Department of Computer Science

MSc Data Science and Machine Learning

Mind the Gap: Can Synthetic Augmentation Boost Skin Feature Classification?

Student Number: 24220958

Supervised by Dr Yuzuko Nakamura & Suran Goonatilake

September 2025

This report is submitted as part requirement for the MSc Data Science and Machine Learning at University College London. It is substantially the result of my own work except where explicitly indicated in the text.

Abstract

The quest for truly personalized skincare is a compelling frontier for Artificial Intelligence (AI), but it's one plagued by a fundamental problem: scarce, noisy, and severely imbalanced data. This thesis tackles this challenge head-on, investigating whether diffusion-based generative augmentation, combined with cutting-edge foundation models, can make accurate cosmetic-dermatology classification a reality in these difficult conditions.

We introduce a unique, real-world dataset of close-up facial images of labeled skin conditions (moisture, oiliness, elasticity, texture, hyperpigmentation) and strategically expand it with a conservative amount of synthetic samples to ensure plausibility. We then rigorously evaluate both dermatology-specific (Google Derm Foundation, PanDerm) and generalist (DINOv2) backbones, using class-balance-aware metrics and chance-respecting baselines to provide a robust assessment of model performance.

Our findings reveal that in this low-data, high-skew environment, calibrated linear probes on frozen features prove more reliable than fine-tuning. This shows the power of these methods for adapting to new domains with minimal training. While synthetic augmentation offers selective benefits, we find it adds limited feature diversity, with its impact concentrated on a subset of features. Crucially, extreme class imbalance remains the primary hurdle for the most challenging targets.

The study is limited by dataset size, severe imbalance, inherited labels, and variable image quality. Despite these constraints, the results provide a pragmatic roadmap for small actors in the field, presenting the first systematic adaptation of dermatology foundation models to cosmetic tasks, taking a step toward robust, clinically useful AI systems despite data constraints.

Acknowledgements

This thesis has been both challenging and rewarding, but impossible without the help of those around me. My thanks go first to Yuzuko for her interest in the thesis topic and her support throughout its development, and to Suran for inspiring me during the project and sharing its precious knowledge of the field.

I am also grateful to my family and friends who have supported me in the idea of pursuing another master degree to fulfill my career dreams.

Contents

1 Introduction

1.1 Research Question & Objectives

Artificial Intelligence (AI) is rapidly transforming healthcare and the cosmetics industry, offering new opportunities for personalized treatment and product development. Traditional skincare regimens often rely on generalized formulations and broad marketing strategies that prioritize sales rather than dermatological efficacy. Such approaches frequently overlook the complexity of individual skin physiology, leading to suboptimal outcomes in both skin health and client satisfaction. In contrast, recent advances in AI have enabled the design of highly tailored skincare strategies. By analyzing intrinsic skin characteristics—such as texture, pigmentation, and hydration levels—alongside demographic and lifestyle factors (e.g., age, gender, environment, and daily habits), AI-driven systems can propose dynamic, adaptive skincare routines that respond to both physiological and external changes, delivering more precise and effective recommendations (Hash et al., 2025).

Despite substantial progress in this direction, significant challenges remain. A central barrier is the scarcity and heterogeneity of dermatological data. Most available datasets are not only small but also highly imbalanced, with limited representation across skin tones, age groups, and genders. These constraints hinder the development of robust classification models and prevent the training of foundation models at scale. While industry leaders such as L'Oréal (L'Oréal, 2024a) have developed proprietary solutions, issues of data privacy, intellectual property, and restricted access mean that smaller clinics and retail settings are left with limited resources. Consequently, practitioners often work with narrow datasets that lack demographic diversity and fail to capture the full spectrum of cosmetic dermatology presentations, limiting the potential impact of AI in everyday clinical contexts.

This project addresses this gap by investigating whether advanced generative AI techniques can alleviate data scarcity in cosmetic dermatology. Specifically, we evaluate whether diffusion-based data augmentation can enrich small, imbalanced image datasets sufficiently to enable meaningful classification of skin conditions. Furthermore, we explore whether foundation models pre-trained on dermatology-specific data can be effectively fine-tuned for cosmetic applications when supplemented with synthetic images.

The contributions of this work are fourfold:

1. Introduction of a novel, real-world dataset of close-up skin images in a cosmetic dermatology context.
2. Application of diffusion-based generative augmentation methods to address data scarcity and class imbalance.
3. Fine-tuning of state-of-the-art foundation models pre-trained on dermatological tasks.

4. Comprehensive evaluation of model performance when trained on real-only datasets versus combined real and synthetic data.

The code base for this research is available on our Github [1]

1.2 The Evolution of Cosmetic Dermatology

Cosmetic dermatology is a branch of dermatology that focuses on improving the appearance and youthfulness of the skin, addressing issues such as wrinkles, photodamage, irregularities of texture and pigmentation disorders, Kania et al. (2024). It combines medical knowledge with aesthetic practice to achieve the desired cosmetic outcomes. Clinically, effective cosmetic treatments can significantly improve patients' quality of life by alleviating dermatologic imperfections that impact self-esteem (for example, acne scarring or signs of aging). Consumer interest in this field is enormous, reflected in a multibillion dollar global skincare and beauty industry that continually drives innovation in products and procedures. However, success in cosmetic interventions has traditionally been subjective, often influenced by individual perceptions of beauty, varying practitioner skill, and manual observation, Kania et al. (2024). This subjectivity means that evaluating and achieving optimal results can vary widely from person to person, underscoring the need for more objective tools to assist in cosmetic dermatology.

Limitations of Traditional Approaches

Historically, cosmetic dermatology and skincare have relied on generalized approaches that may not account for individual variability. Dermatologists and consumers often resorted to broad skin-type classifications (e.g., 'dry', 'oily' or 'sensitive' skin) and the use of trial and error products when formulating skincare regimens. Such traditional methods are limited because they overlook the complex interplay between each person's unique skin physiology and external factors, Hash et al. (2025). A one-size-fits-all product or treatment often fails to address specific needs, leading many individuals to experiment with numerous products before finding a suitable regimen. This trial-and-error approach is not only inefficient and costly, but also frustrating—users frequently encounter suboptimal results or adverse reactions when following generic recommendations. In addition, even clinical evaluations in cosmetic dermatology can lack consistency. For example, visual grading of conditions like acne severity or photoaging signs is inherently variable between examiners, highlighting the subjectivity in traditional dermatologic evaluations, Hash et al. (2025). In summary, conventional cosmetic dermatology faces challenges in providing consistently effective, personalized care due to the reliance on subjective judgment and static treatment algorithms.

Rising Demand for Personalized Skincare

In recent years, there has been a notable rise in consumer demand for personalized skincare solutions. Heightened awareness of skincare (partly driven by social media and beauty influencer culture) has made consumers more knowledgeable about ingredients and more cognizant that individual skin characteristics vary widely.

[1] https://github.com/24220958/Skin$_F$*oundation*

Social media platforms and online communities have exposed millions of people to skincare routines, product reviews, and dermatology education, fueling expectations that one's regimen should be tailored to their unique needs. Surveys indicate that a large proportion of consumers now obtain skincare information and recommendations via social networks, and this visibility of diverse personal skincare journeys has amplified the desire for customized regimens, (Alamer et al., 2023).

Unlike previous generations who might passively accept mass-market products, today's skincare consumers actively seek solutions specific to their skin type, genetic predispositions, lifestyle, and even real-time conditions (such as climate or stress level). This demand is also evident in market trends—personalized and "made-for-me" beauty products are rapidly becoming a core segment of the cosmetics industry. The shift in consumer expectations, combined with increased dermatological awareness, has put pressure on the cosmetic dermatology field to move beyond generic treatments and embrace more individualized, data-driven approaches.

Towards a Data-Driven Approach

To meet the limitations of traditional methods and the growing desire for personalization, cosmetic dermatology is undergoing a paradigm shift toward data-driven, technology-assisted care. In particular, artificial intelligence (AI) has emerged as a key enabler of this transformation. AI systems can analyze and synthesize vast amounts of data — from high-resolution skin images to patient questionnaires, ingredient databases, and even genetic or environmental information — far beyond the capacity of manual analysis by humans, Hash2025. By leveraging advanced algorithms (including machine learning and deep learning), AI offers a more objective and evidence-based approach to complement the expert judgement of dermatologists. Notably, AI has already proven its value in medical dermatology for diagnostic tasks (for instance, achieving accuracy on par with dermatologists in melanoma detection). Building on these successes, similar technologies are now being applied to cosmetic dermatology challenges. The result is a shift from static skincare routines and subjective assessments to dynamic, adaptive skincare management guided by data. In effect, AI-driven tools can continuously adjust recommendations as an individual's skin changes or as new information is collected, ensuring that treatments evolve in real-time with the patient's needs, Hash et al. (2025). This data-driven evolution marks a move toward truly personalized cosmetic dermatology, in which care is customized and refined through computational insights.

Industry Applications

AI technologies are being applied to cosmetic dermatology across four main domains: computer vision analysis, personalized ingredient recommendations, predictive modeling, and real-time adaptive tools. Computer vision models can objectively assess features such as wrinkles, pores, and pigmentation from close-up images, reducing the subjectivity of human grading and enabling more standardized evaluations, (Hash et al., 2025). Ingredient recommendation engines mine large databases to match active compounds with an individual's skin profile, offering a safer and more tailored approach than trial-and-error product usage. Predictive models extend this by forecasting skin outcomes over time, allowing practitioners to anticipate how a

regimen or procedure might affect a patient's skin, (Kania et al., 2024). Finally, adaptive AI-powered devices (e.g., smart dispensers or laser systems) dynamically adjust treatments based on real-time data such as climate or immediate skin response. Collectively, these applications illustrate AI's potential to deliver more precise, consistent, and personalized care in cosmetic dermatology.

The commercial skincare sector has quickly embraced AI as a competitive differentiator. L'Oréal, through its acquisition of ModiFace, launched SkinConsult AI and Perso, which use deep learning and environmental data to provide at-home diagnostics and personalized product formulations, Hash et al. (2025) and L'Oréal (2024a, 2024b). Proven Skincare employs the Skin Genome Project, analyzing thousands of ingredients, products, and peer-reviewed studies to create customized formulations, Proven Skincare (2024). Similarly, Haut.AI uses computer vision to assess user-submitted selfies and generate product recommendations in real time, Haut.AI (2024).

As demonstrated, AI implementation can help scale personalization in the wide consumer market offering insights once limited to clinical settings, L'Oréal (2024c). However, this is widening the gap between large corporation that have high resources in terms of technology equipment and data and smaller companies/clinics.

Limitations and Ethical Concerns

Despite these advances, significant challenges remain. Data scarcity and imbalance limit the development of robust AI models, particularly for underrepresented skin tones, age groups, and genders, (Kania et al., 2024). This not only reduces performance but risks perpetuating bias in recommendations. Privacy concerns are also critical: AI platforms often require sensitive facial data, raising questions about data protection and informed consent. Finally, accessibility poses a barrier — advanced AI tools are costly and remain concentrated within large corporations, limiting their availability to smaller clinics and everyday practice.

Therefore, even though AI implementation is helping scale personalization in the wide consumer market offering insights once limited to clinical settings, (L'Oréal, 2024c), this is mainly guided by large corporation that have high resources in terms of technology equipment, data availability and monetary availability, resulting in innovation being mainly guided by few main players.

2 Related Work

In this section, we describe the preliminary work related to our research objective.

2.1 Literature Review

This review situates the present work within three strands of evidence: (i) the emergence of vision *foundation models* for dermatology, (ii) strategies for coping with data scarcity and imbalance, and (iii) the rise of *generative* augmentation—particularly diffusion models—as a means to expand diversity while preserving clinical signal. The aim is to motivate the methodological choices in §3.1.

Foundation models for dermatology and cosmetic imaging

Large-scale pretraining has reshaped medical image analysis. Domain-specific models such as *PanDerm* leverage millions of multi-institutional dermatology images and consistently set strong baselines across diverse tasks, often with limited fine-tuning data (Yan et al., 2025). In parallel, Google's *Derm Foundation* provides a dermatology-tailored encoder that yields transferable embeddings and has reported robust performance across skin types, suggesting broadly representative features (Rikhye et al., 2024). Generalist backbones (e.g., *DINOv2*) also transfer well to dermoscopic and clinical imagery, frequently outperforming conventional CNNs on standard benchmarks under frozen or lightly adapted regimes (Oquab et al., 2024; Mietkiewicz et al., 2025). Together, these results indicate that strong, pretrained features can unlock performance in low-data settings—a premise this study adopts for the cosmetic domain, where domain shift (lighting, reflectance, aesthetic cues) can differ from medical pathology.

Data scarcity, imbalance, and augmentation baselines

Dermatology datasets are typically small and skewed, with minority classes and underrepresented demographics limiting generalisation (Mietkiewicz et al., 2025). Classical remedies—label-preserving augmentation (flips, rotations, colour jitter), oversampling, class-weighted losses, and transfer learning—offer incremental gains but rarely resolve severe imbalance or distributional gaps (Perez et al., 2018; Aladhadh et al., 2022; Krishna et al., 2023; Qin et al., 2020). The literature also cautions that headline accuracies can mask majority-class dominance; chance-aware baselines and class-balance–sensitive metrics (e.g., Macro-F1, PR-AUC for rare targets) are increasingly emphasised to provide a more faithful picture of performance (Mietkiewicz et al., 2025). These practices inform our evaluation design.

Generative AI-based augmentation

Earlier work used GANs (Generative Adversarial Network) to synthesise lesions and showed that targeted synthetic data can aid minority classes, albeit with stability and control limitations (Qin et al., 2020). Diffusion models now offer higher fidelity and controllability, enabling conditioning on diagnosis, tone, or other attributes.

Recent studies show that diffusion-generated dermatology images can match the downstream utility of real images and, when mixed with real data, improve minority-class performance without degrading majority-class results (Akrout et al., 2023). Conditioning has been used to address demographic gaps directly—e.g., synthesising presentations across Fitzpatrick skin types to reduce performance disparities (P. Wang et al., 2024; Sagers et al., 2022). These findings motivate our use of diffusion-based augmentation where data are rare and class priors are extreme.

Quality, validity, and fairness considerations

Beyond visual plausibility, the field increasingly validates synthetics by their impact on downstream tasks and, where possible, via expert reader studies, which have reported comparable diagnostic signal between real and diffusion-generated images in controlled settings (Ktena et al., 2024). At the same time, label consistency, artefacts, and colour dependence remain practical risks, reinforcing the need for conservative generation protocols, expert spot checks, and subgroup analyses. Generative augmentation has also been explored as a fairness tool—narrowing gaps across skin tones or acquisition domains when paired with careful evaluation and calibration (P. Wang et al., 2024; Sagers et al., 2022; Ktena et al., 2024).

