrect pairs have a low similarity score. This is implemented using a symmetric cross-entropy loss over the similarity scores, often referred to as the InfoNCE loss. The model's task is, for any given image, to "find" its correct text caption from all N text captions in the batch (and vice-versa). The pseudocode provided in the original CLIP paper illustrates this core mechanism:

[2]Also ResNet has been used as image encoder, but ViT outperforms it

```
# image_encoder - ResNet or Vision Transformer
# text_encoder - Text Transformer
# I[n, h, w, c] - minibatch of aligned images
# T[n, l] - minibatch of aligned texts
# W_i[d_i, d_e] - learned proj of image to embed
# W_t[d_t, d_e] - learned proj of text to embed
# t - learned temperature parameter

# extract feature representations of each modality
I_f = image_encoder(I) #[n, d_i]
T_f = text_encoder(T) #[n, d_t]

# joint multimodal embedding [n, d_e]
I_e = l2_normalize(np.dot(I_f, W_i), axis=1)
T_e = l2_normalize(np.dot(T_f, W_t), axis=1)

# scaled pairwise cosine similarities [n, n]
logits = np.dot(I_e, T_e.T) * np.exp(t)

# symmetric loss function
labels = np.arange(n) # The ground-truth is the diagonal
loss_i = cross_entropy_loss(logits, labels, axis=0)
loss_t = cross_entropy_loss(logits, labels, axis=1)
loss = (loss_i + loss_t)/2
```

Listing 2.2: Numpy-like pseudocode for the core of an implementation of CLIP

A critical component of this training paradigm is the batch size N. Given the inherently noisy nature of web-scraped data (e.g. 2.1b), where image-text alignment may be weak, the signal from any single positive pair is unreliable. The contrastive loss compensates for this by leveraging a massive number of negative examples. Consequently, N must be sufficiently large to ensure the gradient is stable and informative. The original CLIP, for instance, was trained with a batch size of 32,768, making the computation highly demanding but essential for learning a robust joint embedding space from noisy supervision. The resulting model is a powerful set of encoders, for which the most common application is zero-shot classification. Given an image, the model can classify it among an arbitrary set of textual class descriptions (e.g., "a photo of a dog", "a rendering of a car") without any further training. This is achieved by embedding the image and all text prompts, and then predicting the class corresponding to the text embedding with the highest cosine similarity to the image embedding. Beyond zero-shot inference, the learned encoders, particularly the image encoder, serve as a strong backbone for various downstream tasks, such as linear probing, image retrieval, and as a visual feature extractor for more complex models, capable of task such object detection and semantic segmentation. While the CLIP architecture was foundational, many modern VLMs diverge from this "train from scratch"

methodology and two tower architecture. Instead, they leverage the powerful, pre-existing capabilities of large pre-trained LLMs: this new paradigm avoids the training a text encoder from scratch and instead focuses on "aligning" a pre-trained, image encoder (often a ViT trained with CLIP's objective) to a pre-trained, frozen LLM [15, 16]. Conversely, an alternative frozen image-encoder paradigm has also been explored: a powerful, pre-trained vision encoder, such as DINOv2 or DINOv3 [17, 5], is frozen, and a text encoder is trained from scratch to align with its rich, fixed representations.

2.3 Perception Encoders

Perception encoders (PE) are a family of VLM released by Meta FAIR in 2025 [4]. They consist of a "CLIP" style pre-trained VLM, called "PE Core", from which other two VLMs have been produced: "PE Spatial", for "spatial" downstream task, such as detection and segmentation; and "PE Lang", aligning the powerful vision encoder to a pre-trained LLM (Llama 3.1).

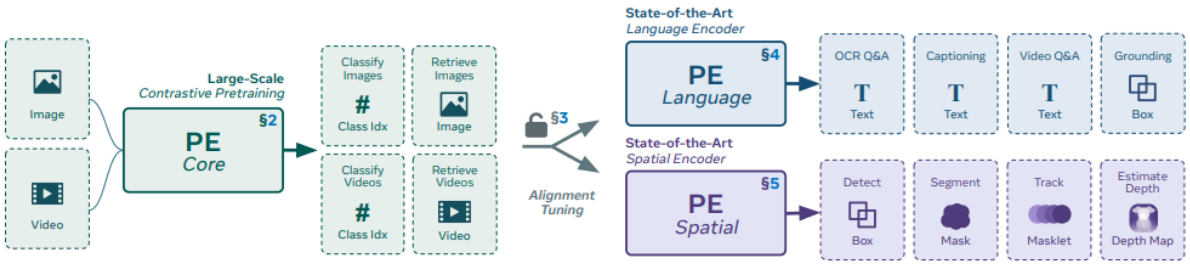


Figure 2.2: The perception encoder family of models

The PE_{core} VLM follows a classic two-tower architecture, utilizing a ViT for the visual component and a text transformer for the textual component. Deviating from the trend of multi-objective pretraining, PE_{core} relies exclusively on a vision-language contrastive loss for its initial training. This approach is enabled by a "robust image pretraining recipe" on a 5.4B image-text pairs, which enhances the standard CLIP training with several innovations. Key components include: progressive resolution training for efficiency; an increased batch size of 64K; the LAMB optimizer to stabilize large-batch training; 2D Rotary Position Embeddings; attention pooling to create the final embedding; and a novel mask regularization loss that aligns masked tokens with their unmasked counterparts. Following this initial pretraining, a second stage extends the model's capabilities to the video domain. This is achieved by finetuning the model on 22 million videos using synthetically generated captions. Video embeddings are created by average pooling features from 8 uniformly sampled frames , which are then aligned to

the text captions using the same contrastive loss. Finally, a third stage uses knowledge distillation to transfer the capabilities of the large G-scale model to the smaller B and L-scale models.

Scale	Tower	Params	Width	Depth	MLP	Heads	CLIP Dim	Context
B	Vision	0.09B	768	12	3072	12	1024	32
	Text	0.31B	1024	24	4096	16		
L	Vision	0.32B	1024	24	4096	16	1024	32
	Text	0.31B	1024	24	4096	16		
G	Vision	1.88B	1536	50	8960	16	1280	72
	Text	0.47B	1280	24	5120	20		

Table 2.1: Architectural configurations of vision and text encoders for base (B), large (L), and giant (G) model scales.

			Zero-Shot Class.					
Model	Params	Res.	ImageNet	IN-v2	ObjectNet	IN-Adv	IN-Rend	IN-Sketch
B Scale								
SigLIP2-B/16	0.1B	224	78.2	71.4	73.6	55.0	91.7	68.9
PEcore-B/16	0.1B	224	78.4	71.7	71.9	62.4	88.7	92.5
L Scale								
SigLIP2-L/16	0.3B	384	83.1	77.4	84.4	84.3	95.7	75.5
PEcore-L/14	0.3B	336	83.5	77.9	84.7	89.0	95.2	80.0
Unbounded								
SigLIP2-g-opt	1.1B	384	85.0	79.8	88.0	90.5	96.6	77.4
PEcore-G/14	1.9B	448	85.4	80.2	88.2	92.6	96.5	83.7

Table 2.2: Benchmarks of PE_{core}, compared to SigLIP2, another popular VLM

2.4 Facial Expression Recognition, solutions in the literature

In the field of Facial Expression Recognition, the research literature has been largely dominated by deep learning methodologies. CNNs set the benchmark for performance, as they are more able to learn from scratch on the size of the available dataset for FER, thanks to their inductive biases and ubiquitous ResNet architecture. Building upon this strong foundation, recent experimentation with hybrid models that combine CNNs and ViTs is proving to be a highly competitive and promising direction. Some notable models that have shown strong performance are:

- **ResEmoteNet** [18], is the current state of the art model for FER, achieving the number one spot on accuracy on popular FER dataset like RAF-DB and AffectNet. It presents an architecture based on convolutional neural networks, residual blocks (ResNets) and squeeze and excitation blocks (SENet), and has been trained on the following datasets: FER2013, RAF-DB, AffectNet-7 and ExpW.

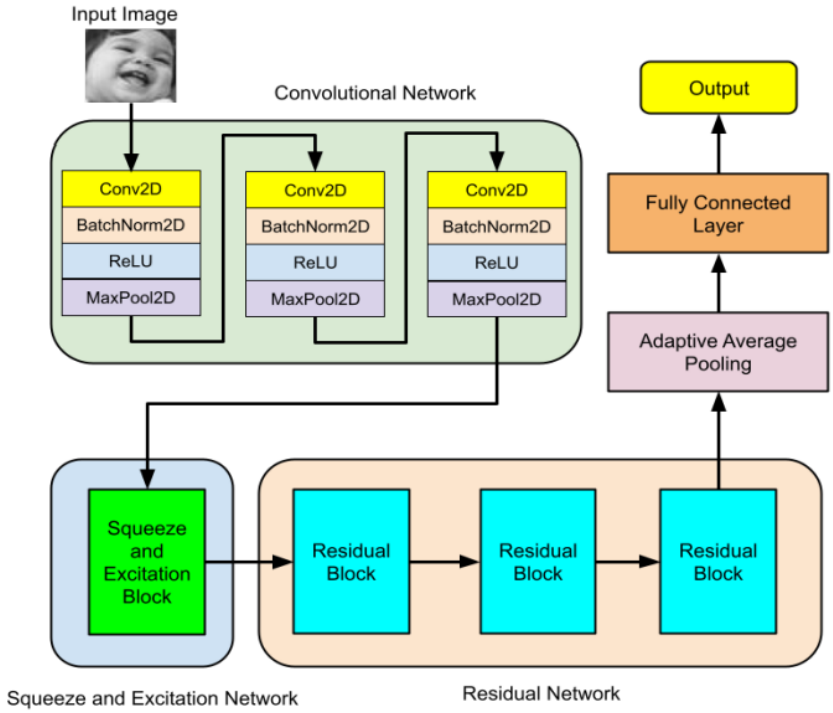


Figure 2.3: ResEmoteNet architecture

- **APViT** [19], employs an hybrid approach, combining CNN and ViT. Instead of having a shallow "CNN" to embed an image in patch embeddings, it employs a deeper CNN based network (first three stages of ir50 pre-trained on Ms-Celeb-1M) to obtain a first

feature map. This features map gets then filtered by only picking top-k token, by choosing only the one with the higher activations. This process has the goal of eliminating the less discriminative part of the face image such as the background and hairs. Then this token are passed to ViT network, with a [CLS] token, and still get gradually dropped, by considering the attention score between an image-token and the [CLS] token.

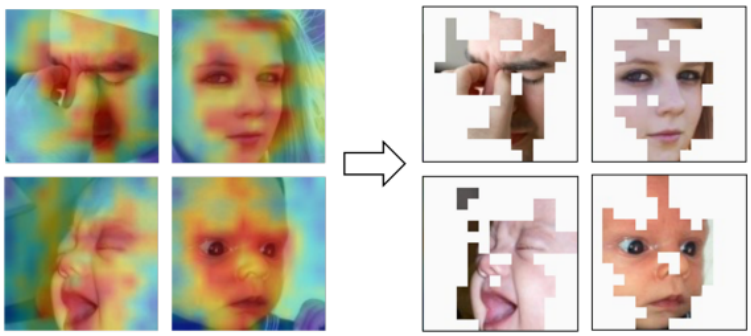


Figure 2.4: APViT approach, discarding less informative areas

- **POSTER++**[20], is built upon a dual-backbone system. It uses a frozen MobileFaceNet for facial landmark detection and an ir50 network, pre-trained on Ms-Celeb-1M, for visual feature extraction. Features are drawn from both models at various scales and are then merged using cross-attention modules. The resulting concatenated features are processed by a compact, two-layer ViT to generate the final embedding that is fed to a classifier.

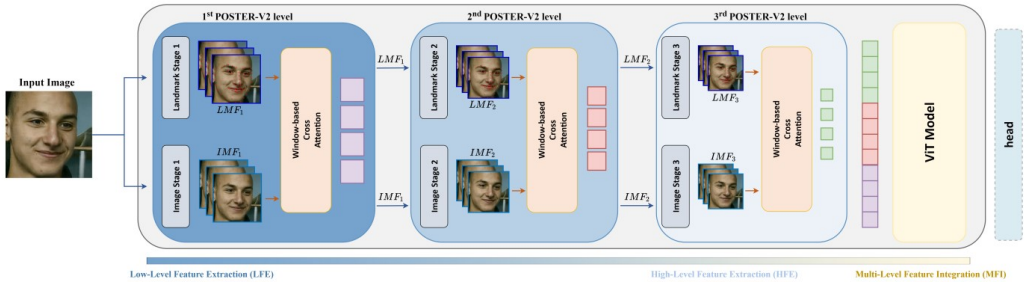


Figure 2.5: POSTER++ dual-backbone architecture

2.5 Age estimation, solutions in the literature

In age estimation, deep learning is the dominant methodology. Foundational CNNs, such ResNets and VGGs, remain a widely used and relevant models, while the research landscape has also expanded to include newer architectures like ViTs and attention based mechanisms. Regarding the methodologies, the problem is generally framed in one of three ways: regression,

that treats age as a continuous variable (e.g., "34.5 years"); age-group classification, assigning a person to a discrete category (e.g., "20-29") and ordinal regression, that decomposes the prediction into a sequence of binary classification tasks, with different granularity, each corresponding to whether the age exceeds a specific threshold (e.g., "Is age $>$ 10?", "Is age $>$ 20?", "Is age $>$ 30?"). Notable models that have shown strong performance are:

- **MIVOLO (Multi Input VOLO)** [21, 22], is a state-of-the-art model that integrates age and gender estimation into a unified, dual-input network. It is built upon the VOLO vision transformer backbone. The architecture is designed to leverage not only facial information but also person/body image data, which improves generalization and allows it to provide satisfactory results even when the face is not visible. Moreover the author collected also a new dataset (LAGENDA) and achieved SOTA results on five major benchmarks.

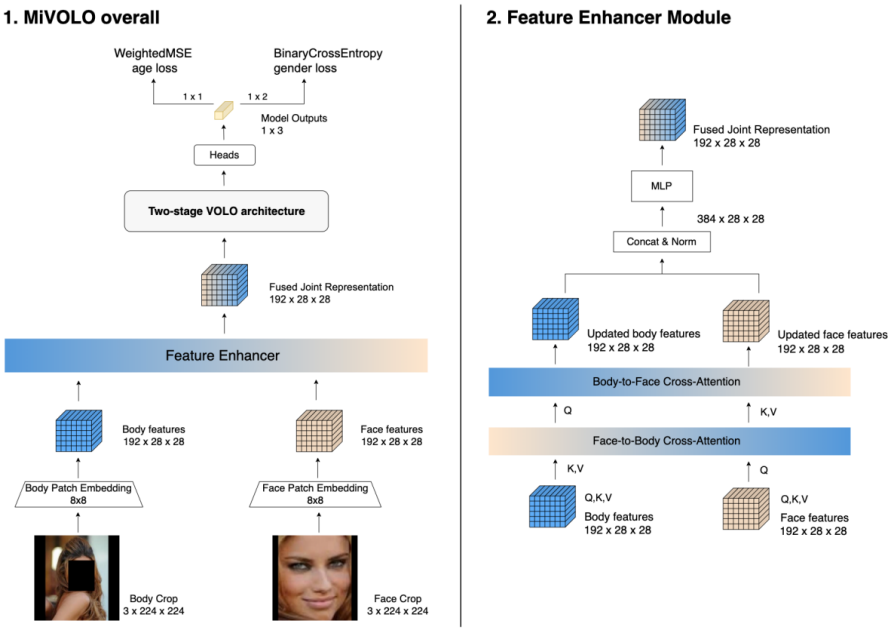


Figure 2.6: MiVolo dual-input approach

- **CoRAL (Rank consistent ordinal regression)** [23], is not a specific model architecture but a training methodology that frames age estimation as ordinal regression with the the addition of a rank consistency constraint. This constraint resolves logical contradictions by guaranteeing that the model's predictions follow the natural ordinal sequence; for example, it ensures that if the model predicts an age is $>$30, it is forced to also predict the age is $>$20. The authors demonstrated this methodology by applying it to a ResNet-34 backbone, which they named CORAL-CNN.