Positioning of this study

Building on these strands, this work examines whether diffusion-based augmentation can complement foundation backbones in cosmetic dermatology, a setting marked by severe data scarcity and domain shift from medical pathology. We therefore (i) prioritise robust, frozen-feature baselines with calibrated linear probes, (ii) use synthetic data conservatively and feature-targeted rather than uniformly, and (iii) adopt chance-aware metrics to interpret improvements credibly, aligning our choices with best practices in the literature.

3 Methodology

3.1 Methods Overview

This chapter presents the methodological framework for our experiment, diving into the architectural details of the models implemented.

We describe:

1. The generative augmentation approach using *diffusion models* with a focus on the **Stable Diffusion** model to synthesize additional training images, aiming to mitigate data scarcity and class imbalance.
2. The vision *foundation models* leveraged in this work, outlining the architecture and training paradigms. In particular, we apply three cutting-edge models:
   (a) **DINOv2**: a self-supervised Vision Transformer model that works with generic visual features.
   (b) **Google Derm Foundation Model**: a domain-specific model pretrained on dermatology images.
   (c) **PanDerm**: a recently introduced multimodal foundation model pretrained on dermatology images.

To better understand the implemented methods, we provide background on the underlying transformer architecture (Vision Transformer, ViT) and relevant key machine learning paradigms §A.1. These concepts are the foundations of how large pretrained models can be harnessed to improve the performance despite having limited examples.

3.2 Diffusion Models for Data Augmentation

Diffusion models (DM) are a class of generative models that have recently achieved state-of-the-art results in image synthesis. At their core, DMs learn data by progressively denoising a random noise input, effectively inverting a gradual noising process that is applied to training images Ho et al. (2020). In the forward (diffusion) process, an image is repeatedly corrupted by adding noise; the model is trained to reverse this process, step by step, to recover the original image distribution. Through this training, diffusion models can sample new, realistic images by starting from pure noise and iteratively refining it. Crucially, this framework is stable to train and capable of modeling complex image distributions, which has made diffusion models competitive or superior to earlier generative approaches like GANs in terms of output fidelity and diversity Dhariwal and Nichol (2021).

In our experiment, we implement the Stable Diffusion model, a latent diffusion model introduced by Rombach et al. (2022). Stable Diffusion operates not in the high-dimensional pixel space, but in a lower-dimensional latent space of an autoencoder.

By diffusing and denoising in the latent space, it achieves a near-optimal trade-off between computational efficiency and image detail preservation. This approach dramatically reduces the resources needed for training and generation, yet produces high-fidelity images. Moreover, Stable Diffusion incorporates a cross-attention mechanism that allows it to condition the image generation on auxiliary inputs (such as text prompts) without retraining the entire model. In practice, this means one can guide the generative process by providing a textual description or other context, enabling flexible generation of images corresponding to a desired concept. Figure 3.1 shows the diffusion model generation process and the mathematical formulation below describes its functioning in details.

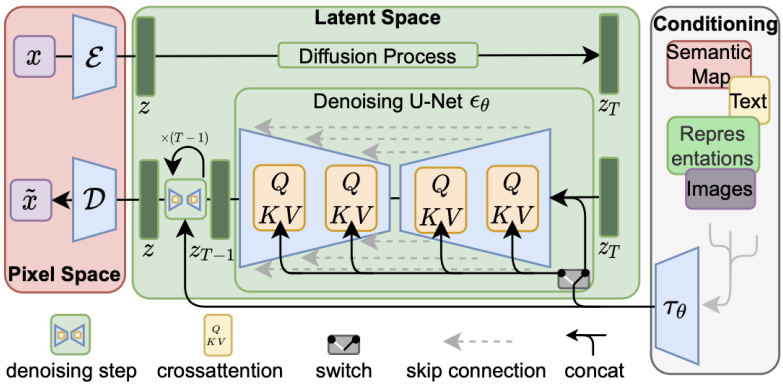


Figure 3.1: Diffusion model process, adapted from Rombach et al., 2022. In the forward pass (top row), an original image is progressively noised over multiple steps until it becomes pure noise. A diffusion model learns the reverse pass (bottom row): starting from random noise, the model iteratively denoises the signal to synthesize a realistic image. By learning this sequence of denoising steps, diffusion models can generate new images drawn from the same distribution as the training data.

Mathematical Formulation

To understand Stable Diffusion, it is useful to formalize how a diffusion model operates in latent space. Following Ho et al. (2020) and the latent adaptation of Rombach et al. (2022), diffusion models define a Markov forward noising process and learn a parametrized reverse denoising process, as shown in Figure 3.1.

Let $x \in \mathbb{R}^{H\times W\times 3}$ be an image. Latent diffusion models (LDMs) make use of Variational Autoencoders:

Variational Autoencoder (VAE). A *variational autoencoder* is a neural network that learns to compress an image $x \in \mathbb{R}^{H\times W\times 3}$ into a lower-dimensional latent representation $z_0 = E(x)$ via an encoder E, and reconstruct it back with a decoder D, i.e. $\hat{x} = D(z_0)$.

Stable Diffusion applies diffusion not directly on pixels, but in this latent space, which is both more compact and semantically meaningful. It first compresses x into a lower-dimensional latent $z_0 = E(x) \in \mathbb{R}^{h\times w\times d}$ using a variational autoencoder

(VAE) encoder E, and learn to denoise *in latent space*. The VAE decoder D later maps latents back to pixel space $\hat{x} = D(z_0)$.

Forward (noising) process. Diffusion models define a forward process where Gaussian noise is gradually added to the latent z_0. At each time step t, noise is injected with variance β_t, giving:

$$q(z_t \mid z_{t-1}) = \mathcal{N}\Big(z_t;\ \sqrt{\alpha_t} z_{t-1},\ \beta_t \mathbf{I}\Big), \tag{3.1}$$

where $\alpha_t = 1 - \beta_t$. This can be written in closed form as:

$$z_t = \sqrt{\bar{\alpha}_t}\, z_0 + \sqrt{1 - \bar{\alpha}_t}\, \varepsilon, \qquad \varepsilon \sim \mathcal{N}(0, \mathbf{I}), \tag{3.2}$$

with $\bar{\alpha}_t = \prod_{s=1}^{t} \alpha_s$. Intuitively, as t increases, the latent representation becomes increasingly noisy, until it is indistinguishable from random noise (see top row of Figure 3.1).

Reverse (denoising) process. The model then learns the reverse process: starting from pure noise z_T, it iteratively denoises back to a clean latent z_0. The reverse-time transitions are Gaussians and each step is modeled as:

$$p_\theta(z_{t-1} \mid z_t, c) = \mathcal{N}\Big(z_{t-1};\ \mu_\theta(z_t, t, c),\ \sigma_t^2 \mathbf{I}\Big), \tag{3.3}$$

where c denotes conditioning (e.g., a text embedding describing the target image). A common parameterization learns a noise-prediction network ε_θ (a U-Net operating in latent space) and sets

$$\mu_\theta(z_t, t, c) \;=\; \frac{1}{\sqrt{\alpha_t}} \left(z_t - \frac{\beta_t}{\sqrt{1 - \bar{\alpha}_t}}\, \varepsilon_\theta(z_t, t, c) \right), \qquad \sigma_t^2 \ \text{ fixed (e.g., } \sigma_t^2 = \tilde{\beta}_t). \tag{3.4}$$

Noise prediction objective. Instead of predicting z_{t-1} directly, the network is trained to predict the noise ε that was added in the forward process. The training objective is the mean squared error:

$$\mathcal{L}_{\text{diff}}(\theta) = \mathbb{E}_{t, z_0, \varepsilon}\Big[\ \big\|\varepsilon - \varepsilon_\theta(z_t, t, c)\big\|_2^2\ \Big]. \tag{3.5}$$

(Variants include v-prediction and hybrid perceptual losses; cf. Rombach et al. (2022).)

This so-called 'simple loss' is the same objective used in Stable Diffusion.

Conditioning via cross-attention. In Stable Diffusion, the denoising U-Net is equipped with *cross-attention layers* that inject conditioning information c into ε_θ with cross-attention: transformer-encoded tokens $\tau(c)$ provide keys/values to attention blocks inside the U-Net, guiding denoising toward the condition. At sampling time, *classifier-free guidance* blends conditional and unconditional predictions to better enforce the condition:

$$\tilde{\varepsilon}_\theta(z_t, t, c;\ s) \;=\; \varepsilon_\theta(z_t, t, \varnothing) \;+\; s\Big(\varepsilon_\theta(z_t, t, c) - \varepsilon_\theta(z_t, t, \varnothing)\Big), \tag{3.6}$$

with guidance scale $s \geq 0$ trading off fidelity and adherence to c.

Latent autoencoding objectives. Since diffusion acts in latent space, LDMs train the VAE jointly or beforehand with a reconstruction and KL regularization loss:

$$\mathcal{L}_{\text{VAE}}(E, D) = \lambda_{\text{rec}}\, \mathbb{E}_x\Big[\ell_{\text{rec}}\Big(D(E(x)),\, x\Big)\Big] + \lambda_{\text{KL}}\, \mathbb{E}_x\Big[\text{KL}\Big(q_E(z\,|\,x)\,\|\,\mathcal{N}(0, \mathbf{I})\Big)\Big], \quad (3.7)$$

where ℓ_{rec} is typically a combination of ℓ_2 and perceptual losses to encourage faithful, high-frequency reconstructions in pixel space (Rombach et al., 2022).

Reconstruction. Finally, with sampling starting from $z_T \sim \mathcal{N}(0, \mathbf{I})$, it iterates (3.3) down to z_0 clean latent space, which is decoded back to image space via the VAE decoder: $\hat{x} = D(z_0)$ (see bottom row of Figure 3.1). For label- or text-conditional augmentation, we set c to the target class/description and sample multiple $\hat{x}$ per under-represented class.

In this way, Stable Diffusion learns to synthesize high-quality images by operating entirely in a compressed latent space rather than directly in pixels. In this project we leverage diffusion models for data augmentation in the domain of dermatology images.

3.3 Foundation Models in Vision

A foundation model is a large-scale neural network trained on an extremely broad dataset (often using self-supervised or weakly-supervised learning) to learn general-purpose representations, which can be adapted via fine-tuning to a wide range of downstream tasks, Bommasani et al. (2021).

Choice of foundation models. This thesis evaluates three vision foundation models that represent complementary philosophies of model development. Together, these three models span the spectrum from general-purpose to domain-specific to multimodal architectures, enabling a systematic comparison of how underlying model design and pretraining choices influence downstream performance. Table 3.1 summarizes key characteristics of these models.

The three models of interest are:

1. DINOv2 (by Meta AI): A self-supervised Vision Transformer model producing universal image features, Oquab et al. (2024). We use this as a representative state-of-the-art foundation model trained on generic images.

2. Derm Foundation Model (by Google): A domain-specific model trained on a large collection of dermatology images (and related data), Google Health AI (2025). This model is used to assess the benefit of a foundation model tailored to skin imagery.

3. PanDerm (by Yan *et al.*, 2025): A multimodal dermatology foundation model trained on a very large dermatology dataset including clinical photos, dermoscopic images, and histopathology ,Yan et al. (2025). This model represents the current cutting-edge in dermatology-specific foundation models.

Model	Architecture	Pretraining Data	Objective	Domain (Purpose)
DINOv2-L (2023)	ViT-Large (Vision Transformer, ∼304M params)	142M images curated from 1.2B web images	Self-supervised distillation (teacher–student); multi-task eval	General-vision FM (universal features for classification, segmentation, etc.)
Derm Foundation (Google) (2023)	BiT-M ResNet-101x3 (CNN)	Stage 1: millions of image–text pairs (public web health data); Stage 2: derm datasets (>400 conditions)	Contrastive pretraining; supervised fine-tuning	Dermatology FM (feature extractor for skin images; few-shot classifier training)
PanDerm (2025)	ViT-Large encoder + decoupled heads (multimodal inputs)	2.1M dermatology images (11 institutions; 4 modalities)	Masked latent modeling; CLIP-based latent alignment	Multimodal dermatology FM (classification, segmentation, longitudinal analysis, etc.)

Table 3.1: Key characteristics of the vision foundation models considered in this work. ViT = Vision Transformer, CNN = Convolutional Neural Network, FM = Foundation Model.

We describe each in turn, focusing on their architecture and training method, as these factors determine how we implement and fine-tune them and what performance gains are expected in our experiment.

Architectural Foundation: Vision Transformer

Both DINOv2 and PanDerm build on the Vision Transformer (ViT) architecture, which has become central to modern vision foundation models, therefore before diving in detail about the models, we provide an overview of the ViT framework. Originally introduced for natural language processing (Vaswani et al., 2017), transformers were adapted to images by Dosovitskiy et al. (2021).

Figure 3.2 shows the ViT architecture and the steps involved in the pipeline

1. Patch embeddings. An image $x \in \mathbb{R}^{H\times W\times C}$ is divided into N patches of size $P \times P \times C$. Each patch is flattened and projected into a d-dimensional embedding:
$$z_0^i = x_p^i E + E_{\text{pos}}^i, \quad i = 1, \ldots, N,$$
where E is a learnable projection and E_{pos}^i are positional encodings. A learnable `[CLS]` token $z_0^{[\text{CLS}]}$ is prepended, yielding the input sequence
$$Z_0 = [z_0^{[\text{CLS}]}, z_0^1, \ldots, z_0^N].$$

2. Transformer encoder. The sequence is processed by L encoder layers, each combining multi-head self-attention (MSA) and a feed-forward Multi Layer Perceptron (MLP), with residual connections and normalization:
$$Z'_\ell = \text{MSA}(\text{LN}(Z_{\ell-1})) + Z_{\ell-1}, \quad Z_\ell = \text{MLP}(\text{LN}(Z'_\ell)) + Z'_\ell,$$

for $\ell = 1, \ldots, L$. Through this, ViTs capture both local and long-range dependencies between patches.

3. Classification head (MLP). After L layers, the final hidden state of the `[CLS]` token $z_L^{[\text{CLS}]}$ acts as a compact representation of the whole image. This vector is passed to a classification head W:

$$\hat{y} = \text{softmax}(W z_L^{[\text{CLS}]}),$$

producing class probabilities.

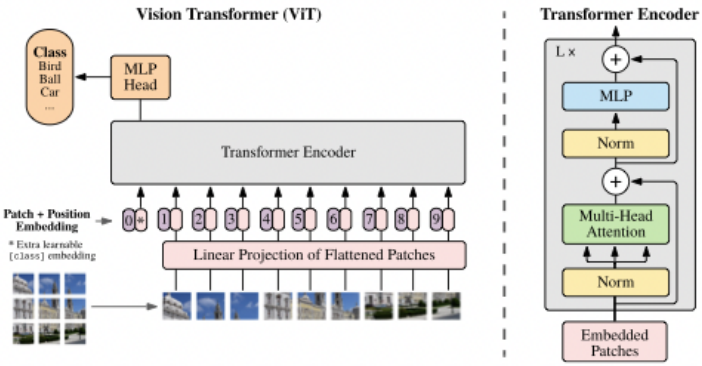


Figure 3.2: Vision Transformer architecture, adapted from Dosovitskiy et al. (2021). An image is split into patches, linearly embedded, and prepended with a learnable `[CLS]` token and positional encodings. The sequence passes through multiple transformer encoder layers (right), where multi-head self-attention and MLP blocks iteratively refine representations. The final `[CLS]` token embedding encodes the global image representation, which is fed into a Multi Layer Perceptron head for classification.

In the context of this work, ViTs are valuable because of their ability to model long-range dependencies across image patches that makes them well suited for dermatology tasks, where subtle spatial differences in patterns (e.g., texture) are critical for accurate classification.

3.3.1 DINOv2

The name "DINO"stands for *Distillation with No Labels*, it was originally proposed by Caron et al. (2021) as a method to train Vision Transformers without any human annotation, by leveraging a form of knowledge distillation between networks. DINOv2 (Oquab et al., 2024) builds upon this idea at a much larger scale.

A ViT backbone produces a global image representation that feeds a task-specific head (e.g., a linear classifier for image classification) (Dosovitskiy et al., 2021; Kloster, 2024). The largest version of DINOv2, contains a ViT with 1 billion parameters (ViT-Giant), and serves as a teacher to train a series of smaller models such as DINOv2-Base (with around 86 million parameters), here implemented. Below is a mathematical description of the model:

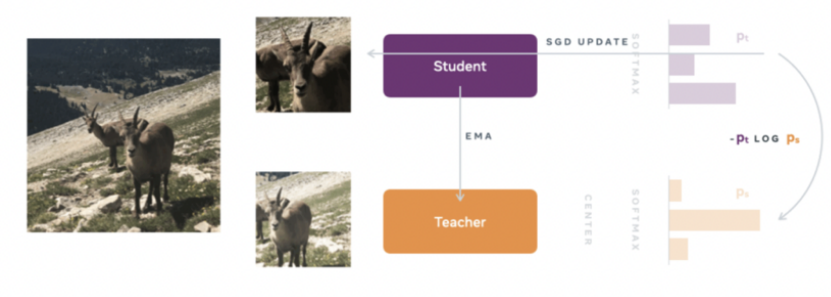


Figure 3.3: Student-teacher mechanism, adapted from official page of DINOv1 (Meta AI, 2023).

Mathematical formulation

DINOv2 builds on the principle of self-distillation (Caron et al., 2021), where a *student* network learns to match the output distribution of a *teacher* network on different augmented views of the same image (see Figure 3.3).

Formally, let x denote an input image, and $\{v_i^s\}_{i=1}^M$, $\{v_j^t\}_{j=1}^N$ be sets of M *student* and N *teacher* views obtained via random augmentations (e.g. crops, color jittering). The student encoder f_θ and teacher encoder f_ξ (typically Vision Transformers with identical architecture but distinct parameters) produce feature representations:

$$z_i^s = f_\theta(v_i^s), \qquad z_j^t = f_\xi(v_j^t). \tag{3.8}$$

Both outputs are projected into a K-dimensional space through a projection head and normalized with a softmax temperature τ to form probability distributions:

$$p_i^s = \mathrm{softmax}(z_i^s/\tau_s), \qquad p_j^t = \mathrm{softmax}\Big((z_j^t - \bar{z}^t)/\tau_t\Big), \tag{3.9}$$

where $\bar{z}^t$ is a centering term to stabilize training, and τ_s, τ_t are student and teacher temperatures.

The student is trained by minimizing the cross-entropy between teacher and student distributions across all pairs of teacher–student views:

$$\mathcal{L}_{\mathrm{DINO}}(\theta) = -\sum_{i=1}^{M}\sum_{j=1}^{N}(p_j^t)^\top \log p_i^s. \tag{3.10}$$

Teacher update and collapse avoidance. The teacher parameters ξ are updated as an exponential moving average (EMA) of the student parameters θ:

$$\xi \leftarrow \lambda\xi + (1-\lambda)\theta, \tag{3.11}$$

where $\lambda \in [0, 1)$ is a momentum coefficient that increases during training. To prevent trivial solutions (e.g. all representations collapsing to a constant), DINOv2 further incorporates: (i) a *centering* operation on teacher outputs, (ii) a *sharpening* operation via low teacher temperature τ_t, and (iii) a *feature spreading regularizer* based on the KoLeo entropy term (Oquab et al., 2024).

Patch-level objectives. Beyond global image representations, DINOv2 adopts patch-level losses inspired by iBOT (Zhou et al., 2021). Masked patches in the student input are encouraged to match the corresponding teacher patch embeddings, enabling the model to learn fine-grained visual correspondences (See Figure 3.4). The total loss thus combines global distribution matching (3.10) with local patch prediction and regularization:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{DINO}} + \lambda_{\text{patch}}\,\mathcal{L}_{\text{patch}} + \lambda_{\text{KoLeo}}\,\mathcal{L}_{\text{KoLeo}}, \tag{3.12}$$

where λ_{patch} and λ_{KoLeo} balance the contribution of patch-level alignment and entropy regularization.

Figure 3.4: Example of Masked Image Modeling (MIM), adapted from Xie et al., 2021.

In summary, DINOv2 provides a strong, generic visual representation. Its ViT backbone with multi-head self-attention is adept at capturing both local textures and global structure in images. For our purposes, we selected the DINOv2-Base model (with 86M parameters) as a pretrained backbone.

This model has demonstrated excellent transfer learning performance; for instance, Mietkiewicz et al. (2025) used DINOv2-B to classify skin disease images and achieved high accuracy across multiple dermatology datasets, indicating that DINOv2's features are well-suited to even fine-grained medical imagery.

Dino permits full end-to-end fine-tuning of the backbone and head, which we exploit in our experiments. By fine-tuning DINOv2-B on our cosmetic dermatology dataset, with an appropriate classification head, we aim to exploit its rich feature space. The expectation is that even with relatively few training images, the model's pretrained knowledge will allow it to distinguish subtle visual differences (e.g., differences in skin moisture/oiliness/elasticity levels), having already learned a broad spectrum of visual concepts during pretraining. Our training will update primarily the final layers to adapt the general DINOv2 representation to the specifics of cosmetic dermatology classes, a strategy in line with standard fine-tuning practices for foundation models.

3.3.2 Google Derm Foundation Model

While DINOv2 represents a general-purpose vision model, our study also considers a *domain-specific* alternative: the Google Derm Foundation Model. Released in late 2023 as part of the Google Health AI Developer Foundation (HAI-DEF), it was designed to accelerate AI for dermatological image analysis by providing a pretrained feature extractor tailored to skin images.

Mathematical formulation

The Derm Foundation model (Google Health AI, 2025) builds on the Big Transfer (BiT) framework(Kolesnikov et al., 2020), which scales supervised pretraining of Convolutional Neural Networks (CNN), (LeCun et al., 1998). In particular, it uses a BiT-M ResNet-101x3 backbone, a widened variant of ResNet that has been shown to transfer effectively from large-scale pretraining to smaller downstream tasks.

ResNet-based CNNs combine hierarchical convolutional features (LeCun et al., 1998) with residual learning to enable very deep networks (He et al., 2016); BiT scales this recipe for strong transfer in data-limited domains (Kolesnikov et al., 2020).

The Derm Foundation model was trained in two main stages:

Supervised pretraining. Given an input image $x \in \mathbb{R}^{H\times W\times 3}$ with associated label $y \in \{1,\dots,C\}$, the backbone network f_θ produces a representation $z = f_\theta(x) \in \mathbb{R}^d$. This is projected to class logits via a linear classifier $W \in \mathbb{R}^{C\times d}$:

$$\hat{y} = \mathrm{softmax}(Wz). \tag{3.13}$$

The supervised pretraining objective is the standard cross-entropy loss over a large dataset $\mathcal{D}_{\text{pre}}$:

$$\mathcal{L}_{\text{pre}}(\theta, W) = -\frac{1}{|\mathcal{D}_{\text{pre}}|} \sum_{(x,y)\in\mathcal{D}_{\text{pre}}} \sum_{c=1}^{C} \mathbf{1}_{[y=c]} \log \hat{y}_c. \tag{3.14}$$

Contrastive pretraining. This contrastive paradigm follows the InfoNCE objective (van den Oord et al., 2018), popularized at scale by CLIP (Radford et al., 2021) and adapted to medical imaging in ConVIRT (Zhang et al., 2022), enabling image embeddings to align with semantically meaningful text descriptions. Given an image embedding z^I and a text embedding z^T, similarity is measured by cosine similarity $s(z^I, z^T) = \frac{z^I \cdot z^T}{\|z^I\|\|z^T\|}$. The objective encourages matched pairs to have high similarity and mismatched pairs low similarity, typically via a softmax cross-entropy loss over a batch of N pairs:

$$\mathcal{L}_{\text{con}} = -\frac{1}{N}\sum_{i=1}^{N}\left[\log \frac{\exp(s(z_i^I, z_i^T)/\tau)}{\sum_{j=1}^{N}\exp(s(z_i^I, z_j^T)/\tau)} + \log \frac{\exp(s(z_i^T, z_i^I)/\tau)}{\sum_{j=1}^{N}\exp(s(z_i^T, z_j^I)/\tau)}\right], \tag{3.15}$$

where τ is a temperature parameter. This stage equips the model with semantically rich embeddings aligned with dermatological descriptions.

After pretraining, the model can be adapted to downstream dermatology tasks. In practice, the released encoder weights θ^* are kept fixed, and only the classification head W is re-optimized on a smaller dermatology dataset $\mathcal{D}_{\text{derm}}$:

$$\mathcal{L}_{\text{derm}}(W) = -\frac{1}{|\mathcal{D}_{\text{derm}}|} \sum_{(x,y)\in\mathcal{D}_{\text{derm}}} \sum_{c=1}^{C_{\text{derm}}} \mathbf{1}_{[y=c]} \log \hat{y}_c. \tag{3.16}$$

At present, the released Google Derm model only supports the extraction of 6144-dimensional embeddings and the training of a classification head on top, without

access to fine-tune the backbone. This contrasts with the other foundation models considered in this work (DINOv2 and PanDerm), which permit end-to-end fine-tuning.

3.3.3 PanDerm

PanDerm is a cutting-edge foundation model explicitly developed for clinical dermatology, introduced by Yan et al. (2025). It represents one of the largest and most comprehensive efforts to date to build a general-purpose model for skin disease understanding. Unlike generalist vision FMs (e.g., DINOv2), PanDerm is *multimodal*, trained to process the diverse image types encountered in dermatology practice: clinical close-up photographs, dermoscopic images, total body photography, and histopathology slides, to cite a few. Its architecture centers on a ViT-Large encoder, complemented by a decoupled regression head and alignment modules, designed to support multiple training objectives and downstream tasks.

Mathematical formulation

PanDerm is trained on over two million dermatology images collected across 11 international institutions and four modalities (Yan et al., 2025). Its learning is governed by two self-supervised objectives: masked latent reconstruction (He et al., 2022) and CLIP-based latent alignment (Radford et al., 2021).

Masked latent modeling. This objective follows the masked autoencoder (MAE) paradigm (He et al., 2022), where parts of the image are hidden and the model learns to reconstruct them, encouraging holistic representations. Given an image x, the encoder f_θ produces patch-level latent representations $z = f_\theta(x)$. A random subset $\mathcal{M}$ of these latent tokens is masked, and the model is trained to reconstruct them via a regression head g_ϕ:

$$\hat{z}_m = g_\phi(z_{\setminus\mathcal{M}}), \qquad m \in \mathcal{M}. \tag{3.17}$$

The objective minimizes mean squared error (MSE) between predicted and original latent tokens:

$$\mathcal{L}_{\text{mask}}(\theta, \phi) = \frac{1}{|\mathcal{M}|} \sum_{m \in \mathcal{M}} \|z_m - \hat{z}_m\|_2^2. \tag{3.18}$$

This encourages holistic feature learning by forcing the encoder to infer missing context.

CLIP latent alignment. In parallel, PanDerm aligns its image embeddings with those of a pretrained CLIP (Contrastive Language–Image Pretraining) image encoder f_{CLIP}. Given an image x, PanDerm produces an embedding $z^P = f_\theta(x)$ and CLIP produces $z^C = f_{\text{CLIP}}(x)$.

Cosine similarity is computed as

$$s(z^P, z^C) = \frac{z^P \cdot z^C}{\|z^P\| \|z^C\|}, \tag{3.19}$$

and PanDerm is trained to maximize alignment via an InfoNCE-style loss over a batch of size N:

$$\mathcal{L}_{\text{align}} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{\exp(s(z_i^P, z_i^C)/\tau)}{\sum_{j=1}^{N}\exp(s(z_i^P, z_j^C)/\tau)}, \tag{3.20}$$

where τ is a temperature parameter. This transfers semantic structure from CLIP into PanDerm's embedding space.

Total objective. The overall training loss combines masked modeling and alignment:

$$\mathcal{L}_{\text{PanDerm}} = \lambda_{\text{mask}}\,\mathcal{L}_{\text{mask}} + \lambda_{\text{align}}\,\mathcal{L}_{\text{align}}, \tag{3.21}$$

with weights λ_{mask} and λ_{align} balancing the contributions.

Transfer to downstream tasks. After pretraining, PanDerm can be adapted to dermatology tasks such as classification, segmentation, or longitudinal analysis by attaching task-specific heads to the pretrained encoder. Its large-scale multimodal training enables strong few-shot performance: the model often matches or exceeds prior benchmarks using only a fraction of labeled data.