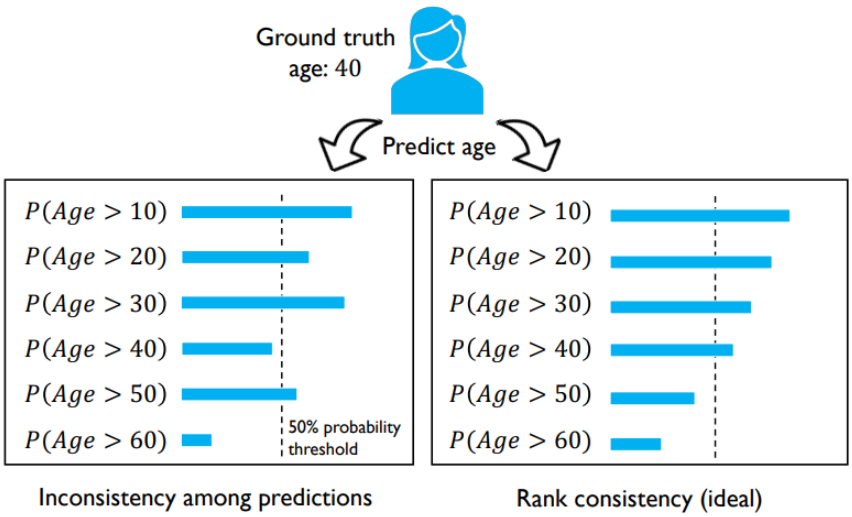


Figure 2.7: CORAL ordinal ranking formulation, with consistency constraint

2.6 Parameter-Efficient Fine-Tuning through LoRA

Low-rank adaptation (LoRA) [24] is a technique first introduced to fine-tune Large Language Models (LLMs) that has also shown successful results in computer vision tasks [25] and image and video generation tasks. The core hypothesis of LoRA is that weight updates during the adaptation of a large pre-trained model to a new task have a low "intrinsic rank". This can be mathematically described as such: given a pre-trained weight matrix $W_o \in \mathbb{R}^{d\times k}$, the weight update matrix ΔW can be approximated as $\Delta W = BA$ where $B \in \mathbb{R}^{d\times r}$ and $A \in \mathbb{R}^{r\times k}$ with the rank $r << min(d, k)$. Given this hypothesis, during training, W_o can be frozen and not receive any gradient updates, and we can reformulate the forward pass as such: $h = W_o x + BAx$ This approach has several significant benefits, making it a highly efficient and practical method for adapting large models:

- **Reduced Number of Trainable Parameters:** By freezing the large pre-trained weight matrix W_o and only optimizing the low-rank matrices A and B, LoRA drastically cuts down the number of parameters that need to be updated during training. This makes the fine-tuning process significantly less computationally intensive.

- **Lower VRAM Consumption:** The reduction in trainable parameters directly leads to a smaller memory footprint. Since gradients and optimizer states are only stored for the low-rank matrices, the overall VRAM requirement is substantially lower, enabling the fine-tuning of large models on hardware with limited memory.

- **Smaller Checkpoint Size:** Instead of saving a full copy of the fine-tuned model, only the small matrices A and B need to be stored for each task. This results in highly portable and lightweight checkpoints that are orders of magnitude smaller than the original model.
- **No Added Inference Latency:** After training, the weight update can be merged directly into the original weights by computing $W = W_o + BA$. This means the model architecture remains unchanged during inference, and there is no additional computational overhead or latency compared to the original pre-trained model.

```
# dense_pt, a pre-trained nn.Linear module
dense_pt.requires_grad = False
k = dense_pt.in_features
d = dense_pt.out_features
rank = 64 # rank << min(k, d)
lora_A = nn.Parameter(torch.zeros(rank, k))
lora_B = nn.Parameter(torch.zeros(d, rank))
nn.init.normal_(self.lora_A, mean=0.0, std = (1 / rank))

def forward_lora(x, lora_A, lora_B, dense_pt):
  # original model output
  pt_model_output = dense_pt(x)

  # the matrix product of lora_B @ lora_A results in
  # a [d,r] @ [r,k] = [d,k] shaped matrix
  # that is of equal shape of the un-approximesed weight update
  lora_output = lora_B @ lora_A @ x

  return F.ReLU(pt_model_output + lora_output)

```

Listing 2.3: LoRA pytorch-like code snippet

Moreover, LoRA can perform just as well as full fine-tuning in some cases [24, 25], but as task complexity increases, full fine-tune may still outperform LoRA considerably [26]. The success of this methodology has inspired many other studies on parameter-efficient adaptation trough low-rank decomposition [27]: Weight-Decomposed Low-Rank Adaptation (DoRA) [28] enhances LoRA by decomposing the W_o weight matrix in its magnitude vector $m \in \mathbb{R}^{1\times k}$ and its direction matrix $V \in \mathbb{R}^{d\times k}$, and directly trains the magnitude vector and uses LoRA to train the direction matrix; QLoRA [29] focuses on drastic reduction of VRAM requirements while maintaining performance through quantization; LoRA+ [30] proposes to set an higher learning rate to the B matrices, to more optimally fine-tune models with larger embedding dimension. Furthermore, solutions like mLoRA [31] have been developed to efficiently train numerous adapters in parallel by leveraging a single shared base model. Subsequently, for inference, systems such as

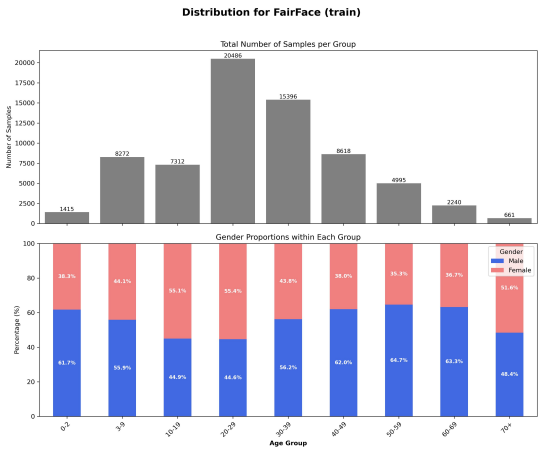
S-LoRA [32] and B-LoRA [33] can serve multiple adapters concurrently, batching requests for different tasks to transform the single large model into an efficient multi-task network.

2.7 Datasets Review

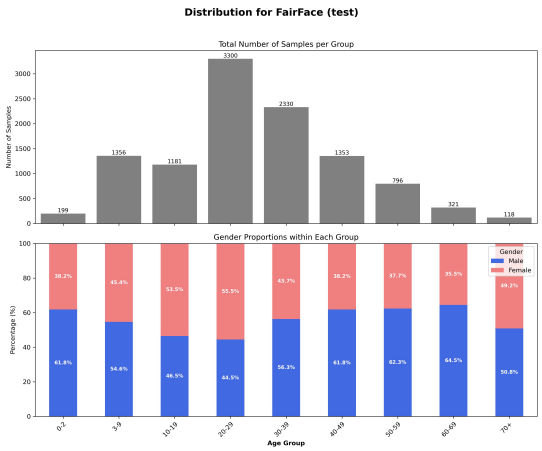
In the following paragraphs we list all the datasets used for the training, validation and testing of our models. Each dataset has been moreover processed to obtain the crop of the faces[3], using a DNN (`res10_300x300_ssd_iter_140000_fp1`) and the OpenCV library, or using the already provided bounding boxes by the authors if present. For each training set listed below, also a validation set will be extracted doing an 80-20 split.

2.7.1 FairFace

The FairFace dataset [34] contains 108,501 images, with an emphasis on balanced ethnicity composition. The faces were collected from the larger YFCC100M dataset, and labeled trough crowdsourcing for gender and age groups. It is from the FairFace dataset that we take the 9 age groups for our age classification tasks. In our experiments the FairFace dataset is used both for training and testing, using the split provided by the authors.



(a) Train Set Distribution



(b) Test Set Distribution

Figure 2.8: FairFace Dataset Distribution

[3]It is not obvious that for our tasks taking the crop of only faces is optimal, but is necessary due to the varied nature of images, which include both full-body and close-up. This preprocessing step ensures a uniform input for the models.

2.7.2 UTKFace

The UTKFace dataset [35] contains 24,103 images, labeled with ages and gender. The age range spans from 0 to 116 years old. In our experiments the UTKFace dataset will be used only for testing purposes, providing a benchmark for the model's cross-dataset generalization capabilities.

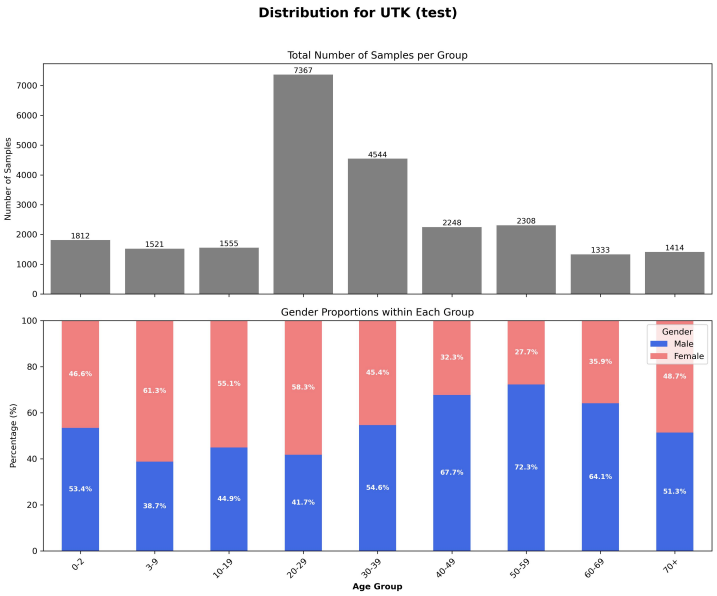


Figure 2.9: UTKFace Dataset Distribution (Test Set)

2.7.3 Lagenda

The Lagenda dataset [21] [22] contains 67,159 samples, each with labels for gender and age (ranging from 0 to 95). The dataset contains minimal celebrity data, to better reflect the real-world, in-the-wild scenario. All samples were annotated through a crowdsourcing platform, where trained and verified annotators assigned both a gender and an age to each individual. The final labels were determined using a voting mechanism based on 10 independent annotations for both age and gender. The dataset was constructed to be balanced by age distribution (in 5-year groups) up to 65 years, while also ensuring gender balance within each group, as illustrated in Figure 2.10. In our experiment, the Lagenda dataset will be used only for training and validation purposes.

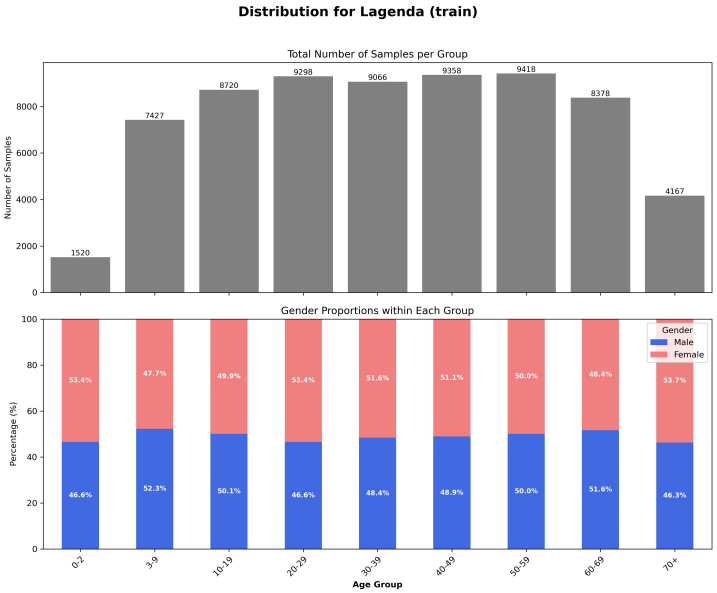
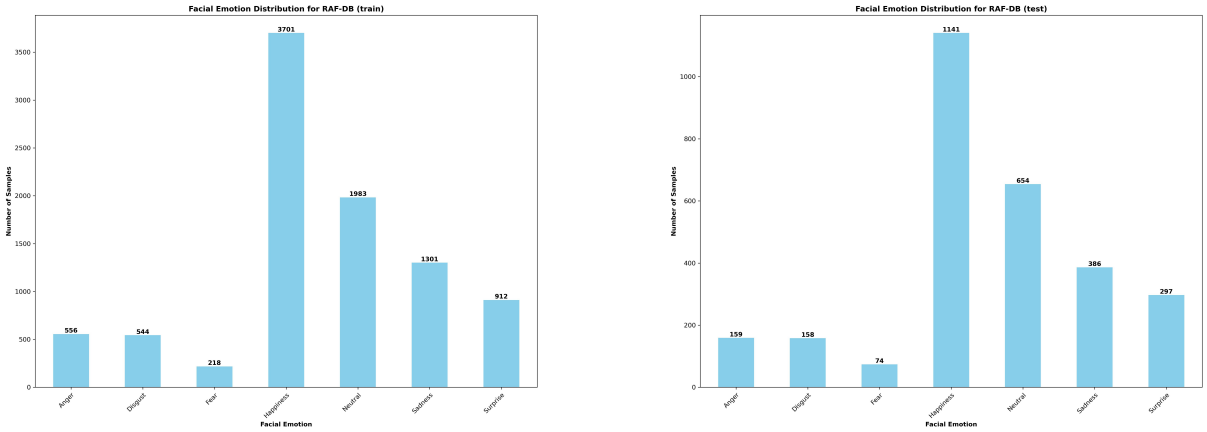


Figure 2.10: Lagenda Dataset Distribution

2.7.4 RAF-DB

The RAF-DB (Real-world Affective Face) dataset [36], is one of the reference datasets for emotion recognition tasks. The samples in it are based on the seven basic emotions theorised by Ekman: "Surprise", "Fear", "Disgust", "Happy", "Sad", "Angry", and "Neutral", and are also labeled for gender. All samples have been rigorously annotated by 40 qualified annotators. The labels were then refined by performing a validation based on an Expectation-Maximization algorithm to remove noisy labels, achieving a Cronbach's alpha[4] coefficient of 0.996, indicative of high reliability. The samples labeled for emotion are by far the fewest in our combined dataset and, as noted, are highly unbalanced; in chapter 3.2 we explain how we tackle this problem. The RAF-DB dataset will be used for both training and testing.

[4]Cronbach's alpha is a measure of internal consistency, used to assess the reliability of a set of items or, in this case, annotations.



(a) Train Set Distribution

(b) Test Set Distribution

Figure 2.11: RAF-DB Dataset Distribution

2.7.5 VggFace2

The VggFace2 [37] dataset contains 3.31 million images of 9131 subjects, labeled by gender. Moreover, each samples has been also labeled with age, with the process documented in [38]. In our experiments the VggFace2 dataset will be used only for testing, providing a well established benchmark dataset for our tasks. The choice to exclude the VggFace2 for training is to avoid to bring an heavy bias on picture of celebrities and limit the training time required.

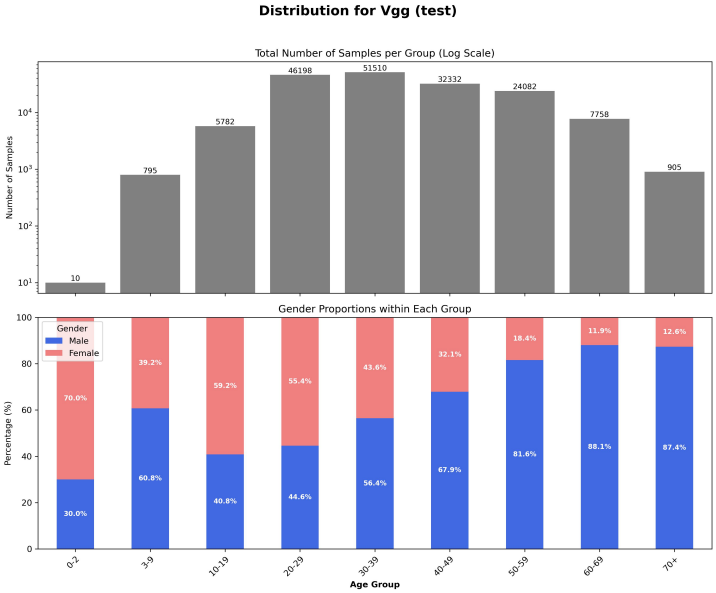
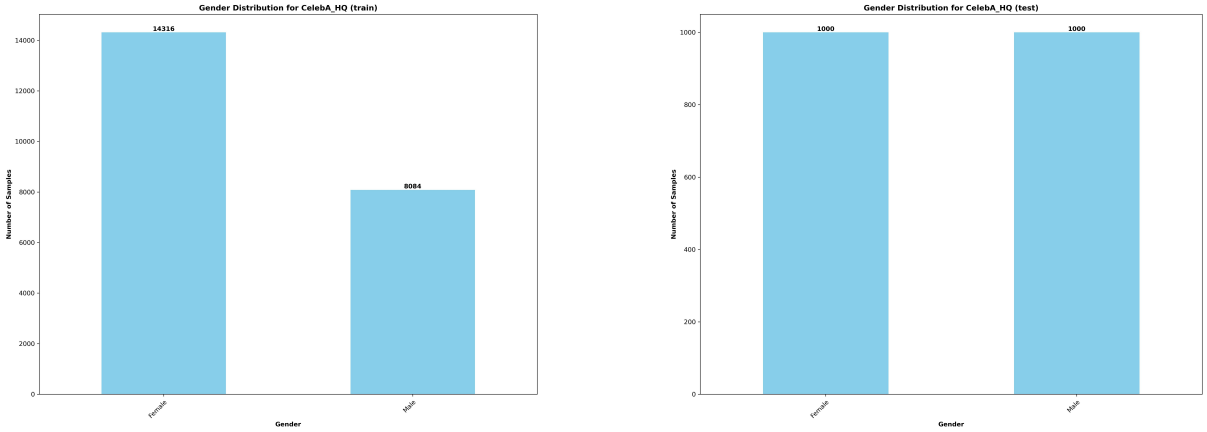


Figure 2.12: VggFace2 Dataset Distribution (Test Set), notably, the age group graph is reported in log-scale

2.7.6 CelebA-HQ

The CelebaHQ dataset [39] [40] provides 30,000 high-quality images, labeled by gender. In our experiments the CelebaHQ dataset will be used both for training and testing.



(a) Train Set Distribution

(b) Test Set Distribution

Figure 2.13: CelebA-HQ Dataset Distribution

Chapter 3

Methodology

In this chapter will go over the techniques used to adapt the pre-trained model for the three tasks of our interest. The model that will be the focus of our experiment is the vision encoder of the already introduced Perception Encoders 2.3, in particular our focus will be on the vision transformer "PE-Core-L/14 336px". This model has been chosen for two reasons: its strong performance on popular benchmark datasets in zero-shot classification settings, that can be seen in table 2.2, and the open-source availability of the model code. For each of the proposed solutions, both a single-task and multi-task approach has been explored.

3.1 Shared training settings

The proposed solutions all share some common approaches, that we will list and explain in this paragraph.

- **Optimizer: AdamW**, the AdamW [41] is an enhancement of the Adam optimizer, that decouples the weight-decay regularization with the adaptive learning rate of Adam, making it equivalent to the L2 regularization in the SGD algorithm. This optimizer has already been used by the authors of Perception Encoders [4] to adapt and probe the chosen vision encoder, making it a reasonable choice. As for the choice of the weight-decay value, the authors of Perception Encoder have used an hyper parameter sweep to find the best value, for linear-probing tasks, but as this parameter has not been made public, we chose 0.01 as our weight decay, that is the default value of Pytorch and reasonable for our task.
- **Scheduler: Reduce on plateau**, the reduce on plateau scheduler lowers by a factor of 10 the learning rate after the validation loss metrics does not improve for 5 epochs.

- **Automatic mixed precision (AMP) training with Autocast and GradScaler**, AMP is a tool offered by the Pytorch framework that allows the training of a DNN with a lower memory foot-print and in faster times. This is done by using Autocast, that casts to half-precision some operations, like the matrix multiplications in the linear layers and convolutions, while keeping the full precision for operation like reductions (that are typically used when calculating loss values). GradScaler ensures that half-precision gradients with small magnitudes don't get flushed to zero ("underflow"), by scaling them [42]. This allows a substantial speed-up in training, while not compromising on accuracy.

- **Data augmentation and processing**, each image sample is resized to 336px and zero-centered. Moreover each sample is subjected to random "safe" transformations before being fed to the model. This process artificially expands the training set, which enhances the model's ability to generalize and mitigates the risk of overfitting.

```
train_transforms = T.Compose([
  T.RandomHorizontalFlip(),
  T.ColorJitter(brightness=0.2,contrast=0.2,saturation=0.2),
  T.RandomAffine(degrees=10,translate=(0.05,0.05), \
  scale=(0.9,1.1)),
  T.Resize((336, 336)),
  T.Normalize([0.5, 0.5, 0.5],[0.5, 0.5, 0.5]),
  T.RandomErasing(p=0.5,scale=(0.02, 0.2),ratio=(0.3, 3.3)),
])

```

Listing 3.1: Transformation applied to images, Pytorch code snippet

This transformation are particularly useful in the multi-task setting and with balanced sampling, as sample gets sampled more than once in the same epoch, this ensures that in an epoch the model never see the exact same image twice.

- **Loss Functions**, As we are approaching three classification tasks, we use **Cross Entropy Loss**, with the loss values averaged (mean reduction) over each batch for all tasks. The task-specific loss functions will be unweighted, as we will tackle class unbalance trough sampling 3.2. For the age-classification task, also ordinal loss has been explored, has a ordinal relation is present between the labels of the problem, but in preliminary experiments has not shown an higher performance compared to cross-entropy loss.

Figure 3.1: Data augmentation transformations examples, starting from already cropped and resized image

All the training will be done using an early-stop policy, based on validation loss for single-task and average accuracy for multi-task, on validation sets: if after 10 epochs there is no improvement on the validation-set, we stop the training. We switch to average accuracy for multi-task, as it is a more reliable composite metric for compared to average validation loss, so to avoid being misled by the varying scales of the different task losses.

The training, validation and testing will be run on an NVIDIA L40S GPU.

Table 3.1 recaps how the dataset will be used.

Table 3.1: Dataset composition and usage

Dataset	**Train**	**Val**	**Test**	**Tasks**
LAGENDA	✓	✓	-	Gender, Age
FairFace	✓	✓	✓	Gender, Age
RAF-DB	✓	✓	✓	Emotion
CelebA-HQ	✓	✓	-	Gender
UTKFace	-	-	✓	Gender, Age
VGGFace2	-	-	✓	Gender, Age

3.2 Balanced Sampling

As it can be seen by the distribution of the datasets that are reported in the dataset review chapter 2.7 and by looking at the table 3.2, for the age and emotion classification tasks we have a very unbalanced datasets. The skewed distributions, if not tackled, would leads toward models with a bias towards the majority class, that do not necessary represents the real distribution. To address the class imbalance, we first compute a weight for each class using the inverse frequency method. Then, for each image, we determine its final sampling weight by averaging the pre-calculated weights of its available age, emotion, and gender labels. These resulting

weights are supplied to PyTorch's `WeightedRandomSampler` to achieve a more balanced class distribution during training.

Table 3.2: Distribution of the training set. The "Weighted %" columns represent the effective distribution per epoch when using weighted sampling, obtained via a 10-iteration Monte Carlo simulation.

Age Group	**Samples**	**%**	**Weighted %**
0-2	2935	2.15%	6.63%
3-9	15 699	11.48%	11.31%
10-19	16 032	11.72%	11.36%
20-29	29 784	21.78%	16.37%
30-39	24 462	17.89%	14.56%
40-49	17 976	13.15%	12.14%
50-59	14 413	10.54%	10.89%
60-69	10 618	7.76%	9.44%
70+	4828	3.53%	7.31%

Gender	**Samples**	**%**	**Weighted %**
Male	82 147	48.79%	49.26%
Female	86 215	51.21%	50.74%

Emotion	**Samples**	**%**	**Weighted %**
Surprise	912	9.90%	12.14%
Fear	218	2.37%	8.44%
Disgust	544	5.90%	9.99%
Happy	3701	40.16%	26.85%
Sad	1301	14.12%	14.20%
Angry	556	6.03%	10.25%
Neutral	1983	21.52%	18.13%

(a) Task distribution

Task	**Samples**	**%**
Gender	168362	100.00%
Age	136747	81.22%
Emotion	9215	5.47%
Total	168362	–

3.3 Linear probing

Linear probing is the simplest and most efficient adaptation strategy, designed to evaluate the raw, out-of-the-box feature quality of a pre-trained encoder. In this methodology, the entire vision encoder backbone, f_θ, is "frozen," meaning its parameters θ do not receive any gradient updates during training.

The adaptation is performed by training only a new classification head, h_ϕ, which is appended to the encoder. This head takes the d_{model} dimensional global embedding from the encoder's attention pooling layer and maps it to the C task-specific classes. We explored two architectures for this head:

- **Simple Head:** A minimal head consisting of a dropout layer, for regularization, followed by a single linear layer that maps the embeddings directly to the C output classes.
- **Deeper Head:** A more complex, non-linear head with the following structure:

  Dropout $\rightarrow$ Linear(d_{model}, d_{model}) $\rightarrow$ GELU $\rightarrow$ Linear(d_{model}, C).

In our experiments, this deeper head consistently achieved better performance. This suggests that while the pre-trained features $f_\theta(x)$ are highly informative, they are not perfectly linearly separable for our specific tasks. The added non-linear projection, allows the model to learn a more complex and robust mapping from the fixed features to the target labels. Given this findings, all other approach will employ a deeper head for classifications (we will keep referring to this approach as linear probing, even if not technically correct). Since the gradients from one task cannot influence the parameters of another (as only the mutually exclusive heads are trained), multi-task learning for linear probing is not possible. We therefore train a separate head for each downstream task individually. As mentioned before, this is the least memory footprint demanding method, as only 0.33% of the parameters will get trained.

Starting LR:	$1e-4$
Trainable parameters:	$1,058,825$
Total parameters:	$320,324,629$
Percentage of trainable:	0.33%

3.4 Attention probing

Attention probing is an extension of the linear probing methodology, offering a slightly deeper level of adaptation. It operates on the hypothesis that while the patch-level features from the pre-trained encoder f_θ are powerful, the default mechanism for aggregating them into a single global embedding, the attention pooling layer, may be sub-optimal for our specific facial analysis tasks. The original pooling mechanism was trained to summarize an entire image for a text caption, whereas our tasks require focusing on specific and potentially subtle facial regions.
In this setup, we freeze the vast majority of the encode, but we "unfreeze" and train the parameters of the attention pooling module, θ_{pool}. This module is responsible for weighting and combining the final patch embeddings into a single d_{model} dimensional vector.

$$\text{logits} = h_\phi(f_{\theta_{pool}}(f_{\theta_{backbone}}(x)))$$

By fine-tuning θ_{pool} alongside the new classification head h_ϕ, the model can adjust how it weights different patch embeddings to construct a global representation that is more discriminative for classifying gender, age, and emotion, rather than relying on the general-purpose summary vector from pre-training.
This approach remains parameter-efficient, as the attention pooling layer represents a small fraction of the total model parameters. Moreover, because the weights of the pooling layer are now trainable, we add a batch normalization layer immediately before the classification head to stabilize its inputs; this layer will be present in all subsequent methods as well.

Starting LR:	$1\text{e}-4$
Trainable parameters:	$13,656,073$
Total parameters:	$320,324,629$
Percentage of trainable:	4.29%

3.5 Fine-tuning

Full Fine-tuning the pre-trained weights of the encoder would require the unfreezing and training of all of the 320 million parameters of the vision encoder, for our setup this is not possible and neither particularly convenient. Firstly, training a model of this size is computationally prohibitive, requiring VRAM exceeding the capacity of our available hardware. Secondly, even

with sufficient hardware, fine-tuning such a large model on our relatively limited datasets carries a very high risk of overfitting. Finally, updating the entire network risks catastrophic forgetting, where task-specific gradients could destroy the powerful, generalized knowledge acquired during the initial 5.4 billion pair pre-training. Given these constraints, we adopt a compromise strategy: partial fine-tuning. In this approach, we freeze the vast majority of the encoder and only unfreeze the parameters of the final four transformer blocks. The new classification head h_ϕ and the attention pooling layer θ_{pool} are also unfrozen and trained. Unfreezing only the final layers is a common technique to adapt large pre-trained models. It allows the model to adjust its final output representations for the new task while keeping the foundational parameters stable. To train with this approach we employ a differential learning rate: the newly initialized classification head h_ϕ and the unfrozen attention pooling layer θ_{pool} are trained with a learning rate of 1e-4. In contrast, the pre-trained weights of the final four transformer blocks are updated with a smaller learning rate of 1e-5 (one-tenth of the head's LR).

$$
\begin{aligned}
\text{Starting LR Head:} &\quad 1\text{e}-4 \\
\text{Starting LR Backbone:} &\quad 1\text{e}-5 \\
\text{Trainable parameters:} &\quad 64,040,969 \\
\text{Total parameters:} &\quad 320,324,629 \\
\text{Percentage of trainable:} &\quad 20.13\%
\end{aligned}
$$

3.6 Parameter-efficient fine-tuning

In this approach, we apply the Low-Rank Adaptation (LoRA) methodology, previously detailed in the literature review, to adapt the vision encoder. Instead of fine-tuning a specific subset of layers, we freeze the entire pre-trained backbone f_θ. We then inject trainable LoRA adapters into **every linear layer** within the backbone, with a **rank of 64**, targeting to half the number of trainable parameters compared to the partial fine-tuning method. We do not use a scaling factor for the LoRA update.

This comprehensive adaptation includes the Query (W^Q), Key (W^K), Value (W^V), and Output (W^O) projection matrices in the multi-head attention mechanisms, as well as the two linear layers of the Feed-Forward Network (FFN), across all 24 transformer blocks. Furthermore, the attention pooling layer is also adapted using this same LoRA technique. The only components trained with their full parameters are the new classification head h_ϕ.

We also adopt the LoRA+ optimization, which was shown to be more effective for large-embedding models like our $d_{model} = 1024$ backbone . This method addresses a suboptimality in the original LoRA by setting different learning rates for the adapter matrices A and B. We set the learning rates according to the ratio $\eta_B = \lambda\eta_A$, which allows for more efficient feature learning, particularly in terms of time of convergence. Based on the empirical findings presented in the article, we use the recommended fixed ratio $\lambda = 6$. We still employ a differential learning rate, by setting to $1e-5$ the learning rate of the LoRA matrices in the backbone and to $1e-4$ the learning rate for the LoRA matrices in the attention pooling layer.

Furthermore, we also used DoRA: **Weight-Decomposed Low-Rank Adaptation**. We already mentioned this method briefly in the literature review, the core technique involves decomposing a pre-trained weight matrix, $W_0 \in \mathbb{R}^{d\times k}$, into two separate components: a magnitude vector, $m \in \mathbb{R}^{1\times k}$, and a direction matrix, $V \in \mathbb{R}^{d\times k}$. This decomposition is initialized based on the pre-trained weight W_0, with the magnitude $m = \|W_0\|_c$ and the direction $V = W_0$. The term $\|\cdot\|_c$ represents the vector-wise norm across columns.

During fine-tuning, DoRA updates both of these components. To maintain parameter efficiency, it specifically applies a LoRA update, $\Delta V = BA$, to the directional component, while the magnitude m is treated as a separate trainable vector. The final adapted weight W' is then calculated as such:

$$W' = m\frac{V+\Delta V}{\|V+\Delta V\|_c} = m\frac{W_0+BA}{\|W_0+BA\|_c}$$

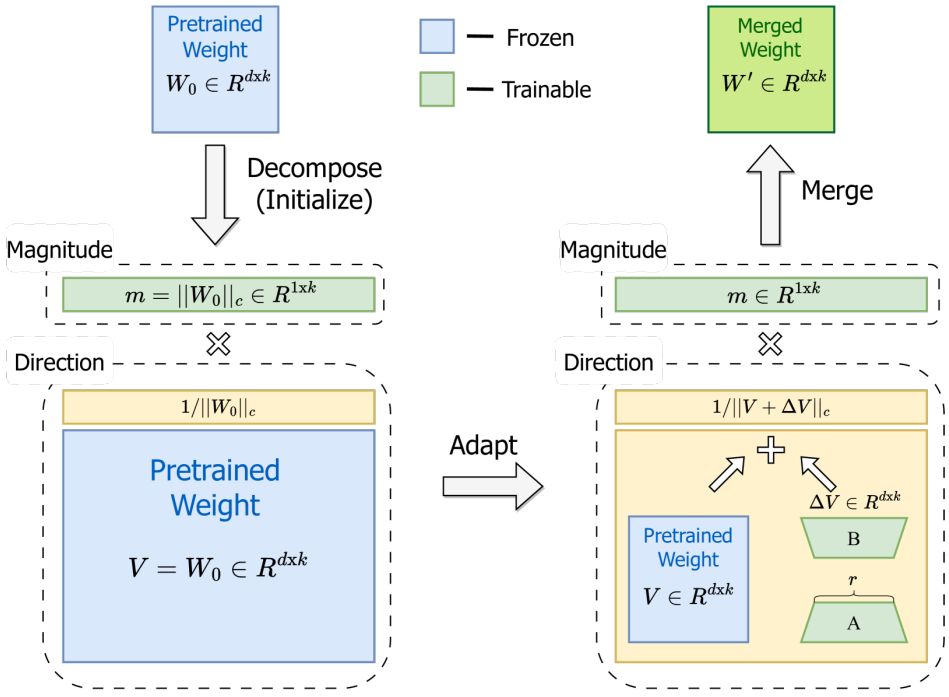


Figure 3.2: DoRA approach

This approach aims to enhance the learning capacity and training stability of LoRA, making its learning behavior more closely resemble that of full fine-tuning. A key advantage is that, like LoRA, DoRA does not introduce any additional inference overhead, as the adapted magnitude m and the final directional component can be merged back into a single weight matrix W' after training.

$$\begin{aligned} \text{Starting LR } (\eta_A): &\quad 1\text{e}-5 \\ \text{LoRA+ Ratio } (\lambda): &\quad 6 \\ \text{Trainable parameters:} &\quad 29,352,981 \\ \text{Total parameters:} &\quad 346,504,213 \\ \text{Percentage of trainable:} &\quad 8.47\% \end{aligned}$$

It's interesting to note that this parameter-efficient approach modifies all the parameters in the 24 blocks yet trains fewer than half the parameters (8.41%) compared to partial tuning, which only unfreezes 4 blocks (20.13%). Moreover, LoRA prevents catastrophic forgetting: the original pre-trained weights are frozen, preserving their general knowledge and all new learning is isolated within the small, low-rank adapters, which prevents destructive updates to the core model.