Summary and usage in this study. In our experiments, we employ PanDerm as a foundation model for cosmetic dermatology classification. Although PanDerm supports multiple modalities, we restrict its use to single-image clinical photographs, corresponding to our dataset. We both (i) extract frozen embedding (to compare with Google Derm FM) and (ii) fine-tune the ViT encoder and classification head with careful regularization and reduced learning rates to mitigate overfitting on our small dataset. This setup allows us to test whether PanDerm's large-scale, dermatology-specific pretraining confers a tangible advantage over generalist (DINOv2) or narrower domain (Derm FM) models.

4 Experiments

4.1 Dataset

The dataset used in this project was provided by BodyMetrics S.p.A. [1] and processed in compliance with current privacy regulations. The images were originally collected in 2017 as part of a pilot study evaluating the performance of a state-of-the-art skin scanning device and its ability to classify skin conditions for personalized product recommendation.

The study involved 39 participants residing in London (UK), primarily within the 20–35 age group, with only three outliers above this range. The pool was predominantly female and consisted mainly of individuals with fair to medium skin tones, with comparatively fewer examples of darker skin types. As such, the dataset cannot be considered representative of the general population in terms of gender balance, age diversity, or skin-type distribution.

The choice to employ this dataset reflects the broader challenge investigated in this thesis: the limited availability of high-quality, representative dermatology datasets for small research start-ups (such as BodyMetrics) or clinical institutions. This mirrors the wider issue in healthcare AI of data scarcity and imbalance, particularly the under-representation of more severe or less common conditions. Addressing such limitations is precisely where the generative augmentation strategies, such as those studied here, may hold promise.

Demographic statistics

Survey data collected during the pilot provides insight into participant demographics. Figure 4.1 illustrates the gender imbalance: 36 subjects (92.3%) identified as female, compared to only 3 subjects (7.7%) identifying as male. This asymmetry reflects a broader trend in cosmetic dermatology research, where women remain the dominant study population, but it also highlights a limitation for model development, as gender-diverse datasets are essential for generalizable outcomes.

Skin tone was categorized into three groups based on the CIE-Lab color calculation (§A.2) shown in (Figure 4.2). The distribution is skewed toward lighter tones, with 20 participants (51.3%) classified as *light*, 9 (23.1%) as *medium*, and 10 (25.6%) as *dark*. Although darker tones are not entirely absent, the under-representation of medium and darker skin types is aligned with a persistent bias in skincare research and product development, where fairer skin has historically been driving the innovation (Georgievskaya et al., 2025).

The age distribution (Figure 4.3) shows a strong concentration in the target range of 20–35 years, with only three participants above this threshold. This narrow age range reduces the dataset's generalizability to older or younger populations.

[1] UCL start-up spinooff working on creating a platform where data about a person's physiognomy is collected and currently updated to improve fashion and skincare experience

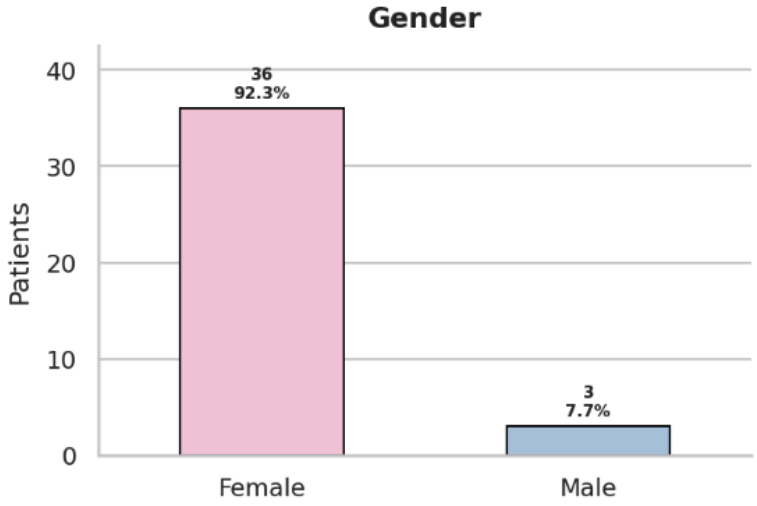


Figure 4.1: Gender distribution

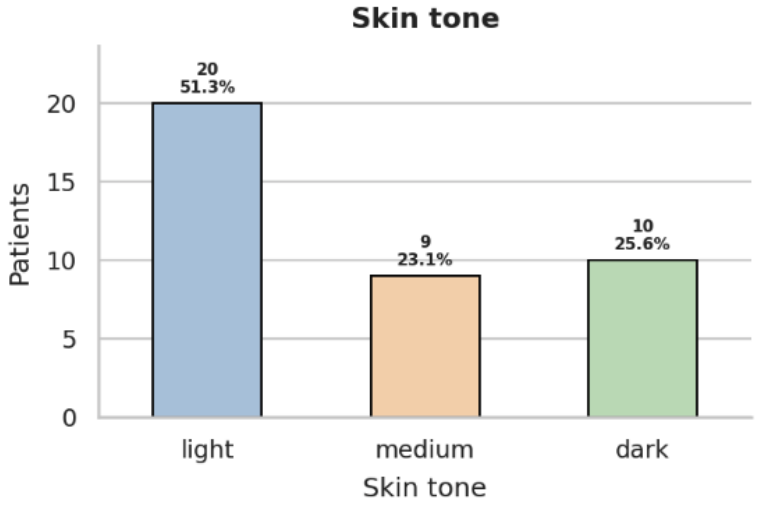


Figure 4.2: Skin tone distribution

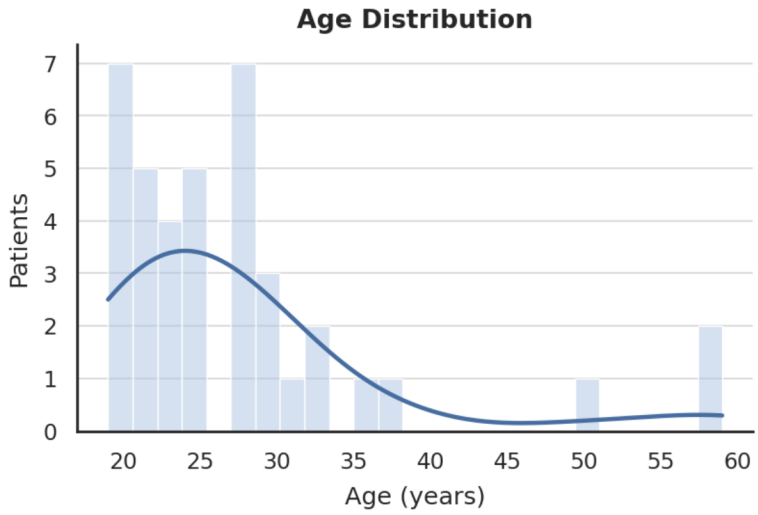


Figure 4.3: Age distribution

Data description

The dataset consists of macro close-up images of facial skin patches extracted from each subject in the study. For every participant, two anatomical regions were examined: the *cheek* and the *forehead*. Within each region, five distinct images were captured, one per target feature. The **features** under investigation are:

- **Moisture**
- **Oiliness**
- **Elasticity**
- **Texture**
- **Hyperpigmentation**

The images were acquired using the *Dermograph*, a specialized skin analysis device produced by *mySkin UK*[2] (now discontinued). Although detailed specifications of the Dermograph are no longer available, comparable dermatological imaging systems typically rely on combinations of multispectral illumination (white, ultraviolet, polarized, and infrared light), fluorescent and polarized light, and magnification optics (up to $\times 30$) to enhance the visibility of surface and subsurface features of the skin (Aram HUVIS Co., 2025).

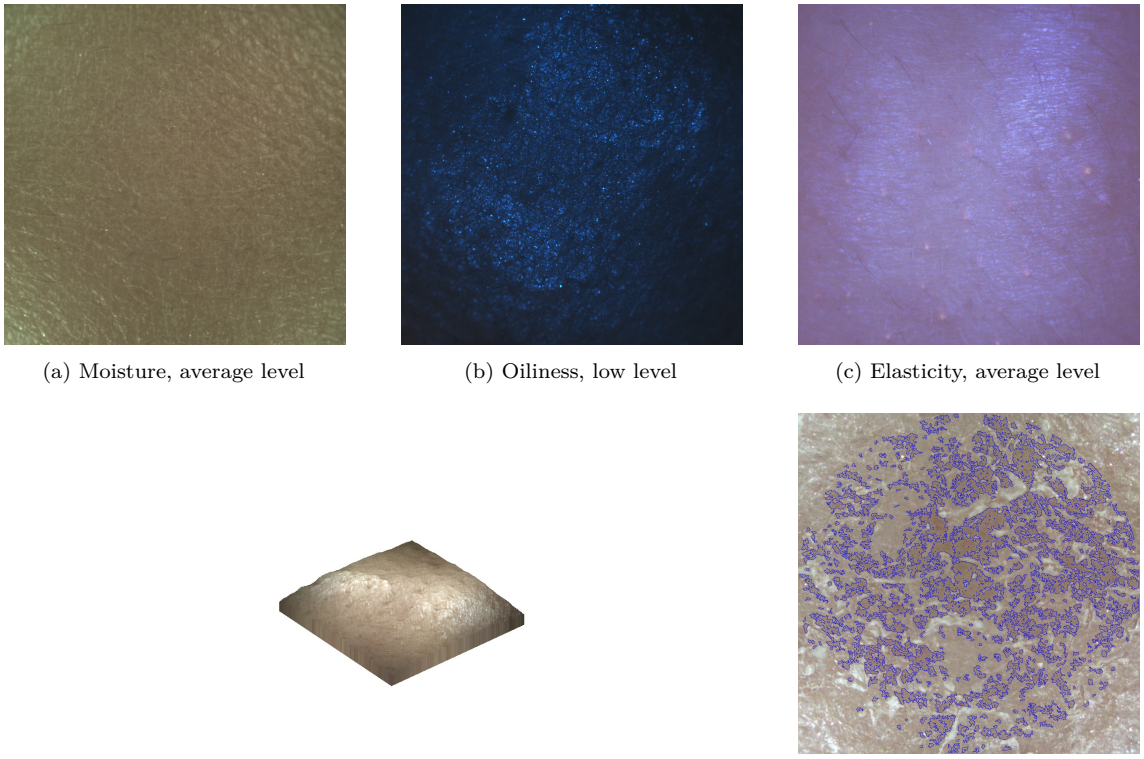
The Dermograph operated in conjunction with proprietary software, which integrated machine learning–based algorithms to detect and quantify the selected skin features of interest. Each feature was assigned a percentile-based score ranging from 0 to 100, calibrated against reference skin-health databases. For interpretability, these continuous scores were discretized into three ordinal categories—*low*, *average*, and *high*—tailored to each feature specifics. In addition, visual indicators are added on the captured images to highlight the analyzed skin regions (see §A.2 for further details on the labeling protocol).

The categorical labels derived from the Dermograph thus constitute the supervision signal for our experiments. However, it is important to acknowledge **limitations** inherent to this process: (i) the scoring pipeline is proprietary and opaque, which makes it difficult to independently validate how percentile thresholds were defined or calibrated. This raises the possibility of algorithmic bias, especially if the underlying reference databases are not fully representative across age, gender, or skin tone groups as is our case(Georgievskaya et al., 2025). (ii) Discretization of continuous scores into three categories (*low*, *average*, *high*) introduces a degree of information loss: subtle but clinically relevant gradations between subjects may be collapsed into the same class. (iii) Finally, ecause the device is no longer supported, reproducibility of results using identical acquisition conditions cannot be guaranteed.

These limitations highlight why augmentation strategies and strong modeling choices could be crucial in compensating for biases in the labeling process and the limited diversity of the original dataset.

[2]Website: https://myskinuk.com/

Figure 4.4 illustrates representative examples of the collected data, showing one image per skin feature.



(a) Moisture, average level

(b) Oiliness, low level

(c) Elasticity, average level

(d) Texture, average level

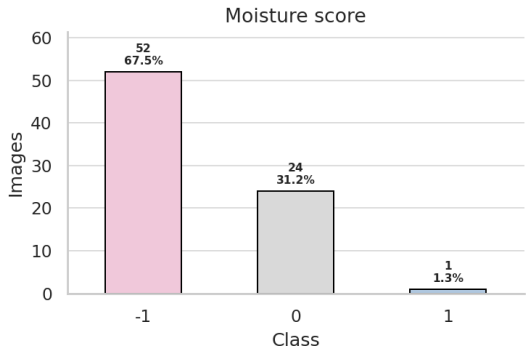
(e) Hyperpigmentation, high level

Figure 4.4: Example close-up images for each target feature.

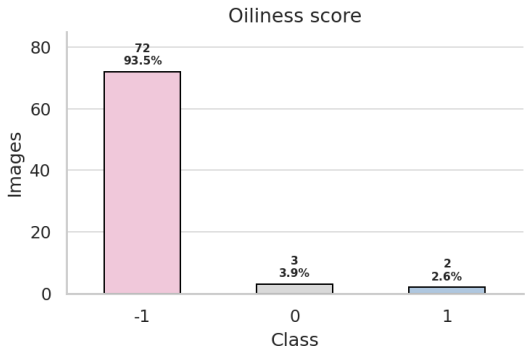
Class distribution

The classes were mapped to the ordinal levels -1, 0, and 1, since the interpretation of these levels depends on the specific feature: a higher score is not always indicative of a "better" outcome. For example, high elasticity reflects taut and healthy skin (mapped to $+1$), whereas high hyperpigmentation indicates an undesirable condition (mapped to -1).

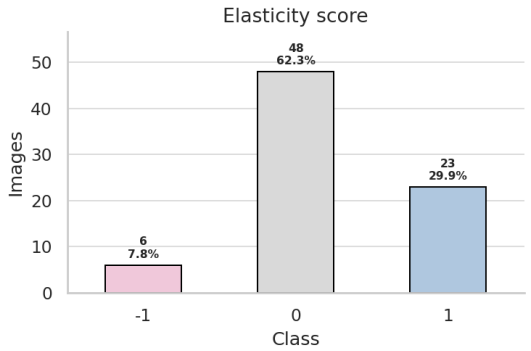
Figure 4.5 illustrates the class distribution for each feature. It is visible how highly imbalanced the classes are: all the features present a clear majority class that reflects the "normal" condition for the group age taken into account, with some extremes like *Moisture* having one data point for class 1, and *Oiliness* having 93.5% of the data belonging to class -1. This will present the highest obstacle in our classification task.