3.6.1 Multi Task LoRA

For the multi-task PEFT setting, we also conduct an experiment using the Multi-Task LoRA (MTLoRA) framework [43]. This approach is specifically designed for multi-task learning and introduces a combination of Task-Agnostic and Task-Specific modules. This structure is intended to disentangle the parameter space, allowing the model to simultaneously learn shared features while also specializing in individual task requirements. The original MTLoRA paper, which focuses on dense prediction and uses hierarchical-transformer as a backbone, inserts task-specifics adapters at multiple stages to capture multi-scale features. However, given that our three tasks are classification-based and we use a standard ViT implementation, we hypothesize that task differentiation is most crucial at the final representation layer rather than at intermediate feature maps.

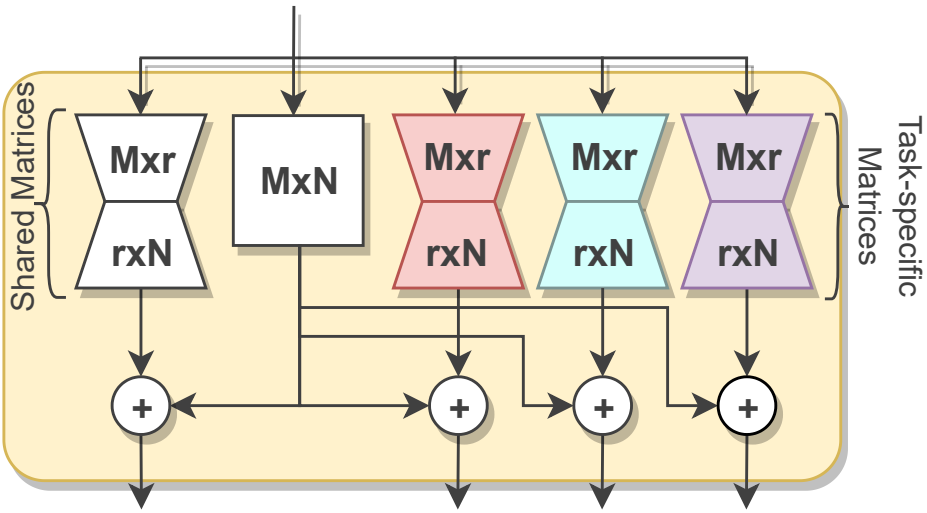


Figure 3.3: Task specific adapters, in our implementation we do not include the shared matrices, as they are used when a TS-LORA module is followed by a TA-LORA module

Therefore, we implement a simplified version of this framework. We define Task-Agnostic (TA-LoRA) adapters as a single set of DoRA weights[1] (rank 64) that are shared by all tasks. Conversely, Task-Specific (TS-LoRA) adapters are unique modules, where a separate set of LoRA weights (without magnitude decomposition) is trained for each individual task. We apply standard TA-LoRA adapters to the first 23 transformer blocks, using the same configuration described previously. Then, we apply TS-LoRA modules only to the linear layers (Q, K, V, O, and FFN) within the final (24th) transformer block. Crucially, the attention pooling layer is also replaced with a MTLoRA equivalent: the modified pooling block uses unique, trainable probes for each task and applies TS-LoRA adapters to all its internal linear layers. This allows the model to take the final shared features from the backbone and perform a specialized, task-specific pooling to generate the final representation for each classification head, while keeping a limited number of trainable parameters.

Notably this MTLoRA approach is the only one that adds adapters that cannot completely be merged into the backbone, and so that marginally affect the memory-footprint and latency of the model during inference.

[1] Different from the original implementation that used standard LoRA

Starting LR (η_A) :	$1e-5$
LoRA+ Ratio (λ) :	6
Trainable parameters:	$36,730,901$
Total parameters:	$354,093,077$
Percentage of trainable:	10.38%

3.7 Multi-task Learning

In the multi-task learning, our goal is to train the model such that its capable of classifying the three tasks in a single forward-pass. To achieve this, we have to adapt our architecture, by adding a classification head per task, and conjointly train them with the backbone. Moreover, Multi-task learning present a list of challenges that have to be tackled:

- **Missing labels**, no sample in our combined dataset presents labels for the three tasks.
- **Task unbalance**, as it can be seen in table 3.2a, the samples labelled with emotion are heavily underrepresented in our combined dataset, being only 5.47%. Without addressing this issue, the training for the emotion task would be heavily hindered in favor of the age and gender task.
- **Multi task loss**, we need to define a new loss functions that combines the three single-task loss functions.

In the following subsections we describe the proposed solution for the listed problems.

3.7.1 Handling missing labels with masked labeling

To address the issue of missing labels, the masked labeling technique is employed during the computation of the loss for the age and emotion tasks. In this approach, when a label for a specific task is absent for a given sample x_i, its corresponding label y_i is assigned a predefined `ignore_idx` value. Then inside a batch the loss is computed as such:

$$L_t = \frac{1}{\sum_{i=1}^{N} 1(y_i \neq \text{ignore\_idx})} \sum_{i=1}^{N} 1(y_i \neq \text{ignore\_idx}) \cdot \ell(f_t(x_i), y_i)$$

Where N is the number of samples in the batch, ℓ is the cross entropy loss function and $f(x_i)$ is the output of the network for the task.

3.7.2 Handling task unbalance by batch balancing

If we sampled our dataset as is, the composition of a batch of samples would only have around 5% of images labelled with emotion. This low value would lead to highly noisy gradients for the training of the emotion task. Especially disruptive would be the batches without even a single label for the emotion task, as it would be equivalent to a loss of zero for the batch that would be highly misleading. To counter-act this, we replicate emotion sample until they become a third of the dataset, so that a batch has on average 33% of samples labelled with emotion, that in combination with the batch size of 128 addresses the noisy gradient problem.

3.7.3 Multi-task loss

As our problem consists of three classification tasks, each with a unique number of labels (two for the gender task, seven for the emotion task, and nine for the age task), this leads to task losses with different scales. A simple summation of the three task may lead the task with higher loss to dominate the training loss at the expense of the other tasks. For this reason, we will explore two methods to balance the losses: exponential moving average (EMA) loss weighting and uncertainty weighting (UW). We will test both methods.

Exponential moving average

We use exponential moving average (EMA) loss weighting [44], with the goal to have each loss on the scale of one. Given the weighted multi-task loss:

$$\mathcal{L}_{mtl} = \lambda_a * L_a + \lambda_g * L_g + \lambda_e * L_e$$

Where L_a correspond to the loss for the age task and λ_a to its corresponding weight, and same nomenclature for the gender and emotion task, we compute the generic λ_k weight for task k as such:

$$\tilde{L}_k(t) = \beta L_k(t) + (1-\beta)\tilde{L}_k(t-1)$$

$$\lambda_k(t) = \frac{1}{\tilde{L}_k(t)}$$

Where $L_k(t)$ represent the loss at training iteration t for task k at (this meaning the loss computed batch per batch) and $\tilde{L}_k(t)$ represent the exponential moving average of that loss. The

hyperparameter β controls the decay rate of the moving average, and for our experiment we set it to 0.95 to ensure a stable calculation of the EMA losses. In fact, we update the λ weight at the start of each epoch so that the loss weights remain constant for an entire pass over the training data: the final value of the EMA at the end of epoch E-1 is used to calculate the fixed weights λ_a, λ_g, and λ_e that will be applied throughout all of epoch E.

Uncertainty Weighting

With Uncertainty Weighting (UW) we let the model learn how to balance the different task losses by itself [45]. The main idea of this method is to model the homoscedastic[2] uncertainty of each task. For classification, this is achieved by changing how we model the output of our classifier, from:

$$Softmax(f^{\theta}(x))$$

to:

$$Softmax(\frac{1}{\sigma^2}f^{\theta}(x))$$

Here, σ^2, is a measure of the task uncertainty. A more uncertain task, will have it's output distribution been made more uniform (squashed), to match the high variance of the task. By deriving the multi-task loss function following this approach, we arrive at an objective that learns these uncertainty weights automatically. For our three classification tasks (age, gender, and emotion), the final multi-task loss to be minimized is the following:

$$\mathcal{L}_{mtl}(W, \sigma_a, \sigma_g, \sigma_e) = \frac{1}{\sigma_a^2}\mathcal{L}_a(W) + \frac{1}{\sigma_g^2}\mathcal{L}_g(W) + \frac{1}{\sigma_e^2}\mathcal{L}_e(W) + \log\sigma_a + \log\sigma_g + \log\sigma_e$$

Where $\mathcal{L}_a(W)$, $\mathcal{L}_g(W)$, $\mathcal{L}_e(W)$ are the standard cross-entropy loss defined for each tasks and σ_a, σ_g, σ_e are learnable positive scalar parameters representing the uncertainty for each task. Notably, the $\log\sigma_k$ terms act as regularizers, penalizing the model if the uncertainties measure grows too large (without them the model would just put the variances to the highest value possible to minimize the loss).

In practice, given that variances have to be non-negative and we have to avoid division by zero and numerical instability, it's more convenient to learn the logarithm of the variance $s_k :=$

[2]This is a type of uncertainty that is task-dependent, not input-dependent, and captures the inherent noise level of a task. It can be seen as a measure of the irreducible error that afflicts a classification task.

$\log(\sigma_k^2)$, transforming the loss function during implementation to:

$$\mathcal{L}_{mtl}(W, s_a, s_g, s_e) = e^{-s_a}\mathcal{L}_a(W) + e^{-s_g}\mathcal{L}_g(W) + e^{-s_e}\mathcal{L}_e(W) + \frac{1}{2}s_a + \frac{1}{2}s_g + \frac{1}{2}s_e$$

With this adaptive weighting approach, we expect the age classification task to be assigned a lower weight. The reason is that age prediction contains more inherent noise compared to gender or emotion recognition: whereas gender and emotion can be directly determined from facial features, age is influenced by external variables such as lifestyle and genetics that aren't apparent in images. Additionally, the distinctions between adjacent age ranges (for example, '20-29' versus '30-39') are naturally vague. As a result, the age loss will automatically receive a reduced weight in the overall objective function. Nevertheless, this reduction may not substantially decrease the task's impact on training, since age classification involves 9 classes, its cross-entropy loss will inherently be larger in magnitude than the 2-class gender task or 7-class emotion task.

Chapter 4

Experimental Result

This chapter presents the results obtained using the methodologies described in Chapter 3. As a point of clarification, any reference here to a DoRA specifically denotes the 'DoRA plus LoRA+' combination (as detailed in Section 3.6). Furthermore, we define "average accuracy" as the mean of the accuracy scores achieved on each separate dataset. This is distinct from, and should not be confused with, calculating a single accuracy score from a test set where all datasets are combined. Moreover, the calculation of the avarage accuracy excludes the value obtained on the VggFace2 datasets for the age task, as they are synthetically obtained. Finally, when analyzing these results, we must also consider that our prediction pipeline includes a face recognition DNN, which introduces an additional potential source of error.

4.1 Metrics

To evaluate the performances of the various experimented we will evaluate the following metrics:

- **Accuracy**: The standard measure of correct predictions over the total number of samples for a given task k.

$$\text{Accuracy}_k = \frac{\text{Number of correct prediction for task } k}{\text{Total number of samples for task } k}$$

- **Balanced Accuracy**: This is the average of recall (sensitivity) obtained on each class. It is a more robust metric than standard accuracy, as it gives a fair score by preventing the majority classes from skewing the results, which is crucial for our imbalanced age and

emotion tasks. For a task k with C_k classes. For a task k with C_k classes, it is defined as:

$$\text{Balanced Accuracy}_k = \frac{1}{C_k} \sum_{i=1}^{C_k} \text{Recall}_i = \frac{1}{C_k} \sum_{i=1}^{C_k} \frac{\text{TP}_i}{\text{TP}_i + \text{FN}_i}$$

where TP_i and FN_i are the number of true positives and false negatives for class i, respectively.

- **Number of parameters**: This metric quantifies the model's memory footprint. We have already reported in chapter 3 the number of parameter instantiated during training, in this chapter, the number will refer to the parameter required for inference.
- **GFLOPs**: (Giga Floating Point Operations) This measures the computational complexity of a single forward pass. It serves as a hardware-agnostic indicator of **inference latency**. A lower GFLOPs count suggests a faster model, which is crucial for deployment. This will be calculated for a single 336x336 image input.

From the checkpoints saved during training, we select the one that achieved the highest validation accuracy for the single-task model, and the one with the highest average mean accuracy for the multi-task model, to be used for testing.

4.2 Baseline

To contextualize the contribution of the methodologies applied, we use as our baseline the pre-trained PE-Core-L model in a zero-shot scenario. The baseline is built by manually designing task-specific text prompts for each class using the following standard templates for each task:

- **Age groups**: *"A photo of a person between ⟨age range⟩ years old"*
- **Gender**: *"A photo of a ⟨gender⟩ person"*
- **Emotion**: *"A photo of a/an ⟨emotion⟩ person"*

The prediction for each task is computed using the standard zero-shot classification method, that has already been briefly described in the literature review in the chapter on VLMs 2.2. First, the image is passed through the visual encoder to get an image embedding. Concurrently, all text prompts for a task (e.g., "A photo of a male person", "A photo of a female person") are passed through the text encoder to get a set of text embeddings. The cosine similarity between

the image embedding and each text embedding is calculated, and the class corresponding to the text prompt with the highest similarity score is selected as the prediction.

This baseline serves as the reference point against all adaptation methods explored in this thesis and allows us to quantify the performance gain achieved by fine-tuning the model for these specific facial analysis tasks.

UTK-Age	UTK-Gender	FairFace-Age	FairFace-Gender	RAF-DB	VGG-Age	VGG-Gender
48.62%	96.63%	46.11%	97.60%	66.57%	42.01%	95.78%

Table 4.1: Baseline zero-shot accuracy results across testing datasets for age, gender, and emotion recognition tasks.

Age*	Gender	Emotion	Global
47.36%	96.67%	66.57%	69.61%

*Average calculation excludes the VggFace2 dataset as its age-labels data are synthetically obtained.

Table 4.2: Mean baseline accuracy for age, gender, and emotion recognition tasks.

Component	Parameters
Text Encoder	353,986,561
Visual Encoder	318,212,106
Total Parameters	**671,137,793**
GFLOPs	**699.76**

Table 4.3: Number of parameters used by the zero-shot baseline during inference

From Table 4.1, which reports the zero-shot accuracy, we can see how the baseline achieves competitive performances on gender classification tasks, with accuracy values ranging from 95.78% to 97.60%, while for age group classification the accuracy values go from 42.01% to 48.72%, reflecting the increased complexity of age estimation. Finally, for emotion recognition the baseline achieves an accuracy value of 66.57%. However, it is important to note that even these modest out-of-the-box scores for age and emotion are still significantly better than random guessing (which would be $\approx 11.1\%$ for age group classification and $\approx 14.2\%$ for FeR), indicating that we start from a solid baseline.

In terms of computational efficiency, the baseline uses the standard classification pipeline used by CLIP's models; it requires both the textual encoder to process the hard prompts and the visual encoder to process the image.

4.3 Single-task results

As we discussed in the chapters on methodology, each experiment will be tried in both a single-task setting and in a multi-task settings. In this section we report the result obtained for the single-task approach.

Table 4.4: Model Accuracy (%) on the Emotion Task

Model	RAF-DB
ZS	66.57
LP	84.32
AP	84.91
FT_4	88.78
DoRA	**90.83**

Table 4.5: Model Accuracy (%) on the Age Task

Model	UTKFace	FairFace	VggFace2	Average*
ZS	48.62	46.11	42.01	47.36
LP	61.56	61.69	57.75	61.28
AP	61.41	61.75	57.80	61.47
FT_4	63.11	62.50	59.78	62.80
DoRA	**63.80**	**63.64**	**61.24**	**63.72**

*Average calculation excludes the VggFace2 dataset as its age-labels data are synthetically obtained.