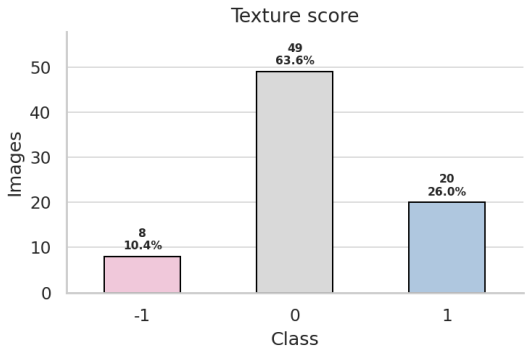
(a) Moisture score class distribution



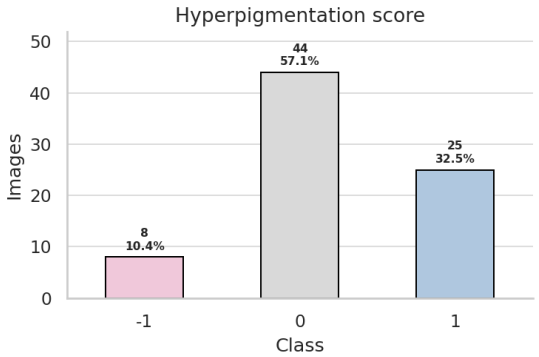
(b) Oiliness score class distribution



(c) Elasticity score class distribution



(d) Texture score class distribution



(e) Hyperpigmentation score class distribution

Figure 4.5: Class distributions for each target feature.

Dataset split

The total considered dataset is composed of 385 images (77 per feature). To prevent information leakage across sets, we perform a grouped split by `patient_id`: all images from a given subject are assigned either to train or to test, never both. We target a 20% test set and use a fixed random seed for reproducibility.

As shown above, because the dataset is small and imbalanced, we adopt a *label-aware* selection of test patients to avoid empty classes. Concretely, let $\mathcal{B} = \{(\text{feature}, \text{class})\}$ denote all feature–class bins observed in the dataset and let $\text{support}(b)$ be the number

of samples in bin $b \in \mathcal{B}$. We require at least one test example for any bin with sufficient support:

$$\text{require\_test}(b) = \begin{cases} 1 & \text{if support}(b) \geq \tau, \\ 0 & \text{otherwise,} \end{cases}$$

with threshold τ=3 in our experiments. We then greedily select test patients that cover the largest number of still-unmet required bins, until all feasible requirements are satisfied; if multiple choices tie, a seedable random order breaks ties. Then, we add additional patients at random (still disjoint from train) until the desired test fraction is reached.

This procedure yields a patient-disjoint split with (i) minimal risk of class absence per feature in the test set when data permit, and (ii) no cross-subject leakage. As a sanity check, we verify zero patient overlap between splits and report per-feature class counts for both train and test.(Figure/Table 4.5 and §A.2). The resulting split selected 8 test patients (20.5% of subjects), with 366 train images, 96 test images across features.

However even with this approach both oiliness and moisture present an empty class (label = 1) in the test set. Therefore for the model implementation these features are considered **binary** classes, combining the minority classes:

- Moisture: non-hydrated skin vs hydrated skin
- Oiliness: dry skin vs. non-dry skin

4.2 Data Augmentation

Given the small size and imbalance of our dataset, data augmentation was introduced as a critical step to improve model robustness. For this purpose, we adopted the `Stable Diffusion v1.5` image-to-image (`img2img`) pipeline (Rombach et al., 2022), which enables the generation of synthetic images conditioned on real inputs while preserving the fine-grained structure of skin patches (based on the Stable Diffusion model architecture, see Section §3.2). Data augmentation was performed on the train set, excluding the test set for a fair model performance assessment.

Model and parameters

We employed the Stable Diffusion v1.5 `img2img` pipeline via the Hugging Face `diffusers` library, a modular PyTorch[3] toolkit that provides provides a unified framework for diffusion models, offering pretrained pipelines and modular components for image-to-image, and other generative tasks (Stable Diffusion v1-5, 2022; von Platen et al., 2022).

Feature-specific parameters. For each feature, we tuned three hyperparameters within controlled ranges, with a small random jitter applied at each run to promote variability in the synthetic images:

[3] https://pytorch.org/

- **Strength** (0.20–0.35): controls how strongly the input image is modified by diffusion; lower values preserve more of the original texture, higher values allow greater transformation. It was kept low because it had tendencies to create external objects on the skin patch from presence of moles/hair/white spots.
- **Guidance scale** (3.5–5.5): determines how closely the generation follows the text prompt; smaller values allow more diversity, larger values enforce prompt fidelity.
- **Steps** (24–36): the number of denoising iterations; trade-off between more steps improve detail and stability at the cost of computation time.

Schedulers. Schedulers define how noise is added and removed during the diffusion process. The **Euler Ancestral scheduler** (Karras et al., 2022) is a fast, stochastic sampler that produces sharp images by approximating the diffusion trajectory with ancestral noise injection. The **DPMSolverMultistep scheduler** (Lu et al., 2022) is a high-order deterministic solver that achieves better stability and reduces exposure artifacts, making it particularly suitable for challenging features such as oiliness.

The model was initialized with the Euler Ancestral scheduler for most features, while the DPMSolverMultistep scheduler was used for oiliness due to its sensitivity to exposure artifacts. Image resolution was fixed at 512×512 **pixels**.

Prompting strategy

Prompts were designed to remain conservative and faithful to the original data, in order to minimize the risk of generating dermatologically implausible artifacts. Each feature was associated with carefully defined textual descriptions at three class levels $(-1, 0, 1)$. Negative prompts were used extensively to filter out common failure modes (e.g., portraits, text overlays, uniform blue washes in oiliness, or heatmap-like false colors). Prompts were token-limited to 75 tokens, necessitating compact descriptors. Because of the inherent difficulty of dermatological prompting — and the fact that the author is not a clinical dermatologist — the strategy explicitly avoided excessive extrapolation. Instead, the aim was to enhance underrepresented cases while staying close to the clinical characteristics visible in the original scans. Oiliness proved particularly challenging: not only were original samples few, but they were often low-quality or dominated by strong blue illumination, requiring additional preprocessing (contrast-limited adaptive histogram equalization) and stricter negative prompting. (For more detail consult Appendix §3.2).

Oversampling approach

We applied proportional oversampling at +80% per (feature, class) bucket. Several percentage were tested starting from +50%, but this choice was motivated by two considerations:

- Ensuring that all classes were represented in the training set while avoiding situations where classes with fewer than five samples would be forced into balance, which is likely to harm model generalization.
- Reflecting recent evidence that strict class balancing may not always improve

learning outcomes and can introduce bias in small, noisy datasets (Verzino, 2021).

The result was an augmented dataset that preserved the underlying class distribution while reducing the prevalence of empty or severely underrepresented classes.

Quality control and metrics

To ensure plausibility, we tested generated samples against quantitative and feature-specific quality checks:

- **LPIPS similarity** (Arabboev et al., 2024): Learned Perceptual Image Patch Similarity measures distance in a deep feature space between original (x) and synthetic (x_0) patches, using unit-normalized activations $\hat{y}^l$ from a pre-trained CNN across L layers:

$$d(x, x_0) = \sum_{l=1}^{L} \frac{1}{H_l W_l} \sum_{h,w} \left\| w_l \odot \left(\hat{y}^l_{hw} - \hat{y}^{l,0}_{hw} \right) \right\|_2^2, \tag{4.1}$$

  where H_l, W_l are spatial dimensions of layer l, w_l are learned channel-wise weights, and $\odot$ denotes element-wise multiplication.

- **Blur ratio**: computed as the ratio of Laplacian variances between synthetic and real patches,

$$\text{BlurRatio} = \frac{\sigma^2_{\text{Laplacian}}(\text{synthetic})}{\sigma^2_{\text{Laplacian}}(\text{real})}, \tag{4.2}$$

  filtering out overly smooth generations.

- **Specular coverage (oiliness only)**: a custom metric estimating the proportion of bright blue specular pixels in normalized illumination, checked against acceptable ranges per class label.

- **SSIM (structural similarity)** (Arabboev et al., 2024): used for logging only, compares luminance $l(x, y)$, contrast $c(x, y)$ and structural similarity $s(x, y)$ between two images x, y:

$$SSIM(x, y) = [l(x, y)]^\alpha \cdot [c(x, y)]^\beta \cdot [s(x, y)]^\gamma, \tag{4.3}$$

  with $\alpha, \beta, \gamma > 0$ weighting the contributions. Higher SSIM values indicate greater perceptual similarity.

- **Specular coverage (oiliness only)**: a custom metric estimating the proportion of bright blue specular pixels in normalized illumination, checked against acceptable ranges per class label.

  Given an image $I \in \mathbb{R}^{H \times W \times 3}$ with RGB channels (R, G, B), we define a binary mask

$$M_{ij} = \mathbf{1}[\, B_{ij} - \max(R_{ij}, G_{ij}) > \delta_b \;\; \wedge \;\; \max(R_{ij}, G_{ij}, B_{ij}) > \tau_v \,], \tag{4.4}$$

where $\delta_b = 15$ and $\tau_v = 80$ are empirical thresholds. The specular coverage score is then computed as

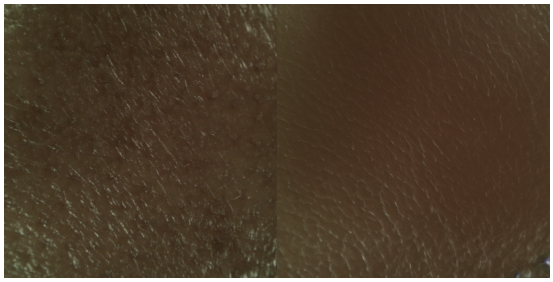
$$\text{SpecCov}(I) = \frac{1}{HW} \sum_{i=1}^{H} \sum_{j=1}^{W} M_{ij}, \tag{4.5}$$

i.e. the proportion of pixels identified as bright, blue-dominant highlights.

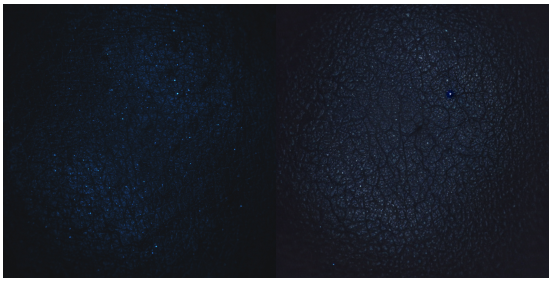
These metrics resulted harsh on our implementation, failing to capture the right characteristics of an image to be kept, therefore the thresholds were kept loose. Images failing to meet these thresholds could be discarded, though in our case we retained all outputs for diversity while logging metrics for transparency.

This augmentation strategy provided a controlled expansion of the training dataset, particularly in low-support regions of the feature space. The final augmented dataset combined original and synthetic samples in proportions that balanced the need for representation with the avoidance of unrealistic class balancing.

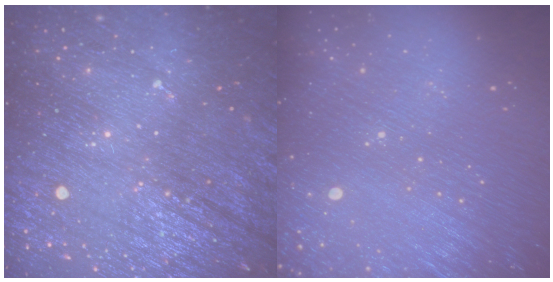
Figure 4.6 shows examples of real (left) versus generated image (right) for each feature.



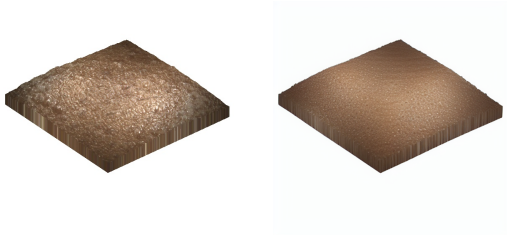
(a) Moisture. Real vs Generated data. LPIPS: 0.28, SSIM: 0.71, blur: 68.75



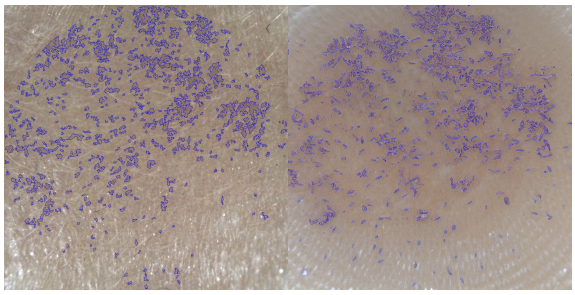
(b) Oiliness. Real vs Generated data. LPIPS: 0.32, SSIM: 0.67, blur: 326.5



(c) Elasticity. Real vs Generated data. LPIPS: 0.19, SSIM: 0.75, blur: 53.9



(d) Texture. Real vs Generated data. LPIPS: 0.22, SSIM: 0.71, blur: 626.72



(e) Hyperpigmentation. Real vs Generated data. LPIPS: 0.32, SSIM: 0.37, blur: 356.4

Figure 4.6: Example of real vs generated images.

4.3 Foundation Models implementation

In this section, we report the implementation of the three selected Foundation Models: **Google Derm Foundation**, **PanDerm**, **DINOvs** (see Section §3.1) and the different performed experiments.

The aim is to test how these robust models, pretrained on millions of images (domain-specific and non) perform on our small datasets and compare their performance. The same models are then implemented on a new dataset composed by the real images plus the generated ones to test whether this improves their performance and allows for better results.

Training protocol and evaluation

We divided the experiment into two parts:

1. Evaluation of the use of vision foundation models as frozen feature extractors, followed by a lightweight classifier head This setup provides a strong baseline

to assess both the representational power of different backbones and the effect of synthetic oversampling on downstream performance.

2. Finetuning of the PanDerm and DINOv2 models

Real vs synthetic data. All experiments were run in two regimes: (i) using only the original dataset, and (ii) augmenting the training pool with Stable Diffusion–generated images (Section §4.2). This comparison allows us to quantify the practical benefit of augmentation for each feature and backbone.

Baselines. In addition to foundation models, we trained **dummy classifiers** (predicting the most frequent label, uniform random guesses or according to the empirical class prior) to establish performance lower bounds.

Evaluation metrics. Model performance was primarily assessed using the **macro-F1** score, which gives equal weight to each class regardless of prevalence, thereby providing a robust measure under class imbalance.

For a set of classes $\mathcal{C}$, with per-class precision P_c and recall R_c, the F1 score for class c is:

$$F1_c = \frac{2 \cdot P_c \cdot R_c}{P_c + R_c}, \quad P_c = \frac{TP_c}{TP_c + FP_c}, \quad R_c = \frac{TP_c}{TP_c + FN_c}, \tag{4.6}$$

where TP_c, FP_c, and FN_c denote the true positives, false positives, and false negatives for class c. The macro-F1 is then the unweighted average across classes:

$$F1_{\text{macro}} = \frac{1}{|\mathcal{C}|} \sum_{c \in \mathcal{C}} F1_c. \tag{4.7}$$

In addition, we report the **weighted-F1**, which accounts for class imbalance by weighting each per-class F1 score by the class prevalence n_c:

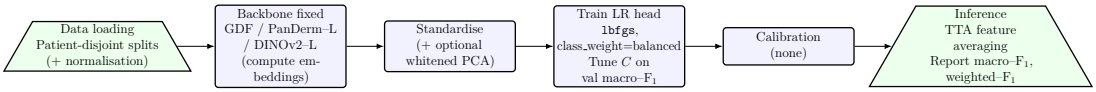
$$F1_{\text{weighted}} = \frac{1}{N} \sum_{c \in \mathcal{C}} n_c \cdot F1_c, \quad N = \sum_{c \in \mathcal{C}} n_c. \tag{4.8}$$

We chose macro-F1 as primary criterion to maintain a focus on *fair evaluation across underrepresented classes*; we also reported weighted-F1 to provide a complementary measure aligned with real class distributions. Prior work has highlighted the relevance of these metrics in imbalanced learning, showing that macro-F1 prevents minority classes from being overshadowed by majority ones, whereas weighted-F1 captures overall performance trends (Mosley et al., 2022; Chicco & Jurman, 2020; Yang & Loog, 2018).

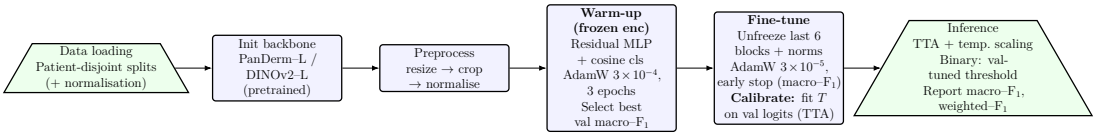
For completeness, additional diagnostic metrics were logged, including accuracy, Matthews correlation coefficient (MCC), Cohen's κ, and AUROC. though these were not used as primary evaluation criteria. All reported results correspond to the held-out test set, which was strictly disjoint by patient identity from the training and validation data.

Test-Time Augmentation (TTA). To increase robustness and reduce variance at inference, we employed *test-time augmentation*. In this strategy, multiple transformed variants of a test image (e.g., random crops, horizontal flips, scaling) are generated and passed independently through the model. Their corresponding embeddings or predictions are then aggregated (e.g. via averaging) to produce the final output. By allowing the model to observe different but plausible representations of the same input, TTA mitigates sensitivity to local artifacts and stabilises predictions, which is particularly valuable for classes represented by very few samples in the held-out test set. Surveys such as Shorten and Khoshgoftaar (2019) document the widespread use of geometric transformations during both training and inference to improve performance under limited data regimes. In lung ultrasound image studies (Yu et al., 2023), TTA has been shown to improve performance when evaluating external datasets in medical imaging by stabilizing output across variable image conditions. For LR, TTA was used for feature averaging; for fine-tuning, it was combined with calibration (see Section 4.3.2).

Repeat-N runs. Each experiment was repeated with three random seeds ($n = 3$), reporting mean $\pm$ standard deviation of the metrics to ensure robustness against stochastic variability caused by the scant classes.



(a) Frozen backbones with logistic regression head.



(b) Fine-tuning with residual MLP head and cosine classifier.

Figure 4.7: Parallel pipelines for frozen and fine-tuned foundation model experiments, with aligned stages (data, backbone, preprocessing, head, calibration, inference).

4.3.1 Frozen embeddings with logistic regression head

To establish a rigorous and reproducible baseline, we first evaluated the foundation models in a frozen setting. For each backbone, image embeddings were computed and kept fixed during training.

Classification head. A lightweight *logistic regression (LR)* (refer to Section A.1 for details) classifier was trained on top of these embeddings to predict skin condition labels. Logistic regression remains a widely adopted baseline in machine learning research due to its robustness and generalization properties, even under limited data conditions (Yang & Loog, 2018; Maalouf, 2011). A light Multi Layer Perceptron was also considered but turned out to be too complex and overfit the dataset.

Preprocessing and dimensionality reduction. To reduce overfitting and stabilise optimisation in the small-n setting, the LR head was embedded within a preprocessing pipeline including standardisation and dimensionality reduction. Binary tasks (e.g., *moisture*, *oiliness*) used threshold tuning on validation data to maximise macro-F1. For multiclass tasks (three-level labels), the multinomial logistic regression formulation was employed directly.

A summary of the LR head characteristics and training protocol is provided in Table 4.1.

Table 4.1: Logistic regression head on frozen embeddings: implementation characteristics and training protocol.

Component	Specification
Input features	Embeddings from GDF (6144-d), PanDerm-L, or DINOv2-L. For test-time augmentation (TTA), multiple crops/flips were embedded and averaged.
Standardisation	Zero-mean, unit-variance scaling on the training set only.

Continued on next page

Table 4.1 (continued)

Component	Specification
Dimensionality reduction	PCA with whitening, capped at $\min(d, n-1, 256)$, where d is feature dimension and n is number of training samples, to stabilise LR training when samples $\ll d$.
Classifier	Logistic regression (`lbfgs` solver, `max_iter`=300), multinomial for multiclass tasks.
Regularisation	L_2 penalty, inverse strength $C \in \{0.5, 1.0, 2.0\}$, tuned on validation.
Class imbalance	`class_weight=balanced` (inverse-frequency reweighting).
Binary decision rule	Threshold $t^\star$ tuned on validation (grid 0.2–0.8) to maximise macro-F1 rather than fixing 0.5.
Validation split	StratifiedGroupKFold with patient-level grouping; validation fraction ≈12.5%. Ensures no patient leakage and class coverage.
Test set	Fixed, patient-disjoint hold-out split. No overlap with training/validation.
TTA	FiveCrop + horizontal flips, averaged per image before classification.
Model selection	C and PCA dimension chosen by validation macro-F1; final model retrained on the corresponding fold.
Reproducibility	Controlled seeds for NumPy/PyTorch; Repeat-N protocol (N=3) with mean ± std reported.

4.3.2 Fine-tuning of foundation models

We fine-tuned two ViT-based backbones—*PanDerm-Large* (ViT-L/16; initialised from the released PanDerm checkpoint) and *DINOv2-Large* (ViT-L/14; initialised from `timm` pretrained weights)—for cosmetic dermatology classification. ViTs have demonstrated strong transfer under limited labels when pre-trained at scale (Dosovitskiy et al., 2021). DINOv2 provides robust self-supervised features competitive across tasks and data regimes (Oquab et al., 2024). PanDerm is a recent dermatology-specific foundation model trained self-supervised on multi-institutional, multi-modality skin images; we use its public ViT-L checkpoint for domain-adapted transfer (Yan et al., 2025; Yan & colleagues, 2024).

Input & normalisation. To minimise train–test drift, we adopted each backbone's native preprocessing (input size, interpolation, channel-wise mean/std) resolved via specific `timm`[4] configuration; these settings were applied uniformly across splits (see `timm` documentation, Wightman, 2023). Images were resized to `RESIZE_TO`=384

[4]resolved via `timm.data.resolve_data_config` and instantiated with `create_transform`

then randomly cropped to `CROP_TO`=224 before normalisation, consistent with ViT patchified inputs (Dosovitskiy et al., 2021).

Classification head. On encoder features we attach a compact `StrongHead`: a LayerNorm–Linear projection to width 1024 with GELU activation function and dropout(0.35), followed by two pre-norm residual MLP blocks and a cosine (angular) classifier with a learnable scale s:

LayerNorm → Linear (1024) → GELU → Dropout (0.35) → Pre-norm Residual MLP ×2 → Cosine classifier

Cosine-margin classifiers (e.g., CosFace/ArcFace) preserve angular geometry on the hypersphere and tend to yield better separability and calibration in low-data regimes (H. Wang et al., 2018; Deng et al., 2019). GELU, LayerNorm and dropout choices follow standard transformer practice for stable optimisation (Dosovitskiy et al., 2021).

Data pipeline and imbalance handling. Training augmentations included `RandomResizedCrop(224, scale=[0.85, 1.0])` and horizontal flips; implementations follow `torchvision` (TorchVision Contributors, 2023a, 2023b). To mitigate skew on binary targets (*moisture*, *oiliness*), we used a weighted sampler targeting $\approx 45\%$ minority per mini-batch and applied stronger photometric jitter to minority samples with higher probability (a cost-sensitive augmentation strategy for class-imbalance) (Chen et al., 2024). Multiclass tasks used `class_weight=balanced` in the loss.

Optimisation schedule. Training proceeded in two phases:

1. *Warm-up*: encoder frozen, optimise only the head using AdamW (lr[5]=3×10^{-4}, wd[6]=2×10^{-4}).
2. *Fine-tuning*: unfreeze the last 6 ViT blocks (and final norms), optimise with AdamW (lr=3×10^{-5}, same decay).

We trained models using the AdamW optimiser, which decouples weight decay from adaptive updates and is widely regarded as effective for transfer learning in vision tasks (Loshchilov & Hutter, 2019). The loss was standard cross-entropy (PyTorch Contributors, 2023), augmented with label smoothing (0.05 for binary tasks, none for three-class) to reduce overconfidence and improve calibration and generalisation (Müller et al., 2020). To stabilise optimisation, we applied gradient-norm clipping at 1.0 preventing occasional exploding updates that can arise in deep networks (Pascanu et al., 2013). Training was monitored with early stopping on validation macro-F_1 (patience of 5 epochs), a classical but effective safeguard against overfitting in small-n regimes (Prechelt, 1998).

Validation protocol At validation and test time we employed *test-time augmentation* (TTA), combining `torchvision.transforms.FiveCrop` and horizontal flips (`functional. hflip`) to generate ten views per image (TorchVision Contributors, 2023b, 2023c). Model predictions were averaged across these views, which reduces sensitivity to local artefacts and stabilises outputs. This strategy is particularly

valuable under limited data and distributional variability, and is widely used in medical imaging applications (Shorten & Khoshgoftaar, 2019; Yu et al., 2023).

Temperature scaling. To further calibrate predictive probabilities, we applied post-hoc *temperature scaling* (Guo et al., 2017). Given logits $z(x)$ for an input x, probabilities are obtained as

$$p(y|x;T) = \text{softmax}\left(\frac{z(x)}{T}\right),$$

where $T > 0$ is a scalar temperature. The optimal T was estimated on the validation set by minimising the negative log-likelihood of TTA-averaged logits using the LBFGS optimiser, which is well-suited for precise optimisation of low-dimensional parameters. At test time, the learned T was fixed and applied uniformly. For binary tasks, an additional threshold was tuned on the validation set (sweep 0.2–0.8) after TTA and temperature scaling to maximise macro-F_1.

Uncalibrated neural networks often produce overconfident probabilities, a well-documented issue in medical imaging where downstream decisions can be sensitive to uncertainty; temperature scaling directly addresses this by aligning predicted confidence with empirical accuracy, thereby improving the clinical trustworthiness of the model outputs (Guo et al., 2017).

Table 4.2: Fine-tuning configuration for PanDerm-L and DINOv2-L on cosmetic dermatology classification.

Component	Specification
Backbone	PanDerm-Large (ViT-L/16) initialised from the released PanDerm checkpoint; DINOv2-Large (ViT-L/14) from `timm` pretrained weights. Model config provides input size and mean/std used consistently across splits.
Input & normalisation	Resize to `RESIZE_TO` = 384, crop to `CROP_TO` = 224; normalise with backbone-specific (μ, σ) from `resolve_data_config`.
Augmentation (train)	RandomResizedCrop(224, scale=[0.85, 1.0]) and HorizontalFlip. For binary tasks, minority-targeted photometric jitter (brightness/contrast, sharpness, Gaussian blur) with higher probability.
Imbalance handling	Binary (*moisture*, *oiliness*): weighted sampler targeting $\approx$ 45% minority per batch. Multiclass: `class_weight=balanced` in loss.
Head	LayerNorm $\rightarrow$ Linear(1024) $\rightarrow$ GELU $\rightarrow$ Dropout(0.35) $\rightarrow$ two pre-norm residual MLP blocks $\rightarrow$ cosine (angular) classifier with scale $s \in [16, 30]$.

Continued on next page

Component	Specification
Loss	Cross-entropy; label smoothing 0.05 for binary tasks, 0 for three-class tasks.
Warm-up	Encoder frozen; optimise head only with AdamW (lr = 3×10^{-4}, wd = 2×10^{-4}), 3 epochs; best validation macro-F_1 (with TTA) retained.
Fine-tuning	Unfreeze last 6 ViT blocks (and final norms); AdamW with lr = 3×10^{-5}, wd = 2×10^{-4}, up to 15 epochs; gradient clipping at 1.0; early stopping on validation macro-F_1 (patience 5).
Batching & split	Batch size 8; StratifiedGroupKFold with patient-level grouping; validation fraction $\approx 12.5\%$.
Validation/Test	TTA (FiveCrop + horizontal flips; 10 views). Probabilities from TTA-averaged logits. Temperature T fitted on validation and used at test. Binary threshold tuned on validation (grid 0.2–0.8).
Metrics	Primary: macro-F_1. Secondary: weighted-F_1; additional logs include accuracy, MCC, QWK, and AUROC (binary) / AUROC micro/macro and Top-2 (multiclass).
Reproducibility	Fixed seeds (NumPy/PyTorch); patient-disjoint splits; Repeat-$N = 3$ protocol with mean±std reporting.

5 Results

This section reports the performance on the held-out test set across all features and models. We report means over $n = 3$ runs ($\pm$ one standard deviation). Because the dataset is very small and strongly imbalanced, we prioritise *Macro-F1* for class balance and report *Weighted-F1* to indicate majority-class behaviour. For reference, we include three dummy baselines (most-frequent, uniform, and stratified); the stratified baseline is our primary "chance" comparator and is visualised as a horizontal line in the figures.