Table 4.6: Model Accuracy (%) on the Gender Task

Model	UTKFace	FairFace	VggFace2	Average
ZS	96.63	97.60	95.78	96.67
LP	97.00	97.70	97.92	97.54
AP	96.92	**97.74**	97.98	97.55
FT_4	**97.02**	97.71	97.99	**97.57**
DoRA	96.97	97.72	**98.01**	97.56

4.3.1 Gender Classification

Our single-task gender classification models show an improvement compared to the zero-shot baseline, in fact each model outperform this method by around 0.9%. Linear probing, the approach that modifies less parameters achieves an avarage value across datasets of 97.54%. In comparison, the deeper models outperform it of only 0.01, for attention probing, 0.03 for partial fine-tune and of 0.02 for DoRA, all the while trading top-spot by small margin on perfomance on single benchmark (each of the deeper model achieves best perfomance on one the test-sets). From the fact that there is a really small fork of perfomance, of just 0.03%, we can safely assume that the gender task, in a single-task classification environment, does not benefit from deeper fine-tuning, and we can consider the improvement from linear-probing to partial fine-

tuning non significant, as the model saturates its performance capabilities on this specific task with its MLP classification head.

This plateau suggests that the features required for accurate gender classification are already robustly encoded in the pre-trained model's representations, as it's likely that during the robust image pre-training, it has seen many image-text pair that contained strong and frequent correlations between visual depictions of people and gendered terms (man, woman, girl, boy...) in the accompanying text.

Given these results, linear probing presents the optimal trade-off between performance and efficiency. The marginal, and likely statistically insignificant, gains from deeper methods (like partial fine-tuning or DoRA) do not justify the substantial increase in training time and in the number of modified parameters. We conclude that for this single-task problem, the knowledge is already well-contained within the frozen backbone, and more complex adaptation strategies yield diminishing, negligible returns.

4.3.2 Emotion Classification

For the emotion task, all adaptation methods show a very large improvement over the zero-shot (ZS) baseline of 66.57%. However, unlike the gender task, performance does not saturate with simple linear probing. Linear probing (LP) achieves 84.32%, an important improvement of 17.75% with attention probing (AP) offering a minor improvement at 84.91%, that is in line with the benchmark results provided also in the perception encoder paper, where attention probing yield marginal improvement respect to the linear probing baseline. It is important to note that this 17.75% gain, is not due to the added capacity of the MLP head: it is clear that the 1 million added parameters are not creating new visual understanding. Instead, the MLP head functions as an effective probe into the feature space of the frozen vision encoder. This 17.75% gain is attributable to this head being explicitly trained to find and isolate the specific, pre-existing representations within the backbone that are relevant to the emotion task, as the pre-trained model has already learned a rich visual representation, that we cannot access it with simple zero-shot classification with hard-prompting. Said this, the emotion task clearly benefits from deeper adaptation methods. There is a significant performance jump when moving from shallow probing (LP at 84.32% and AP at 84.91%) to methods that actually alter the backbone's representations, like FT_4 (88.78%) and DoRA (90.83%). This shows that the pre-trained features are not sufficient and can benefit from adaptation. However, the results show that performance does

not scale with the raw number of modified parameters. While partial fine-tuning (FT_4) modifies a large portion of the model (20.13% of parameters), it is outperformed by DoRA, which modifies more than half the number of parameters (8.47% of parameters). This is a key insight: DoRA provides a more efficient and effective adaptation than partial fine-tuning. It achieves the best result (90.83%) by a clear margin, suggesting that refining the model's representations at all levels of the feature hierarchy, trough small, low-rank weights, is more effective than only re-training the final high-level representations (FT_4). Given the substantial 6.51% performance gap between the simple linear probing and DoRA, deeper adaptation is clearly justified and we can conclude that DoRA is the superior method for this single-task problem, offering the best performance with the highest parameter efficiency.

4.3.3 Age Group Classification

For the age group task, all adaptation methods similarly show a massive improvement over the zero-shot (ZS) baseline average of 47.36%. Linear probing (LP) achieves an average accuracy of 61.28%, representing a 13.92% gain over ZS, demonstrating that the MLP head is effectively probing relevant features. Attention probing (AP) provides only a marginal improvement, reaching 61.47%.

The age task shows a trend that is similar to the one shown by the emotion tasks. In fact, similar to emotion, this task clearly benefits from deeper adaptation methods. We see a consistent performance increase when moving from shallow probing to methods that alter the backbone: FT_4 achieves an average of 62.80%, and DoRA achieves the best performance across all three individual datasets, culminating in the highest average of 63.72%. While the 2.44% average gain from LP to DoRA is not as big as the gain seen in the emotion task, it is still a significant and meaningful improvement, in contrast to the negligible gains observed in the saturated gender task. This smaller margin of improvement is likely also a reflection of the task's inherent difficulty, as classifying age groups is a more challenging problem than classifying emotion, and the model may be approaching the limits of achievable accuracy on these benchmarks, with the current setups. Despite this, the results show that the pre-trained features for age are good but not fully optimized, and that refining the backbone is necessary to achieve the best results. Once again, DoRA outperforms FT_4 (63.72% vs. 62.80%) while modifying fewer parameters, reinforcing the finding from the emotion task: a model-wide refinement of representations across all layers, using low-rank updates, is a more effective and efficient strategy than only retraining

the final high-level layers (FT_4). Given these results, the additional complexity of deeper adaptation is justified, and we can conclude that DoRA is the superior method for this single-task problem.

4.4 Multi-task results

4.4.1 Exponential moving average or uncertainty weighting

As explained in 3.7.3, we presented two way of balancing the three task losses. To limit the number of experiments, we will test the two methods by training two model with partial fine-tuning and MTLoRA.

Table 4.7: Model Accuracy (%) on the Emotion Task

Model	Emotion Acc	Avg. Age Acc	Avg. Gender Acc	Overall Avg
FT_4 EMA	86.47	62.96	97.44	82.29
FT_4 UW	88.43	63.00	97.41	82.94
MTLoRA EMA	88.98	63.29	97.51	83.26
MTLoRA UW	90.06	64.03	97.51	83.86

From table 4.7, we can see that UW outperforms EMA, even if slightly. In fact, the two methods performed quite similarly, analyzing the training logs reveals that both method under weighted the loss of the age the task, and over weighted both the loss of the emotion and gender task. Both tried to strike a balance in this manner, as the training loss for the age task resulted in both context an order of magnitude greater than the other two. The key difference seems to have been that, EMA severely under weighted the age task loss, and slightly over weighted the other two, while UW slightly under weighted the age loss and greatly over weighted the other two. Going forward, the reported multi-task model will have been trained using UW.

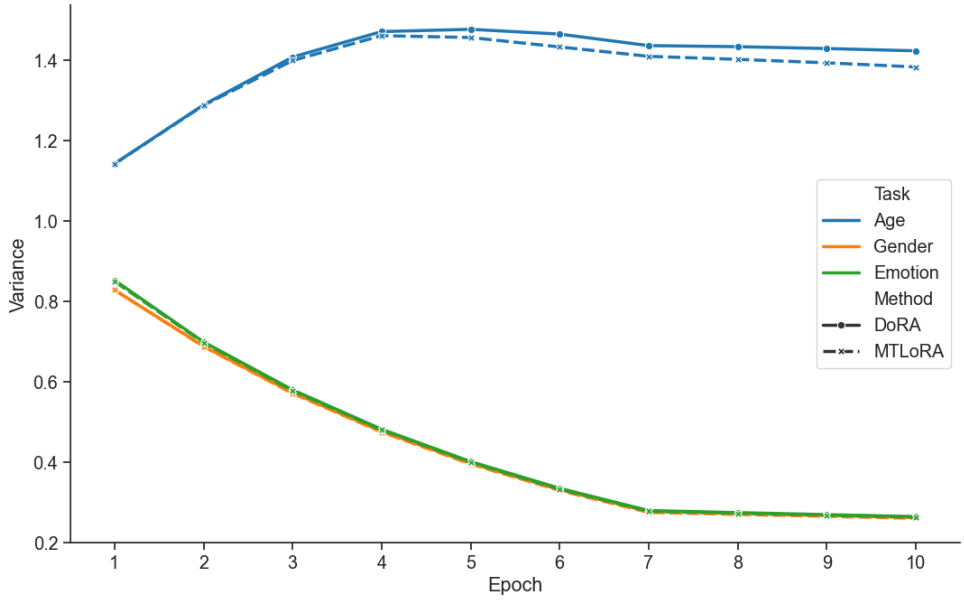


Figure 4.1: Variances obtained by MTLoRA and DoRA, the model converge to similar weighting for the three task

Results

We also repeat the linear probing results in this section. The reason is that since all three probing heads use the same frozen backbone, they can easily be loaded together onto that single encoder. This allows for the assembly of a multi-task network, even though the probes were trained individually and not through multi-task learning.

Table 4.8: Model Accuracy (%) on the Emotion Task

Model	RAF-DB
LP	84.82
AP	85.12
FT_4	88.42
DoRA	**91.21**
MTLoRA	90.06

Table 4.9: Model Accuracy (%) on the Age Task

Model	UTKFace	FairFace	VGGFace2
LP	61.56	61.00	57.75
AP	63.04	61.89	59.94
FT_4	62.54	63.45	61.68
DoRA	63.34	63.73	61.69
MTLoRA	**63.96**	**64.11**	**62.74**

Table 4.10: Model Accuracy (%) on the Gender Task

Model	UTKFace	FairFace	VGGFace2
LP	97.00	97.70	97.92
AP	96.79	97.63	97.79
FT_4	96.68	**97.71**	97.81
DoRA	96.90	97.57	**98.00**
MTLoRA	**96.93**	97.62	**98.00**

Table 4.11: Summary of Model Average Performance in Single-Task (%)

Model	Emotion Acc	Avg. Age Acc	Avg. Gender Acc	**Overall Average**
ZS	66.57	47.36	96.67	70.20
LP	84.32	61.28	97.54	81.05
AP	84.91	61.47	97.55	81.31
FT_4	88.78	62.80	97.57	83.05
DoRA	90.83	63.72	97.56	**84.04**

Table 4.12: Summary of Model Average Performance in Multi-Task (%)

Model	Emotion Acc	Avg. Age Acc	Avg. Gender Acc	**Overall Avg**
ZS	66.57	47.36	96.67	69.61
LP	84.91	61.28	97.54	78.76
AP	85.12	62.46	97.40	81.66
FT_4	88.43	63.00	97.41	82.94
DoRA	91.21	63.53	97.49	**84.07**
MTLoRA	90.06	64.03	97.51	83.86

4.4.2 Gender Classification

In the multi-task setting, all adaptation methods again outperform the zero-shot (ZS) baseline of 96.67%. However, the results reinforce the conclusion from the single-task analysis: the gender task is saturated and does not benefit from complex adaptation.

The performance across all multi-task models is compressed with a total range of only 0.09% (from 97.40% for AP to 97.49% for DoRA and MTLoRA). Deeper methods like FT_4 (97.41%), DoRA (97.49%), and MTLoRA (97.51%) show no meaningful advantage over the shallow attention probing approach.

Comparing these results to their single-task counterpart, we observe that multi-task learning provides no benefit in this case. For instance, single-task DoRA achieved 97.56%, whereas

multi-task DoRA scored 97.49%. In fact, the linear probing (LP) model (97.54%), which simply probes the original, unadapted features, performs better than every single multi-task adapted model (AP, FT_4, DoRA, and MTLoRA). This suggests that the features from the frozen vision encoder, as already observed, are already highly discriminative for the gender task. Consequently, the multi-task optimization for the more complex emotion and age tasks introduces a minor negative transfer, slightly pulling the shared representation away from the optimal space for the simpler gender task.

MTLoRA performs on par with the multi-task DoRA approach, with a marginal improvement of 0.02%. The specialized method was not able prevent this marginal negative transfer that we have observed, performing still slightly worse than the single-task methods (still noting that we are talking about a really small difference).

4.4.3 Emotion Classification

For the emotion task in a multi-task environment, the shallowest adaptation method is Attention Probing (AP), which achieves 85.12%. This is a massive improvement over the ZS baseline (66.57%) and is also a notable step up from the LP approach that achieves 84.32% as accuracy. As with the single-task case, this demonstrates that while probes can find relevant features, performance does not saturate with shallow methods. Moreover, it also shows that MTL does not hinder emotions, as it even perform better than its single-task equivalent, that achieves a score of 84.91%

A significant performance jump is seen with deeper adaptation: FT_4 reaches 88.43%, and DoRA achieves 91.21%. This confirms that adapting the backbone's representations is crucial for this task.

The most important insight comes from comparing single-task and multi-task results. The DoRA approach benefits from multi-task learning, improving from 90.83% to 91.21% . This indicates positive knowledge transfer, where learning to classify age and gender simultaneously provides a regularizing effect that enhances the model's understanding of emotion.

However, the specialized MTLoRA, which uses task-specific matrices on the final block, performs worse than standard DoRA (90.06% vs. 91.21%) and even worse than single-task DoRA (90.83%). This suggests that for the emotion task, sharing the DoRA updates across all layers is more effective. The task-specific specialization in the final layer seems to hinder, rather than help, this particular task, as it annuls the positive knowledge transfer that we see with task

agnostic approach.
Therefore, for emotion classification, deeper adaptation is essential, as already shown for single-task, and DoRA in a multi-task setting provides the optimal result, leveraging positive knowledge transfer.

4.4.4 Age Group Classification

For the age group task, all adaptation methods similarly show a massive improvement over the zero-shot (ZS) baseline average of 47.36%. The shallowest multi-task method, Attention Probing, achieves 62.46%. This is an interesting result: while in the single-task setting, AP (61.47%) offered a negligible gain over Linear Probing (61.28%), here in the multi-task setting, it shows a clear benefit. This positive knowledge transfer demonstrates that jointly training the shared attention pooling layer for all three tasks improves its ability to aggregate features.
Similar to the emotion task, this task benefits from deeper adaptation methods. We see a consistent performance increase when moving from AP (62.46%) to methods that alter the vision encoder transformers blocks: FT_4 achieves 63.00%, and DoRA achieves 63.53%. This confirms that the pre-trained features are not fully optimized and that refining the backbone is necessary.
A crucial insight appears when comparing standard DoRA across settings. We observe a minor case of negative transfer: single-task DoRA (63.72%) actually outperforms the multi-task standard DoRA (63.53%). This suggests that a fully shared adaptation forces a small representational compromise that hinders the age task.
MTLoRA, in this situations, works as expected. By utilizing task-specific LoRA matrices in the final transformer block, MTLoRA achieves an average accuracy of 64.03%. This is the highest performance for the age task across all models and both tables. It successfully overcomes the negative transfer seen in the standard DoRA model, demonstrating that the age task benefits from both shared representation learning in early layers and dedicated, specialized high-level representations.

4.4.5 Singe-Task vs. Multi-Task

The multi-task learning framework demonstrates considerable solidity and robustness across all adaptation methods, from the shallowest (AP) to the deepest (DoRA). A primary concern in multi-task learning is negative transfer, where joint optimization degrades performance on individual tasks. The results show that this framework successfully avoids this possible deficit

of MTL, achieving a global performance on par with the collection of specialized single-task models that we trained.
This robustness is immediately evident when comparing the 'Overall Average' performance of each multi-task model against its single-task counterpart:

- Attention Probing (AP) sees a clear gain of +0.35% (81.66% vs 81.3%).
- Partial Fine-Tuning (FT_4) sees a negligible drop of just -0.11% (82.94% vs 83.05%).
- DoRA sees a negligible gain of +0.03% (84.07% vs 84.04%).

A task-specific analysis of the top-performing DoRA framework confirms this balance. The standard MTL DoRA (84.07%) and its single-task counterpart (84.04%) achieved nearly identical accuracy. The MTL model yielded clear positive transfer for emotion classification ($+0.38\%$), offset by only minor, negligible degradation in age (-0.19%) and gender (-0.07%). This suggests the joint optimization provides a regularizing effect, as the shared DoRA adapter weights capture complementary information.
This performance parity suggests that the multi-task DoRA framework largely avoids significant negative transfer. The negligible degradation in age and gender, combined with the clear positive transfer in the emotion task, indicates that the shared DoRA adapter weights capture complementary information. It appears the joint optimization provides a regularizing effect that benefits emotion recognition, while only minimally impacting the other tasks.
This confirms the stability of the multi-task approach and the frameworks ability to maintain performance, regardless of the adaptation method.
Finally, the MTLoRA experiment employed a more complex architecture designed to create task-specific representations. It achieved an overall average performance of 83.88%, a strong result that surpassed shallower approaches like AP (81.66%) and FT_4 (82.94%). However, it fell slightly short of the standard multi-task DoRA model (84.07%). This indicates that, from a global performance standpoint, the simpler, fully-shared parameterization of standard DoRA offered a marginally better overall balance than the specialized MTLoRA architecture, while also providing advantages in inference speed and memory efficiency.

4.5 Comparison to the state of the art

In this section, we report a comparison of our best-performing models against established state-of-the-art (SOTA) methods from the literature. It is crucial to contextualize this comparison, as a direct evaluation is complicated by key differences in methodology. Firstly, while we evaluate on the same standard datasets, the specific training and testing splits used in our work may not be identical to those employed by all published SOTA models (e.g. for UTKFace we utilize the entire dataset for testing, and not a split). Secondly, our entire pipeline includes a mandatory face cropping step prior to classification (as detailed in Section 2.7), where the face is first detected and isolated from the image. This preprocessing step may differ from other SOTA methods that might operate on the full, uncropped image or use different alignment techniques. Despite these variations, the following comparison provides a valuable reference for positioning our results within the broader research landscape for facial attribute analysis.

	Age (Acc. %)		**Gender (Acc. %)**		**Emotion (Acc. %)**
Method	**FairFace**	**UTKFace**	**FairFace**	**UTKFace**	**RAF-DB**
SOTA (Age/Gender Focused)					
MIVOLO$_{224}$ [21]	61.07	3.7 MAE	95.73	**98.84**	-
MIVOLO$_{384}$ [22]	62.28	...	97.5	...	-
CLIP ViT-L/14 336px*[14]	63.45	...	97.1	...	-
SOTA (Emotion Focused)					
ResEmoteNet [18]	-	-	-	-	**94.76**
APViT [19]	-	-	-	-	92.21
POSTER++ [20]	-	-	-	-	91.98
Our Models					
MTLoRA	**64.11**	63.96	**97.62**	96.93	90.06
LoRA	63.73	63.34	97.57	96.90	91.21

Table 4.13: Performance comparison with state-of-the-art (SOTA) methods for age, gender, and emotion recognition on standard benchmarks.

* The original CLIP paper reports accuracy disaggregated by ethnicity ('White' and 'Other'). The values shown are the average of these two groups (Age: 63.8% and 63.1%; Gender: 96.5% and 97.7%).
'-' Indicates the method is not designed for this task.
'...' Indicates the result was unavailable.

As we can see our models are competitive with SOTA solutions: in the age-task we achieve SOTA perfomances, for the FairFace dataset. For the FairFace dataset we also maintain a top-spot for the gender-recognition problem, while for UTKFace we under perform with our multi-task model, by 1.9%. This discrepancy may be due to the fact that we are evaluating our

performances on the entire dataset and not only the test-split. Moreover, we may also keep in considerations as a possible source of error our face recognition error, as we may have faulty detection, or detection that do not match with the associated label, in the case of pictures containing more than one individual.

The most significant performance gap is observed in the FeR task, where our model trails the top-performing ResEmoteNet by 3.55%. As noted in the literature, ResEmoteNet is a convolutional based neural network trained from scratch specifically for FeR. CNNs possess inherent inductive biases, such as parameter sharing and local receptive fields, which make them highly effective for training from scratch on smaller datasets like RAF-DB (approx. 12k samples).

In contrast, the gap is much smaller when compared to methods using pre-trained backbones, such as Poster++ (1% gap) and ApViT (0.77% gap). These two methods employ backbones pre-trained on specialized, face-centric datasets. Our model's perception encoder, however, was pre-trained on more generalist datasets and objectives. This difference in pre-training specialization likely accounts for the performance discrepancy.

It is important to note, however, that when evaluating using balanced accuracy, our model's performance is on par with these competitors, as detailed in Section 4.7.

4.6 Efficiency comparison

Table 4.14: Model Size and Computational Cost, and Average Accuracy on Multi-Attribute Tasks

Model	Params (M)	GFLOPs	Avg. Acc
ZS	671	699.76	69.61
FT_4	320	352.18	80.56
DoRA (merged)	320	352.18	84.07
MTLoRA	329	368.01	83.89

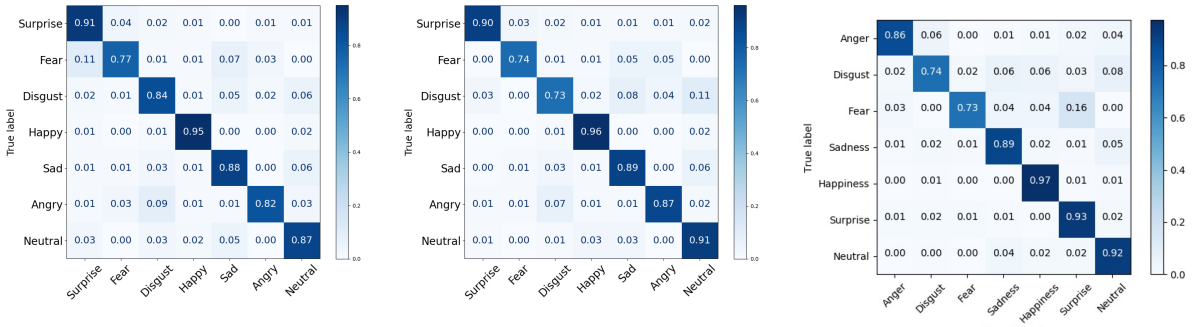
One of our primary goals was to develop a framework that is significantly more efficient, in terms of parameters and inference, than the zero-shot baseline approach. The results presented in Table 4.14 confirm that we have successfully achieved this objective. As the table illustrates, all proposed approaches substantially outperform the zero-shot baseline in efficiency. By discarding the textual encoder, we cut the memory footprint and associated computational load by nearly half, reducing the GFLOPs from 699.76 to 352.18. This drastic reduction in resources

does not come at the cost of performance. On the contrary, all our proposed adaptation strategies also greatly outperform the zero-shot baseline's accuracy. In our analysis of MTLoRA, we observed that a 2.8% increase in parameters and a 4.49% increase in GFLOPs (compared to the standard DoRA adaptation) did not yield a corresponding increase in average accuracy. Finally, we could have considered an alternative: loading three separate, task-specific DoRA adapters. This approach, in its simpler form, would involve sequentially switching between adapters and is computationally prohibitive (around 1400 GFLOPs and 400M parameters). Moreover, our work demonstrated that our single multi-task DoRA adapter achieves accuracy on par with this inefficient, multi-adapter method. Therefore, we did not explore advanced parallel serving methodologies that could bring up the computational efficiency of this method, as the ones that we have cited in the literature review (S-LoRA, B-LoRA).

4.7 Balanced Accuracy

In the interest of fairness, in this chapter we will report the balanced accuracy obtained by our multi-task models in the age and emotion tasks, as they are the task with a strong class imbalance, we will also report the obtained confusion matrix to further examine the model predictive behavior.

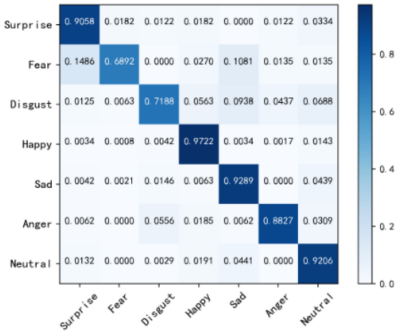
4.7.1 Emotion Recognition confusion matrices



(a) MTLoRA confusion matrix on RAF-DB, balanced accuracy of 86.17 (Acc. 90.06)

(b) DoRA confusion matrix on RAF-DB, balanced accuracy of 85.90 (Acc. 91.21)

(c) APViT confusion matrix on RAF-DB, balanced accuracy of 86.36 (Acc. 92.21)

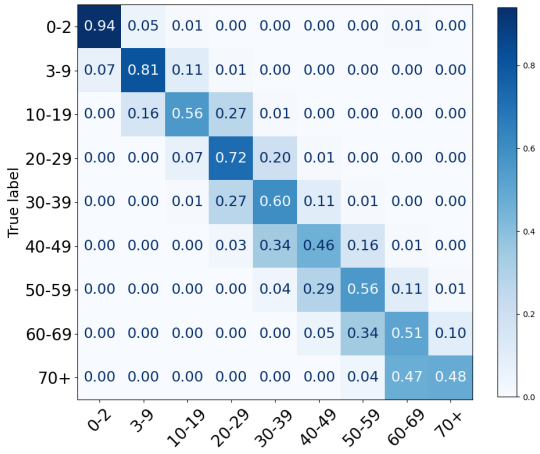


(d) Poster++ confusion matrix on RAF-DB, balanced accuracy of 85.97 (Acc. 91.98)

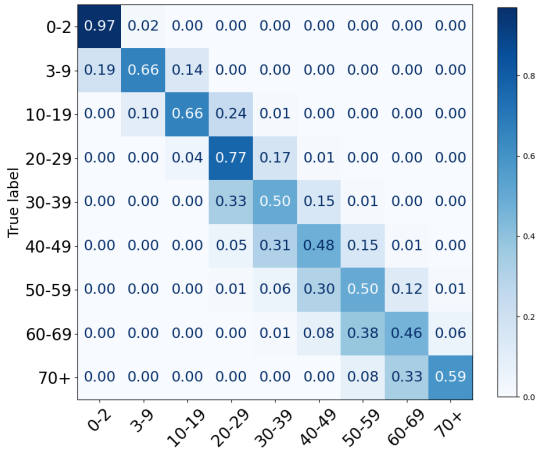
Figure 4.2: Confusion matrices on the RAF-DB dataset, with APViT and Poster++ reported for comparison.

Based on the confusion matrices 4.2, our multi-task model's performance is highly comparable to that of APViT and Poster++, with a difference in balanced accuracy of less than 0.2 percentage points for our MTLoRA model and of 0.46 for the DoRA model compared to ApViT. The comparison with Poster++, is even more favorable, with the MTLoRA model surpassing it in this metric by 0.2% and DoRA lagging behind by just 0.07%. The matrices also show that the models perform similarly, struggling most with the "fear" class, which they often misclassify as "surprise." Both models also have difficulty identifying the "disgust" and "anger" classes. This behavior is likely due to the visual similarity between fear and surprise, and the fact that the anger and disgust classes have fewer samples compared to the others.

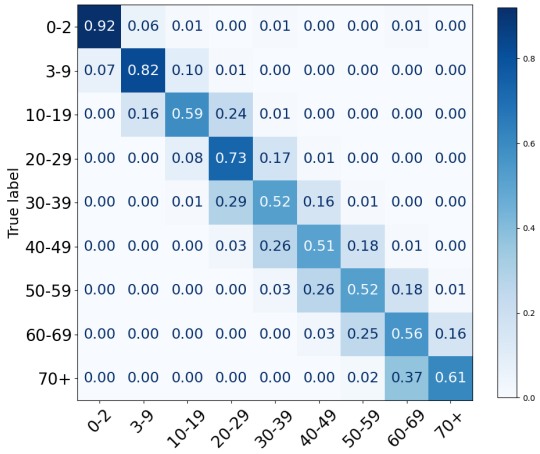
4.7.2 Age classification confusion matrices



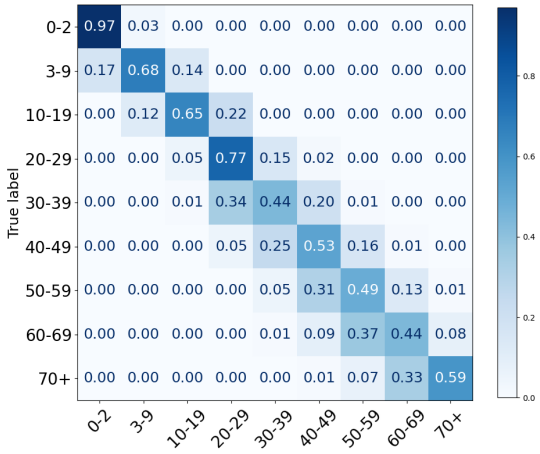
(a) MTLoRA confusion matrix on FairFace, balanced accuracy of 62.78 (Acc. 64.11)



(b) MTLoRA confusion matrix on UTKFace, balanced accuracy of 61.98 (Acc. 63.96)



(c) LoRA confusion matrix on FairFace, balanced accuracy of 64.26 (Acc. 63.73)



(d) LoRA confusion matrix on UTKFace, balanced accuracy of 61.91 (Acc. 63.34)

Figure 4.3: Comparison of age confusion matrices across FairFace and UTKFace.

As we can observe from the matrices 4.3, the models produced show variable performance that are strongly dependent on the age group. It is quite accurate at identifying young individuals, but its precision decreases when classifying adults, especially from middle age onward. This may be explained by the high intra-class variance within adult age groups and the low inter-class variance separating adjacent ones.

An observation that can be made is that the model indirectly "understands" the ordinality of the task, in fact almost all significant errors are "off-by-one" misclassifications into an adjacent age bracket and there are virtually no "severe" errors (e.g., classifying "0-2" as "50-59"). This indicates the model is good at estimating age, but struggles with the precise boundaries of the 10-year bins.

In conclusion, the analysis of the balanced accuracy evidences a particular trend, where a high standard accuracy does not necessarily translate to an equally high balanced accuracy, in fact, accounting for this metric results in a swap of the leading models, identifying MTLoRA as the top performer for FeR and the LoRA model as the best for age classification.

C-Index

Task	LP vs. ZS	LoRA vs. ZS	LP vs. LoRA
FairFace Age	56.13%	53.45%	81.89%
FairFace Gender	98.69%	98.30%	98.79%
RAF-DB Emotion	69.29%	66.75%	85.22%

Table 4.15: Concordance of prediction between zero-shot baseline, linear probe and LoRA.

This table presents the Concordance Index (C-Index), which measures the percentage of agreement between the predictions of three of the model produced, and is calculate as such:

$$C = \frac{\text{Number of equal predictions}}{\text{Total number of samples}} \times 100\%$$

We can see as for the gender task, there is almost no real "disagreement" as expected, as the perfomance of the baseline was already quite strong. For age and emotion instead, we can see how there is a low-agreement between the baseline and the trained model. Crucially, LP and LoRA show an high agreement with each other, indicating similar predictive patterns.

DoRA and LoRA, small ablation experiment

The DoRA method has been introduced by its author as a way to more closely match the training pattern of full-fine tune for the LoRA adaptation method. As we have noted that a full-fine tune is not necessarily something we desire, it's fair to doubt the effectiveness of DoRA in our setting where full-fine tune may lead to overfitting. This is also an observation done by the authors of the DoRA paper, and they examine in section 5.3 [28] a similar situation, where they have a FT

model that performs worse than it's LoRA adaptation. In their setting DoRA still outperformed LoRA, by a smaller margin. To see if we are in the same case, we ran an ablations, by training a MTL LoRA model, without the DoRA enhancements (but still using the LoRA+ learning rate scheme), using the same hyperparameters detailed in the methodology section.

Model	Avg Age Acc.	Avg Gender Acc.	Emotion Acc.
LoRA	63.45%	97.52%	90.44%
DoRA	63.53%	97.49%	91.21%

Table 4.16: LoRA and DoRA comparison.

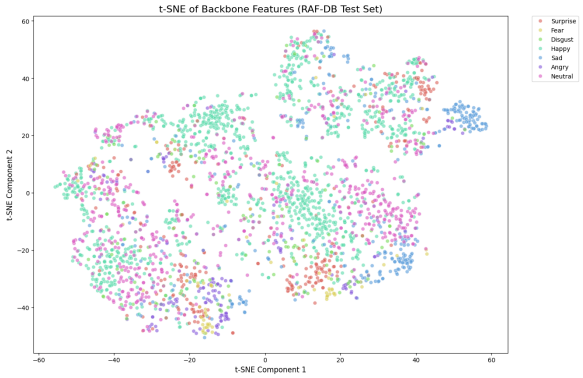
The results of this ablation (Table 4.16) in our multi-task learning setting are consistent with the findings reported by the DoRA authors.

The primary takeaway is that DoRA provides a clear and significant performance boost on the Emotion task, outperforming the standard LoRA by +0.77% (91.21% vs. 90.44%). This is the most impactful result and suggests that for complex, nuanced tasks that require substantial adaptation, DoRA's method of separating magnitude and direction provides a more effective update than LoRA alone. This trend is mirrored, though to a much smaller degree, on the Age task, which also benefits from deeper adaptation. Here, DoRA provides a marginal +0.08% improvement. Conversely, on the gender task, which we previously established as being "saturated" and not benefiting from deeper adaptation, the methods are statistically identical. The negligible -0.03% difference confirms that this simple task is insensitive to the nuances between these two advanced adaptation methods.