Overview: frozen embeddings implementation

Figure 5.1 provides an at-a-glance comparison of Macro-F1 across features for Google Derm Foundation, PanDerm (frozen) and DINOv2 (frozen) on real-only vs. real and synthetic data. The full numbers are given in the tables below.

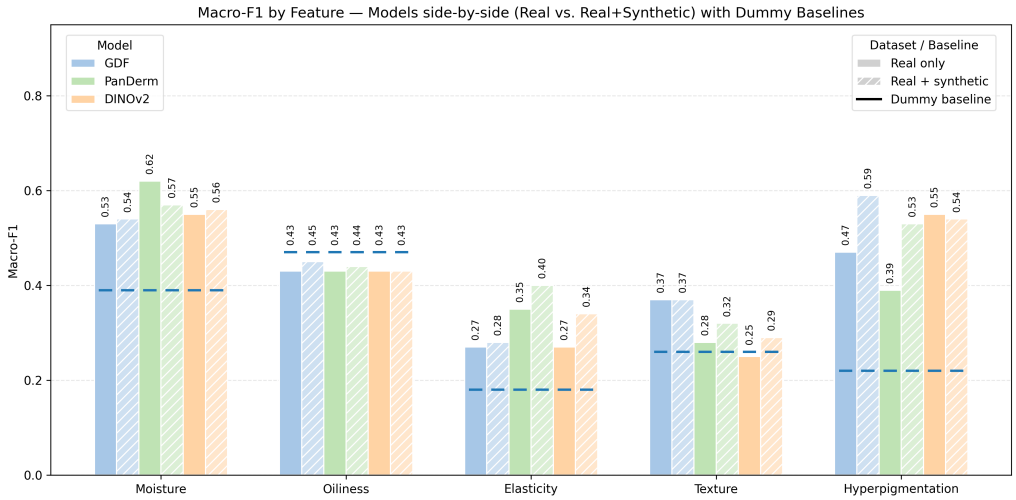


Figure 5.1: Comparison of MacroF1 metrics across models for each feature (Real vs Real+Synthetic data): Frozen embeddings implementation

Performance across models (frozen embeddings)

Table 5.1: Performance of **GDF** (*frozen*) across features (mean $\pm$ std over 3 runs).

	Real only		With synthetic data	
Feature	**Macro-F1**	**Weighted-F1**	**Macro-F1**	**Weighted-F1**
Moisture	0.53±0.09	0.54±0.09	0.54±0.07	0.56±0.06
Oiliness	0.43±0.00	0.75±0.00	0.45±0.01	0.80±0.02
Elasticity	0.27±0.02	0.34±0.02	0.28±0.10	0.27±0.06
Texture	0.37±0.06	0.54±0.04	0.37±0.11	0.55±0.09
Hyperpigmentation	0.47±0.04	0.61±0.06	0.59±0.09	0.72±0.03

Table 5.2: Performance of **PanDerm** (*frozen*) across features (mean ± std over 3 runs).

	Real only		With synthetic data	
Feature	**Macro-F1**	**Weighted-F1**	**Macro-F1**	**Weighted-F1**
Moisture	0.62±0.06	0.63±0.06	0.57±0.12	0.60±0.12
Oiliness	0.43±0.02	0.75±0.04	0.44±0.01	0.77±0.02
Elasticity	0.35±0.09	0.46±0.13	0.40±0.07	0.47±0.07
Texture	0.28±0.13	0.45±0.11	0.32±0.14	0.48±0.06
Hyperpigmentation	0.39±0.05	0.52±0.06	0.53±0.20	0.59±0.17

Table 5.3: Performance of **DINOv2** (*frozen*) across features (mean ± std over 3 runs).

	Real only		With synthetic data	
Feature	**Macro-F1**	**Weighted-F1**	**Macro-F1**	**Weighted-F1**
Moisture	0.55±0.06	0.57±0.08	0.56±0.06	0.59±0.07
Oiliness	0.43±0.03	0.76±0.05	0.43±0.04	0.75±0.06
Elasticity	0.27±0.08	0.33±0.11	0.34±0.03	0.45±0.04
Texture	0.25±0.08	0.40±0.08	0.29±0.04	0.45±0.09
Hyperpigmentation	0.55±0.12	0.65±0.04	0.54±0.08	0.71±0.11

Effect of fine-tuning (PanDerm & DINOv2)

Figure 5.2 compare the performances of the models PanDerm and DINOv2 when implemented with frozen embedding extraction vs. when a light fine-tuning wad applied. An horizontal line is overlaid for the most-frequent dummy baseline. Detailed results are tabulated below.

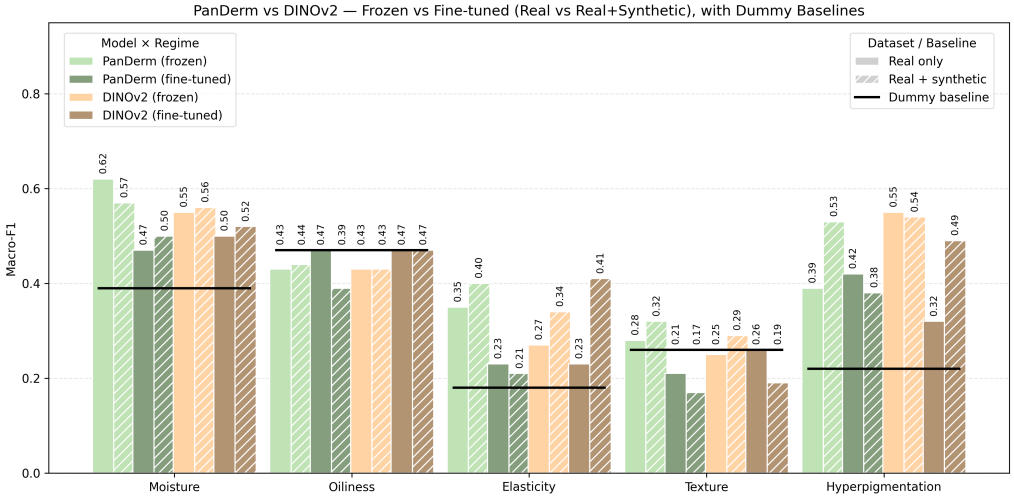


Figure 5.2: Comparison of MacroF1 metrics across models for each feature (Real vs Real+Synthetic data): Frozen embeddings implementation vs finetuned PanDerm/DINOv2

Performance across models (finetuned)

Table 5.4: Performance of **PanDerm** (*fine-tuned*) across features (mean ± std over 3 runs).

	Real only		With synthetic data	
Feature	**Macro-F1**	**Weighted-F1**	**Macro-F1**	**Weighted-F1**
Moisture	0.47±0.13	0.55±0.09	0.50±0.16	0.57±0.13
Oiliness	0.47±0.00	0.82±0.00	0.39±0.04	0.68±0.07
Elasticity	0.23±0.06	0.27±0.10	0.21±0.04	0.25±0.07
Texture	0.21±0.04	0.40±0.07	0.17±0.05	0.26±0.14
Hyperpigmentation	0.42±0.03	0.56±0.04	0.38±0.11	0.52±0.13

Table 5.5: Performance of **DINOv2** (*fine-tuned*) across features (mean ± std over 3 runs).

	Real only		With synthetic data	
Feature	**Macro-F1**	**Weighted-F1**	**Macro-F1**	**Weighted-F1**
Moisture	0.50±0.10	0.56±0.06	0.52±0.08	0.56±0.06
Oiliness	0.47±0.00	0.82±0.00	0.47±0.00	0.82±0.00
Elasticity	0.23±0.07	0.28±0.10	0.41±0.21	0.43±0.14
Texture	0.26±0.00	0.48±0.00	0.19±0.05	0.27±0.16
Hyperpigmentation	0.32±0.11	0.31±0.08	0.48±0.20	0.53±0.14

Dummy baselines

To contextualise performance, we compare all models against three non–learning baselines computed on the training priors:

- **Most–frequent**: always predicts the majority class.
- **Uniform**: samples a class uniformly at random.
- **Stratified (primary chance baseline)**: samples a class according to the empirical class prior.

We treat the stratified baseline as our primary "chance" comparator because it respects the observed imbalance and therefore sets a realistic bar for Macro–F1 under severe class skew. For quantitative comparisons we report

$$\Delta\text{F1} \;=\; \text{Macro-F1}_{\text{model}} \;-\; \text{Macro-F1}_{\text{stratified}},$$

where positive Δ indicates improvement over chance.

Observed baselines. Across features the stratified Macro–F1 is: Moisture 0.47, Oiliness 0.82 (extremely skewed), Elasticity 0.26, Texture 0.33, and Hyperpigmentation 0.30. Figures show the Most-frequent baseline instead for completion. These values are reproduced in Table 5.6. Table 5.7 and Table 5.8 show the computed Δ =s between the best performance of each model per feature and the corresponding stratified baseline.

Table 5.6: Dummy baselines by feature. Stratified is the primary chance baseline; bold marks the highest Macro-F1 per feature.

	Most frequent		Uniform		Stratified	
Feature	**Macro-F1**	**Bal. Acc**	**Macro-F1**	**Bal. Acc**	**Macro-F1**	**Bal. Acc**
Moisture	0.39	0.50	0.42	0.43	**0.47**	0.47
Oiliness	0.47	0.50	0.39	0.36	**0.82**	0.75
Elasticity	0.18	0.33	0.24	0.22	**0.26**	0.31
Texture	0.26	0.33	**0.41**	0.57	0.33	0.35
Hyperpigmentation	0.22	0.33	0.24	0.26	**0.30**	0.32

Results analysis

Table 5.7: Macro-F1 compared to the **stratified** dummy baseline (chance). Δ = model − stratified dummy. Best over (R) real-only and (S) real+synthetic. *Frozen models:* GDF, PanDerm (frozen), DINOv2 (frozen).

Feature	**Stratified**	**GDF best**	Δ	**PanDerm (frozen) best**	Δ	**DINOv2 (frozen) best**	Δ
Moisture	0.47	0.54 (S)	+0.07	0.62 (R)	+0.15	0.56 (S)	+0.09
Oiliness	0.82	0.45 (S)	−0.37	0.44 (S)	−0.38	0.43 (R/S)	−0.39
Elasticity	0.26	0.28 (S)	+0.02	0.40 (S)	+0.14	0.34 (S)	+0.08
Texture	0.33	0.37 (R/S)	+0.04	0.32 (S)	−0.01	0.29 (S)	−0.04
Hyperpigmentation	0.30	0.59 (S)	+0.29	0.53 (S)	+0.23	0.55 (R)	+0.25

Table 5.8: Macro-F1 compared to the **stratified** dummy baseline (chance). Δ = model − stratified dummy. Best over (R) real-only and (S) real+synthetic. *Fine-tuned models:* PanDerm (FT), DINOv2 (FT).

Feature	**Stratified**	**PanDerm (FT) best**	Δ	**DINOv2 (FT) best**	Δ
Moisture	0.47	0.50 (S)	+0.03	0.52 (S)	+0.05
Oiliness	0.82	0.47 (R)	−0.35	0.47 (R/S)	−0.35
Elasticity	0.26	0.23 (R)	−0.03	0.41 (S)	+0.15
Texture	0.33	0.21 (R)	−0.12	0.26 (R)	−0.07
Hyperpigmentation	0.30	0.42 (R)	+0.12	0.48 (S)	+0.18

Overall picture. On this very small and imbalanced dataset, frozen backbones (with linear probes) are generally more reliable than fine–tuning: Finetuning often overfits and degrades Macro–F1. Performance changes via adding synthetic data is *feature–dependent*: neutral-to-positive for Moisture and Hyperpigmentation, while mixed for Elasticity and Texture, and largely unhelpful for Oiliness. Compared to the stratified baseline, we see **clear, clinically meaningful gains** on Moisture and Hyperpigmentation, and modest gains on Elasticity; Texture hovers near chance for every model; Oiliness remains below chance in Macro–F1 due to extreme priors.

Per–feature findings.

- **Moisture.** Best Macro–F1 0.62 with *PanDerm (frozen, real)*; $\Delta = +0.15$ over chance. All backbones outperform the baseline, with small variance and limited sensitivity to synthetic data. This task appears learnable with current supervision.

- **Oiliness.** The stratified baseline is unusually high (0.82) because the positive class is extremely rare. All models sit near 0.43–0.47 Macro–F1 (below chance), while *Weighted–F1* is high, indicating majority-class dominance. Improving minority recall will likely require class–balanced or focal losses, minority oversampling, and threshold tuning; treating this as *positive–class detection* with PR–AUC might be more appropriate.
- **Elasticity.** This has proved to be the hardest task overall. *PanDerm (frozen, synth)* reaches 0.40 ($\Delta = +0.14$) and *DINOv2 (FT, synth)* reaches 0.41 ($\Delta = +0.15$), suggesting some benefit from synthetic augmentation and limited, carefully constrained FT. The colour/illumination artefacts in these images (variation of purple/pink) likely confound features; colour–invariant augmentations or channel normalisation should help.
- **Texture.** Performance hovers near chance across backbones (best ≈ 0.37; $\Delta \in [-0.04, +0.04]$). Projecting the 3D surface pattern into 2D leaves weak, scale–dependent cues that are easily swamped by noise and illumination changes. Despite feature–specific augmentations during training, the signal remains under–represented; more data and texture–centric strategies—e.g., multi–crop / patch–based training, high–frequency or local–contrast emphasis, and explicit scale jitter—are likely required to improve performance.
- **Hyperpigmentation.** This feature shows consistent gains over chance across models. Best 0.59 with *GDF (frozen, synth)* ($\Delta = +0.29$) and competitive 0.55 with *DINOv2 (frozen, real)* ($\Delta = +0.25$). This represent the feature with the most balanced classes (8, 44, 25) and the most visible skin condition, here synthetic data often helps but introduces volatility, hinting at variable synthetic quality.

Frozen vs. fine–tuned. In this data regime, a frozen self–supervised backbone with a linear classifier acts as a low–variance estimator that preserves the inductive bias of pretraining; fine–tuning, by contrast, introduces many more degrees of freedom than the minority classes can constrain, which typically manifests as calibration drift and threshold instability (often before a clear drop in overall accuracy). Gains from fine–tuning appear only when there is genuine feature–space misalignment and the effective sample size is augmented (e.g., *Elasticity* with DINOv2 when synthetic images); otherwise, variance dominates bias. When Fine-tuning is attempted, a direction to investigate woult be *last–block unfreeze*, strong regularisation, early stopping on Macro–F1, and post–hoc calibration/threshold tuning. Absent a demonstrated misalignment, a calibrated linear probe on frozen features provides the best risk–performance trade–off for this small, imbalanced setting.