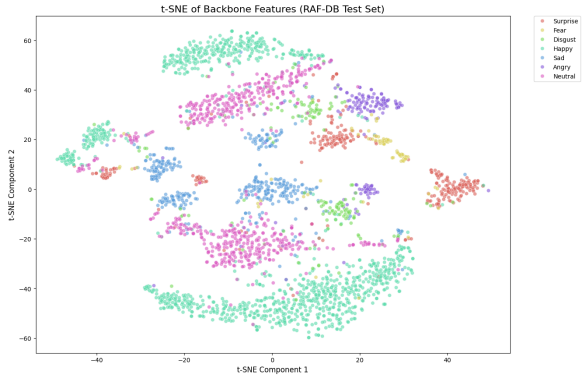
4.8 T-SNE and PCA visualizations

In this section, we present visualizations of the RAF-DB and UTKFace datasets using principal component analysis (PCA) and t-distributed stochastic neighbor embedding (t-SNE), applied to the 50 principal components obtained from PCA. We present the out-of-the box PE-Core-L visualization, the DoRA MTL model visualizations and the MTLoRA visualization. For this last one, we can observe the main effect of this methodology: producing a task-specific feature map per task.

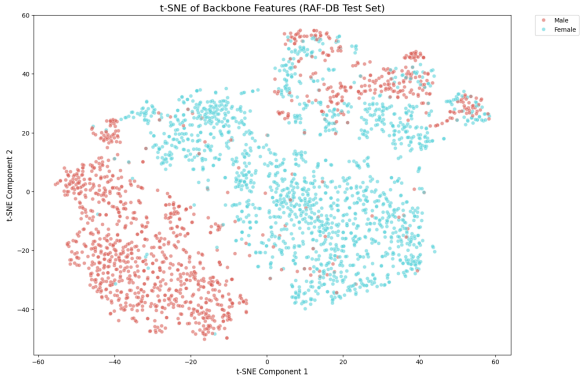
Male Female Surprise Fear Disgust Happy
Sad Angry Neutral 0-2 3-9 10-19
20-29 30-39 40-49 50-59 60-69 70+



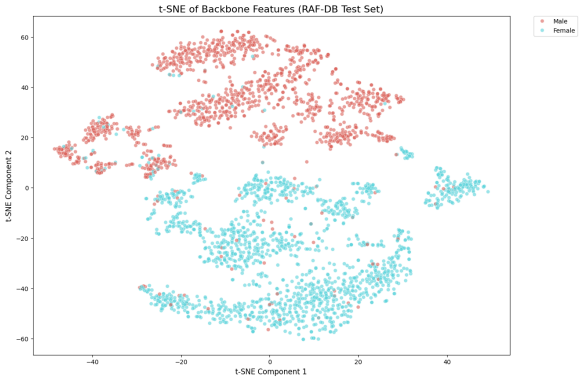
(a) t-SNE Emotion (Untrained)



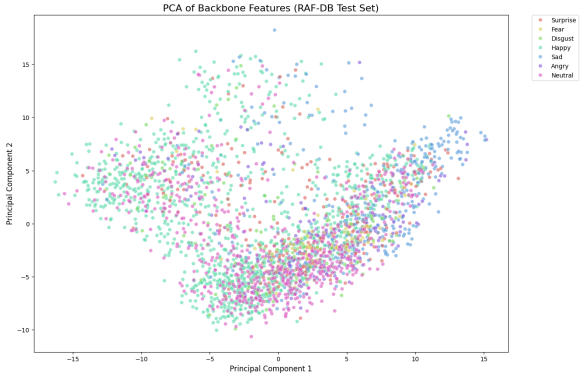
(b) t-SNE Emotion (Trained)



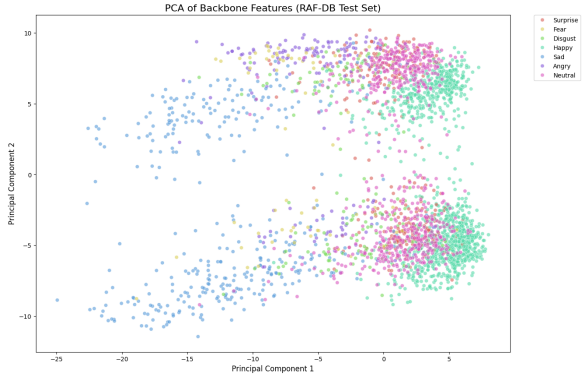
(c) t-SNE Gender (Untrained)



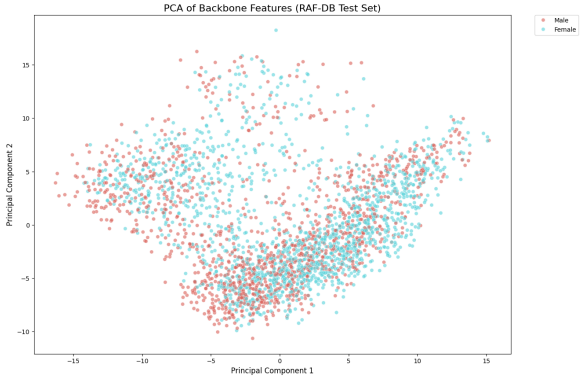
(d) t-SNE Gender (Trained)



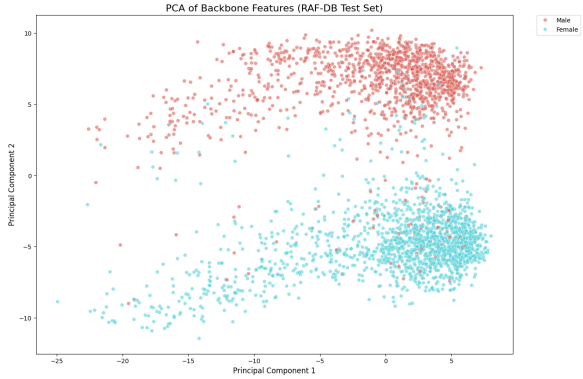
(e) PCA Emotion (Untrained)



(f) PCA Emotion (Trained)

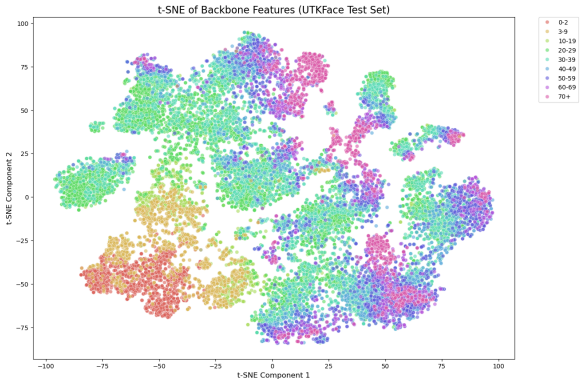


(g) PCA Gender (Untrained)

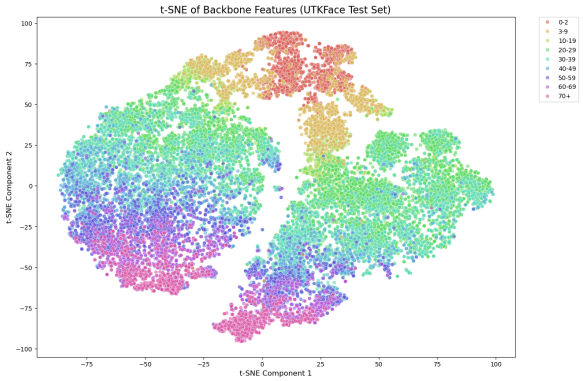


(h) PCA Gender (Trained)

Figure 4.4: Comparison of untrained and trained model features on RAF-DB, for the DoRA MTL model.

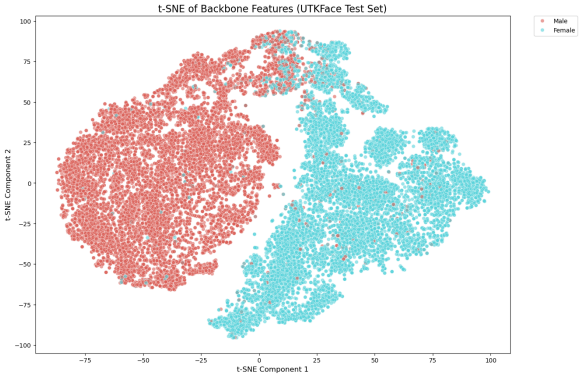


(a) t-SNE Age (Untrained)

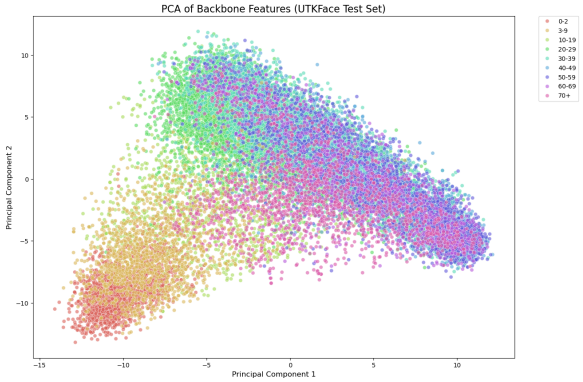


(b) t-SNE Age (Trained)

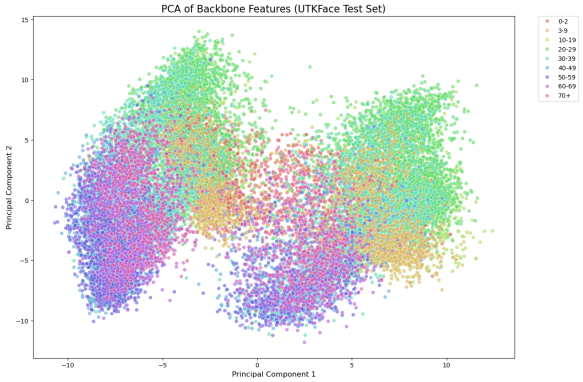
(c) t-SNE Gender (Untrained)



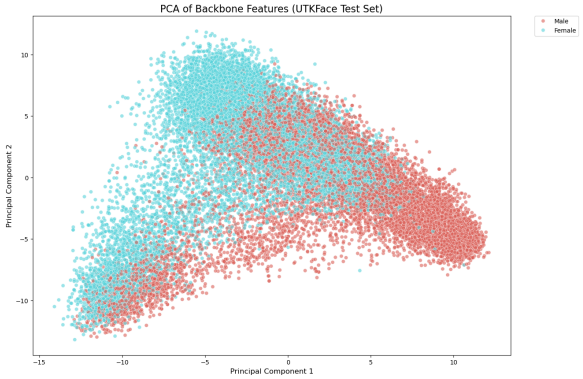
(d) t-SNE Gender (Trained)



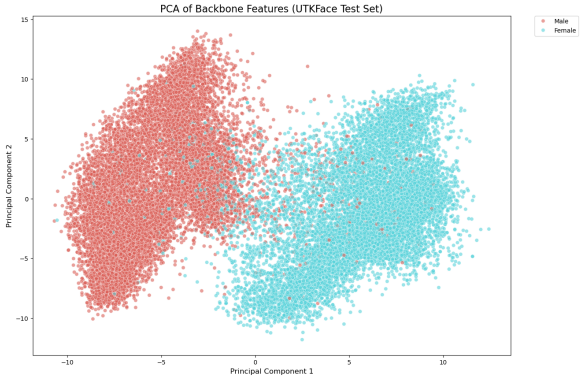
(e) PCA Age (Untrained)



(f) PCA Age (Trained)

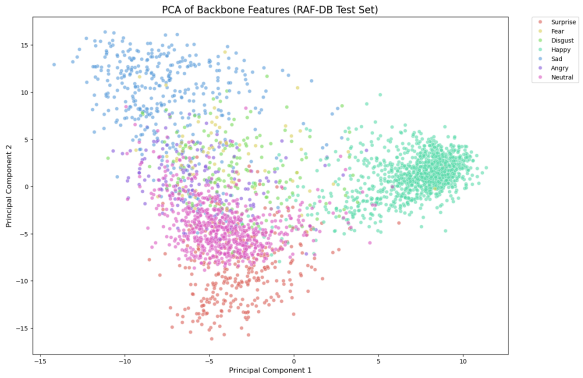


(g) PCA Gender (Untrained)

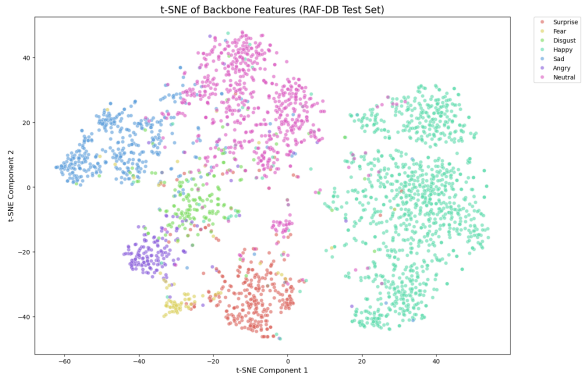


(h) PCA Gender (Trained)

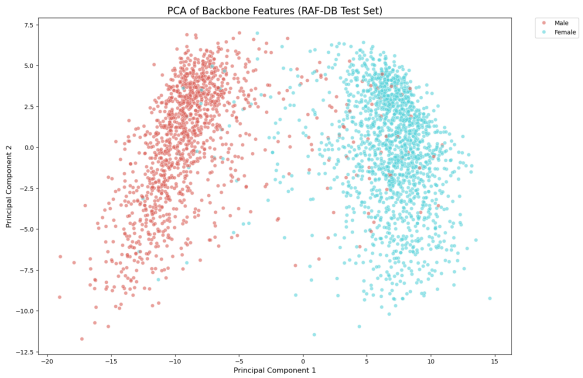
Figure 4.5: Comparison of untrained and trained model features on UTKFace, for the DoRA MTL model.



(a) PCA Emotion



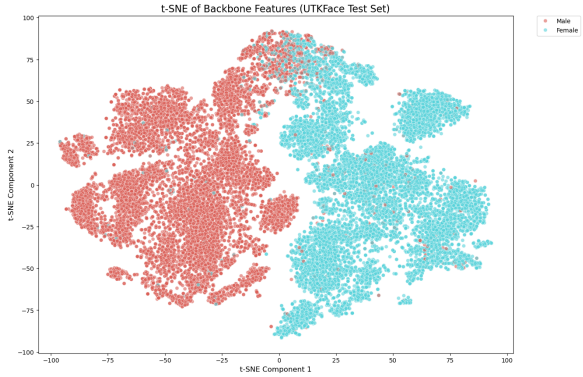
(b) t-SNE Emotion



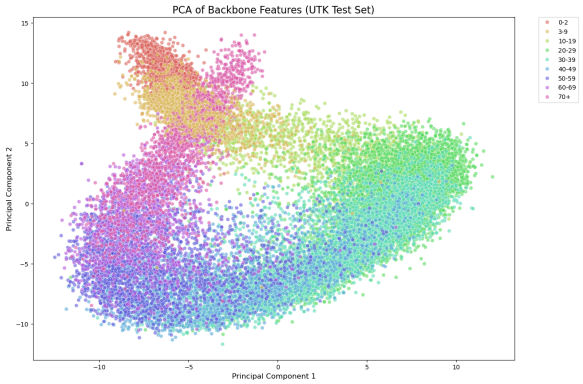
(c) PCA Gender

(d) t-SNE Gender

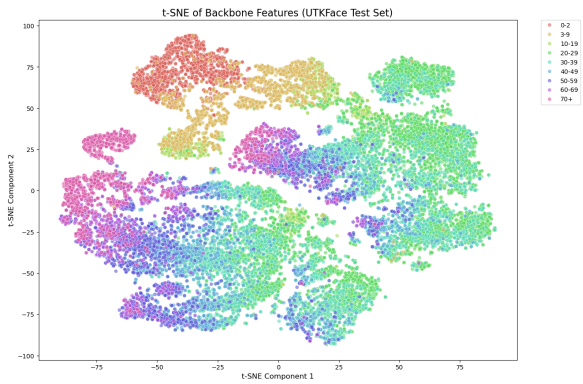
(e) PCA Gender



(f) t-SNE Gender



(g) PCA Age



(h) t-SNE Age

Figure 4.6: Visualization of MTLoRA features. We can appreciate how each task has its own representation

Chapter 5

Conclusion

5.1 Analysis of findings

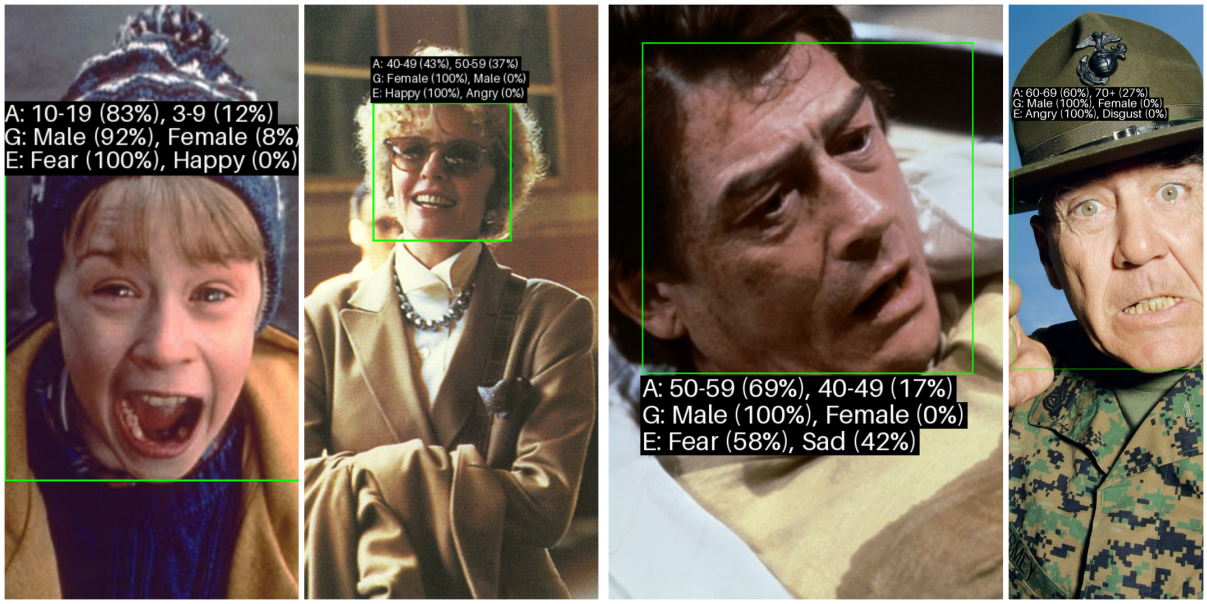


Figure 5.1: Example of MTLoRA predictions

The experimental results presented in Chapter 4 confirm the central hypothesis of this thesis: a large-scale, pre-trained VLM vision encoder can serve as an exceptionally effective foundation for the specialized domain of multi-task facial attribute classification. A primary finding is the inadequacy of the zero-shot (ZS) baseline: while the ZS model, which requires both the visual and text encoders, showed modest capability (69.61% overall average), every adaptation method significantly outperformed it. More importantly, by adapting only the vision encoder, we developed models that are far more efficient. All proposed solutions discard the text

encoder, effectively halving the computational load from 699.76 GFLOPs to ≈350 GFLOPs, while simultaneously achieving substantial accuracy gains. Among the adaptation strategies, the Parameter-Efficient Fine-Tuning approach using LoRA (with DoRA and LoRA+) emerged as the optimal solution: in the single-task setting, LoRA achieved the highest accuracy on the two most complex tasks, Age and Emotion, notably, it outperformed partial fine-tuning (FT_4) while training less than half the number of parameters (8.47% vs. 20.13%), demonstrating that a parameter-efficient approach can be a more effective strategy, that mitigates overfitting and better preserves the generalized knowledge of the pre-trained backbone for modest-size datasets.

The multi-task learning (MTL) experiments revealed a trade-off: conjoint training consistently benefited the most complex task, Age, which saw a performance increase across all MTL models, suggesting the models successfully leveraged shared representations from the Gender and Emotion tasks to improve its understanding of age-related features, showing an example of positive-transfer. However, this came at the cost of a slight performance dip in the Emotion and Gender tasks, indicating a degree of negative transfer for FT_4 and MTLoRA. The standard LoRA model instead provided the best overall compromise, achieving the highest overall average accuracy (84.07%). The more complex MTLoRA, designed to explicitly mitigate this negative transfer, did not yield a superior overall result, suggesting that a single, well-tuned set of LoRA adapters was more effective. In conclusion, this work developed an efficient, multi-task facial attribute classifier by adapting the PE-Core-L vision encoder. The LoRA methodology proved to be the superior adaptation strategy, creating a final model that is computationally efficient (352 GFLOPs) compered to our baseline, parameter-efficient (8.47% trainable parameters), and accurate, achieving SOTA-competitive performance on complex emotion and age classification tasks.

5.2 Future works

This thesis demonstrated the effectiveness of adapting a pre-trained VLM vision encoder for multi-task facial attribute classification. The results, particularly with LoRA, are competitive and highly efficient and opens several promising avenues for future investigation:

- **Compare Vision Encoders:** Evaluate other models from the PE family (e.g., PE-Spatial) and encoders with different pre-training, such as the self-supervised DINOv3.

- **Evaluate Model Scaling:** Test smaller PE variants (Base, Small and Tiny) to obtain "mobile" version of our models, and be able to compare LoRA's efficiency against a feasible full fine-tuning on these models.
- **Reformulate Age Task:** Explore a fine-grained ordinal regression alternative to coarse age group classification, treating each individual year (1, 2, 3... 100+) as a distinct rank and optimizing with a loss like CORAL.
- **Advance Loss Balancing:** Explore more multi-task loss balancing methodologies, such as Dynamic Weight Averaging (DWA) or GradNorm.
- **Dynamic Adapter Serving:** Investigate efficient deployment strategies like S-LoRA to serve the best-performing single-task adapters on a shared backbone, analyzing the GFLOPs and accuracy trade-off.
- **Explore multi-scale MTLoRA:** Implement a similar approach as presented in the MT-LoRA paper, were TS-LoRA are added at various scale and the feature-map at different scale are then fused to produce a task-specific embedding. Moreover explore the possibility of defining the TS-LoRA as TS-DoRA.

List of Tables

List of Figures

Bibliography

[1] Alexey Dosovitskiy et al. *An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale*. 2021. arXiv: 2010.11929 [cs.CV]. URL: https://arxiv.org/abs/2010.11929.

[2] Ashish Vaswani et al. "Attention is All you Need". In: *Advances in Neural Information Processing Systems*. Ed. by I. Guyon et al. Vol. 30. Curran Associates, Inc., 2017. URL: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf.

[3] Jianlin Su et al. *RoFormer: Enhanced Transformer with Rotary Position Embedding*. 2023. arXiv: 2104.09864 [cs.CL]. URL: https://arxiv.org/abs/2104.09864.

[4] Daniel Bolya et al. *Perception Encoder: The best visual embeddings are not at the output of the network*. 2025. arXiv: 2504.13181 [cs.CV]. URL: https://arxiv.org/abs/2504.13181.

[5] Oriane Siméoni et al. *DINOv3*. 2025. arXiv: 2508.10104 [cs.CV]. URL: https://arxiv.org/abs/2508.10104.

[6] Michael Tschannen et al. *SigLIP 2: Multilingual Vision-Language Encoders with Improved Semantic Understanding, Localization, and Dense Features*. 2025. arXiv: 2502.14786 [cs.CV]. URL: https://arxiv.org/abs/2502.14786.

[7] Ze Liu et al. *Swin Transformer: Hierarchical Vision Transformer using Shifted Windows*. 2021. arXiv: 2103.14030 [cs.CV]. URL: https://arxiv.org/abs/2103.14030.

[8] Hugo Touvron et al. *Training data-efficient image transformers and distillation through attention*. 2021. arXiv: 2012.12877 [cs.CV]. URL: https://arxiv.org/abs/2012.12877.

[9] Asifullah Khan et al. *A Survey of the Self Supervised Learning Mechanisms for Vision Transformers*. 2025. arXiv: 2408.17059 [cs.CV]. URL: https://arxiv.org/abs/2408.17059.

[10] Hangbo Bao et al. *BEiT: BERT Pre-Training of Image Transformers*. 2022. arXiv: 2106.08254 [cs.CV]. URL: https://arxiv.org/abs/2106.08254.

[11] Mathilde Caron et al. *Emerging Properties in Self-Supervised Vision Transformers*. 2021. arXiv: 2104.14294 [cs.CV]. URL: https://arxiv.org/abs/2104.14294.

[12] Maxime Oquab et al. *DINOv2: Learning Robust Visual Features without Supervision*. 2024. arXiv: 2304.07193 [cs.CV]. URL: https://arxiv.org/abs/2304.07193.

[13] Asifullah Khan et al. "A survey of the vision transformers and their CNN-transformer based variants". In: *Artificial Intelligence Review* 56.S3 (Oct. 2023), pp. 2917–2970. ISSN: 1573-7462. DOI: 10.1007/s10462-023-10595-0. URL: http://dx.doi.org/10.1007/s10462-023-10595-0.

[14] Alec Radford et al. *Learning Transferable Visual Models From Natural Language Supervision*. 2021. arXiv: 2103.00020 [cs.CV]. URL: https://arxiv.org/abs/2103.00020.

[15] Haotian Liu et al. "Visual Instruction Tuning". In: *Advances in Neural Information Processing Systems*. Ed. by A. Oh et al. Vol. 36. Curran Associates, Inc., 2023, pp. 34892–34916. URL: https://proceedings.neurips.cc/paper_files/paper/2023/file/6dcf277ea32ce3288914faf369fe6de0-Paper-Conference.pdf.

[16] Junnan Li et al. *BLIP-2: Bootstrapping Language-Image Pre-training with Frozen Image Encoders and Large Language Models*. 2023. arXiv: 2301.12597 [cs.CV]. URL: https://arxiv.org/abs/2301.12597.

[17] Cijo Jose et al. *DINOv2 Meets Text: A Unified Framework for Image- and Pixel-Level Vision-Language Alignment*. 2024. arXiv: 2412.16334 [cs.CV]. URL: https://arxiv.org/abs/2412.16334.

[18] Arnab Kumar Roy et al. "ResEmoteNet: Bridging Accuracy and Loss Reduction in Facial Emotion Recognition". In: *IEEE Signal Processing Letters* 32 (2025), pp. 491–495. DOI: 10.1109/LSP.2024.3521321.

[19] Fanglei Xue et al. “Vision Transformer With Attentive Pooling for Robust Facial Expression Recognition”. In: *IEEE Transactions on Affective Computing* 14.4 (2023), pp. 3244–3256. DOI: 10.1109/TAFFC.2022.3226473.

[20] Jiawei Mao et al. “POSTER++: A simpler and stronger facial expression recognition network”. In: *Pattern Recognition* 157 (2025), p. 110951. ISSN: 0031-3203. DOI: https://doi.org/10.1016/j.patcog.2024.110951. URL: https://www.sciencedirect.com/science/article/pii/S0031320324007027.

[21] Maksim Kuprashevich and Irina Tolstykh. “MiVOLO: Multi-input Transformer for Age and Gender Estimation”. In: (2023). eprint: arXiv:2307.04616.

[22] Maksim Kuprashevich, Grigorii Alekseenko, and Irina Tolstykh. “Beyond Specialization: Assessing the Capabilities of MLLMs in Age and Gender Estimation”. In: (2024). eprint: arXiv:2403.02302.

[23] Wenzhi Cao, Vahid Mirjalili, and Sebastian Raschka. “Rank consistent ordinal regression for neural networks with application to age estimation”. In: *Pattern Recognition Letters* 140 (2020), pp. 325–331. ISSN: 0167-8655. DOI: https://doi.org/10.1016/j.patrec.2020.11.008. URL: https://www.sciencedirect.com/science/article/pii/S016786552030413X.

[24] Edward J. Hu et al. *LoRA: Low-Rank Adaptation of Large Language Models*. 2021. arXiv: 2106.09685 [cs.CL]. URL: https://arxiv.org/abs/2106.09685.

[25] Xuehai He et al. “Parameter-Efficient Model Adaptation for Vision Transformers”. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 37.1 (June 2023), pp. 817–825. DOI: 10.1609/aaai.v37i1.25160. URL: https://ojs.aaai.org/index.php/AAAI/article/view/25160.

[26] Dan Biderman et al. *LoRA Learns Less and Forgets Less*. 2024. arXiv: 2405.09673 [cs.LG]. URL: https://arxiv.org/abs/2405.09673.

[27] Lingling Xu et al. *Parameter-Efficient Fine-Tuning Methods for Pretrained Language Models: A Critical Review and Assessment*. 2023. arXiv: 2312.12148 [cs.CL]. URL: https://arxiv.org/abs/2312.12148.

[28] Shih-Yang Liu et al. *DoRA: Weight-Decomposed Low-Rank Adaptation*. 2024. arXiv: 2402.09353 [cs.CL]. URL: https://arxiv.org/abs/2402.09353.

[29] Tim Dettmers et al. *QLoRA: Efficient Finetuning of Quantized LLMs*. 2023. arXiv: 2305.14314 [cs.LG]. URL: https://arxiv.org/abs/2305.14314.

[30] Soufiane Hayou, Nikhil Ghosh, and Bin Yu. *LoRA+: Efficient Low Rank Adaptation of Large Models*. 2024. arXiv: 2402.12354 [cs.LG]. URL: https://arxiv.org/abs/2402.12354.

[31] Zhengmao Ye et al. *mLoRA: Fine-Tuning LoRA Adapters via Highly-Efficient Pipeline Parallelism in Multiple GPUs*. 2024. arXiv: 2312.02515 [cs.LG]. URL: https://arxiv.org/abs/2312.02515.

[32] Ying Sheng et al. "S-LoRA: Serving Thousands of Concurrent LoRA Adapters". In: *arXiv preprint arXiv:2311.03285* (2023).

[33] Ali Sabet. *BLoRA: Maximize GPU util by routing inference through multiple LoRAs in same batch.* GitHub repository. 2024. URL: https://github.com/sabetAI/BLoRA.

[34] Kimmo Kärkkäinen and Jungseock Joo. "FairFace: Face Attribute Dataset for Balanced Race, Gender, and Age". In: *CoRR* abs/1908.04913 (2019). arXiv: 1908.04913. URL: http://arxiv.org/abs/1908.04913.

[35] Zhifei Zhang, Yang Song, and Hairong Qi. "Age Progression/Regression by Conditional Adversarial Autoencoder". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017), pp. 4352–4360. URL: https://api.semanticscholar.org/CorpusID:810708.

[36] Shan Li, Weihong Deng, and Junping Du. "Reliable Crowdsourcing and Deep Locality-Preserving Learning for Expression Recognition in the Wild". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (2017), pp. 2584–2593. URL: https://api.semanticscholar.org/CorpusID:11413183.

[37] Qiong Cao et al. *VGGFace2: A dataset for recognising faces across pose and age*. 2018. arXiv: 1710.08092 [cs.CV]. URL: https://arxiv.org/abs/1710.08092.

[38] Antonio Greco et al. "Effective training of convolutional neural networks for age estimation based on knowledge distillation". In: *Neural Computing and Applications* 34 (2022), pp. 21449–21464. DOI: 10.1007/s00521-021-05981-0. URL: https://doi.org/10.1007/s00521-021-05981-0.

[39] Weihao Xia et al. “TediGAN: Text-Guided Diverse Face Image Generation and Manipulation”. In: *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2021.

[40] Weihao Xia et al. “Towards Open-World Text-Guided Face Image Generation and Manipulation”. In: *arxiv preprint arxiv: 2104.08910* (2021).

[41] Ilya Loshchilov and Frank Hutter. *Decoupled Weight Decay Regularization*. 2019. arXiv: 1711.05101 [cs.LG]. URL: https://arxiv.org/abs/1711.05101.

[42] PyTorch. *Automatic Mixed Precision package - torch.amp*. Accessed: 2025-10-13. URL: https://arxiv.org/abs/1710.08092.

[43] Ahmed Agiza, Marina Neseem, and Sherief Reda. “MTLoRA: A Low-Rank Adaptation Approach for Efficient Multi-Task Learning”. In: June 2024, pp. 16196–16205. DOI: 10.1109/CVPR52733.2024.01533.

[44] Anish Lakkapragada et al. “Mitigating Negative Transfer in Multi-Task Learning with Exponential Moving Average Loss Weighting Strategies (Student Abstract)”. In: *Proceedings of the AAAI Conference on Artificial Intelligence* 37 (June 2023), pp. 16246–16247. DOI: 10.1609/aaai.v37i13.26983.

[45] Roberto Cipolla, Yarin Gal, and Alex Kendall. “Multi-task Learning Using Uncertainty to Weigh Losses for Scene Geometry and Semantics”. In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018, pp. 7482–7491. DOI: 10.1109/CVPR.2018.00781.

Ringraziamenti

Desidero esprimere i miei più profondi e sinceri ringraziamenti, innanzitutto, alla mia famiglia, per il loro sostegno che non è mai mancato in questi anni.
Vorrei ringraziare i miei relatori, il *Prof. Mario Vento* e il *Prof. Antonio Greco*, che sono sempre stati disponibili e cordiali, sia durante le lezioni che durante l'elaborazione di questa tesi.
In conclusione, desidero esprimere la mia sincera gratitudine a tutte le persone che hanno contribuito al raggiungimento di questo obiettivo.