6 Conclusion

Often companies aim to implement advanced technologies but do not have the resources and data to efficiently implement them. This situation is particularly emphasized in the healthcare (and adjacent) sector where the current research aims at classifying a condition and suggest the appropriate cure or treatment to improve understanding and prevent the development of more serious conditions. Nevertheless, it is hard to collect data that actually present serious levels of the conditions which makes the task of learning its presence harder.

This thesis set out to test whether generative augmentation and foundation models can make cosmetic-dermatology classification feasible under the "real world" constraints that small clinics face: scarce, noisy, and highly imbalanced data. Diffusion-based augmentation successfully increased the volume of training images but—by design—added limited variability, because we intentionally kept generations close to the originals to preserve clinical plausibility. As a result, the dataset became larger but not markedly more informative for learning minority classes. Across models, the most reliable recipe in this regime was a frozen backbone with a linear head: fine-tuning frequently overfit and reduced Macro-F1 on the very small, skewed test sets. Performance gains were strongly feature-dependent. We observed clear improvements over the stratified "chance" baseline for *Moisture* and *Hyperpigmentation*(e.g., PanDerm-frozen on Moisture, and GDF-frozen with synthetic data on Hyperpigmentation), modest gains for *Elasticity*, near-chance behaviour for *Texture*, and below-chance Macro-F1 for *Oiliness* due to extreme label skew (despite high Weighted-F1 reflecting majority dominance). Domain-specific backbones (Google Derm Foundation; PanDerm) transfer only partially to cosmetic settings because their pre-training emphasises medical pathology under controlled illumination, while cosmetic cues often depend on subtle reflectance and lighting variations. A generalist model (DINOv2) proved competitive: its frozen features matched or exceeded domain-specific models on several tasks and, when lightly fine-tuned with synthetic data, showed the clearest upside on *Elasticity*; otherwise, frozen probes remained the safer choice. Overall, despite severe data limitations, results consistently exceeded dummy baselines on multiple features, indicating that foundation-model backbones plus careful evaluation can be a viable starting point for accessible, clinic-grade decision support in cosmetic dermatology.

Limitations

This study is constrained first and foremost by data. Classes are strongly imbalanced with tiny minority supports (in some test splits only 1–2 images), making metrics volatile and generalisation brittle; outcomes often hinged on whether a single minority sample was correctly classified. Image acquisition was non-professional: many images contained glare, blur, hair or moles, and we could not trace the original labelling protocol or access raw, unlabeled images to re-grade severity levels—factors that likely confused the models. Because we are not dermatology experts, we

adopted conservative generation settings to avoid implausible artefacts; this limited synthetic diversity, especially for classes with only a few exemplars, and traditional augmentations tended to inject noise rather than signal (we therefore kept only mild, on-the-fly transforms at training time). Computational budget further restricted diffusion fine-tuning, exhaustive hyper-parameter search, and end-to-end adaptation of the most promising foundation models (e.g., Google Derm), forcing us to rely primarily on frozen embeddings. Finally, the dataset lacks demographic coverage and metadata (skin tone, age, lighting conditions), preventing a meaningful assessment of fairness, calibration, and colour sensitivity—central considerations for any deployment in healthcare or adjacent settings.

Future work

A natural next step concerns **data quality and coverage**. Progress is likely to come from standardising image capture, controlling illumination, distance, and focus,ideally with an improved scanning device that records raw images and rich metadata. A better quality data extraction combined with a larger and more diverse cohort would enable balanced test sets and meaningful subgroup analyses. Moreover, re-labelling under expert supervision, guided by a transparent grading rubric and complemented by multi-rater adjudication, would help quantify and reduce label noise.

On the **modelling** side, the most skewed targets (notably *Oiliness*) are better understood as rare-event detection problems, for which imbalance-aware objectives and sampling strategies tend to be more informative. In such settings, calibration and decision-threshold selection become central, and a deeper dive into PR-AUC reports alongside Macro-F1 could provide a fuller picture of the operating characteristics.

For **generation**, fine-tuning diffusion models,where computational budgets permit, appears promising, particularly with class-conditional guidance, exemplar-based editing, and automated quality filters. Diversity would be most productively directed toward minority classes and difficult visual cues rather than uniform upsampling. A colour-dependence study (e.g., grayscale ablations, channel-drop/jitter sensitivity, illumination-normalisation baselines) would expose reliance on spurious chromatic signals and help surface potential biases.

Where licensing and resources allow, an end-to-end evaluation of Google Derm Foundation in this cosmetic setting would be informative, both as separate single-feature learners and as a multi-task head sharing a common backbone. In parallel, fairness and calibration should be assessed across skin tone, age, and gender, with reliability diagrams and per-class PR curves making operating-point trade-offs explicit. Active-learning protocols could then prioritise expert annotation for the most informative or uncertain samples. Taken together, these directions address the principal bottlenecks identified here (data fidelity and coverage, extreme imbalance, and domain shift) and outline a path from proof-of-concept to robust, well-calibrated systems suitable for cosmetic dermatology in everyday practice.

Bibliography

Hash, M. G., Forsyth, A., Coleman, B.-A., Li, V., Vinagolu-Baur, J., & Frasier, K. M. (2025). Artificial intelligence in the evolution of customized skincare regimens. *Cureus*, *17*(4), e82510. https://doi.org/10.7759/cureus.82510

L'Oréal. (2024a). *L'oréal and modiface: An artificial intelligence-powered skin diagnostic*. Retrieved October 27, 2024, from https://www.loreal.com/en/news/research-innovation/loreal-and-modiface-an-artificial-intelligencepowered-skin-diagnostic/?utm_medium=email&utm_source=transaction

Kania, B., Montecinos, K., & Goldberg, D. J. (2024). Artificial intelligence in cosmetic dermatology [Epub ahead of print Aug 27, 2024]. *Journal of Cosmetic Dermatology*, *23*(10), 3305–3311. https://doi.org/10.1111/jocd.16538

Alamer, A., Aldhafeeri, F., Alghamdi, S., Alanazi, E., Alsubaie, M., & Alanazi, M. (2023). Impact of social media influencers on skincare consumer behavior: A cross-sectional survey. *Journal of Cosmetic Dermatology*, *22*(5), 1425–1434. https://doi.org/10.1111/jocd.15478

L'Oréal. (2024b). *Unveil perso: The world's first ai-powered device for skincare and cosmetics*. Retrieved October 27, 2024, from https://www.loreal.com/en/news/research-innovation/unveil-perso-the-worlds-first-aipowered-device-for-skincare-and-cosmetics/?utm_medium=email&utm_source=transaction

Proven Skincare. (2024). *The skin genome project*. Retrieved October 27, 2024, from https://www.provenskincare.com/why-proven/?utm_medium=email&utm_source=transaction

Haut.AI. (2024). *Ai skin analysis & personalized product recommendation for beauty brands*. Retrieved October 27, 2024, from https://haut.ai/

L'Oréal. (2024c). *L'oréal at vivatech 2024: Beauty tech innovation*. Retrieved October 27, 2024, from https://www.loreal.com/en/press-release/research-and-innovation/vivatech-2024/?utm_source=chatgpt.com

Yan, S., Yu, Z., Primiero, C., Vico-Alonso, C., Wang, Z., Yang, L., Tschandl, P., Hu, M., Tan, G., Tang, V., et al. (2025). A multimodal vision foundation model for clinical dermatology. *Nature Medicine*. https://doi.org/10.1038/s41591-025-03747-y

Rikhye, R. V., Loh, A., Hong, G. E., Singh, P., Smith, M. A., Muralidharan, V., Wong, D., ..., Matias, Y., & Corrado, G. S. (2024). Learning clinical representations from large-scale dermatology image-text data [Available as part of the Health AI Developer Foundations ("Derm Foundation") release]. *arXiv preprint*, *arXiv:2411.15128*. https://arxiv.org/abs/2411.15128v1

Oquab, M., Darcet, T., Moutakanni, T., Vo, H. V., Szafraniec, M., Khalidov, V., Fernandez, P., Haziza, D., Massa, F., El-Nouby, A., Assran, M., Ballas, N., Galuba, W., Howes, R., Huang, P.-Y., Li, S.-W., Misra, I., Rabbat, M., Sharma, V., ... Mairal, J. (2024). Dinov2: Learning robust visual features without supervision. *Transactions on Machine Learning Research*. https://arxiv.org/abs/2304.07193

Mietkiewicz, L., Ciechanowski, L., & Jemielniak, D. (2025). The skin game: Revolutionizing standards for ai dermatology model comparison. https://arxiv.org/abs/2502.02500

Perez, F., Vasconcelos, C. N., Avila, S., & Valle, E. (2018). Data augmentation for skin lesion analysis [arXiv:1809.01442]. *ISIC Skin Image Analysis Workshop @ MICCAI*. https://arxiv.org/abs/1809.01442

Aladhadh, S., Alhumyani, H., Alotaibi, B., et al. (2022). An effective skin cancer classification mechanism via medical vision transformer. *Sensors*, *22*(11), 4008. https://doi.org/10.3390/s22114008

Krishna, A., Kumar, R., Gupta, D., et al. (2023). Lesionaid: Learning from synthetic skin lesions. https://arxiv.org/abs/2307.07936

Qin, Z., Chen, Z., Wang, Z., & Zhang, Z. (2020). Style-based gans for skin lesion augmentation: Improving skin lesion classification via synthetic images. *Medical Image Analysis*, *65*, 101802. https://doi.org/10.1016/j.media.2020.101802

Akrout, M., Gyepesi, B., Holló, P., Poór, A., Kincső, B., Solis, S., Cirone, K., Kawahara, J., Slade, D., Abid, L., et al. (2023). Diffusion-based data augmentation for skin disease classification: Impact across original medical datasets to fully synthetic images. In A. Mukhopadhyay, I. Oksuz, S. Engelhardt, D. Zhu, & Y. Yuan (Eds.), *Deep generative models* (pp. 99–109, Vol. 14533). Springer Nature Switzerland AG. https://doi.org/10.1007/978-3-031-53767-7_10

Wang, P., Zhang, H., Zhang, Z., Chen, S., Ma, Y., & Qu, Q. (2024). Diffusion models learn low-dimensional distributions via subspace clustering. *arXiv preprint*, *arXiv:2409.02426*. Retrieved September 14, 2025, from https://arxiv.org/abs/2409.02426

Sagers, S., Wu, H., Tu, T., et al. (2022). Improving skin tone diversity in dermatology datasets using generative ai. https://arxiv.org/abs/2211.02155

Ktena, I., Wiles, O., Albuquerque, I. A., Roberts, A., Belgrave, D., Cemgil, T., Karthikesalingam, A., & Gowal, S. (2024). Generative models improve fairness of medical classifiers under distribution shifts. *Nature Medicine*, *30*, 1166–1173. https://doi.org/10.1038/s41591-024-02838-6

Ho, J., Jain, A., & Abbeel, P. (2020). Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems (NeurIPS)*. https://proceedings.neurips.cc/paper/2020/file/4c5bcfec8584af0d967f1ab10179ca4b-Paper.pdf

Dhariwal, P., & Nichol, A. (2021). Diffusion models beat gans on image synthesis. *Advances in Neural Information Processing Systems (NeurIPS)*. https://proceedings.neurips.cc/paper/2021/file/49ad23d1ec9fa4bd8d77d02681df5cfa-Paper.pdf

Rombach, R., Blattmann, A., Lorenz, D., Esser, P., & Ommer, B. (2022). High-resolution image synthesis with latent diffusion models. *CVPR*. https://arxiv.org/abs/2112.10752

Bommasani, R., Hudson, D. A., Adeli, E., Altman, R., et al. (2021). On the opportunities and risks of foundation models. *arXiv:2108.07258*. https://arxiv.org/abs/2108.07258

Google Health AI. (2025). Derm foundation model card [Accessed 11 Sep 2025].

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., & Polosukhin, I. (2017). Attention is all you need. *Advances in Neural Information Processing Systems (NeurIPS)*. https://arxiv.org/abs/1706.03762

Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., & Houlsby, N. (2021). An image is worth 16x16 words: Transformers for image recognition at scale. https://arxiv.org/abs/2010.11929

Caron, M., Touvron, H., Misra, I., Jégou, H., Mairal, J., Bojanowski, P., & Joulin, A. (2021). Emerging properties in self-supervised vision transformers. *ICCV*. https://openaccess.thecvf.com/content/ICCV2021/papers/Caron_Emerging_Properties_in_Self-Supervised_Vision_Transformers_ICCV_2021_paper.pdf

Kloster, M. (2024). Dinov2: Exploring self-supervised vision transformers. Retrieved May 16, 2023, from https://blog.marvik.ai/2023/05/16/dinov2-exploring-self-supervised-vision-transformers/

Meta AI. (2023). Dino-paws: Computer vision with self-supervised transformers and 10× more efficient training. Retrieved September 15, 2025, from https://ai.meta.com/blog/dino-paws-computer-vision-with-self-supervised-transformers-and-10x-more-efficient-training/

Zhou, J., Wei, C., Wang, H., Shen, W., Xie, C., Yuille, A., & Kong, T. (2021). Ibot: Image bert pre-training with online tokenizer. *arXiv:2111.07832*. https://arxiv.org/abs/2111.07832

Xie, Z., Zhang, Z., Cao, Y., Lin, Y., Bao, J., Yao, Z., Dai, Q., & Hu, H. (2021). Simmim: A simple framework for masked image modeling [Submitted Nov 18, 2021; Revised Apr 17, 2022]. *arXiv preprint*, *arXiv:2111.09886*. https://arxiv.org/abs/2111.09886

Kolesnikov, A., Beyer, L., Zhai, X., Puigcerver, J., Yung, J., Gelly, S., & Houlsby, N. (2020). Big transfer (bit): General visual representation learning. *ECCV*. https://www.ecva.net/papers/eccv_2020/papers_ECCV/papers/123500477.pdf

LeCun, Y., Bottou, L., Bengio, Y., & Haffner, P. (1998). Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, *86*(11), 2278–2324. https://doi.org/10.1109/5.726791

He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778.

van den Oord, A., Li, Y., & Vinyals, O. (2018). Representation learning with contrastive predictive coding. *Advances in Neural Information Processing Systems (NeurIPS) Workshop*.

Radford, A., Kim, J. W., Hallacy, C., Ramesh, A., Goh, G., Agarwal, S., Sastry, G., Askell, A., Mishkin, P., Clark, J., Krueger, G., & Sutskever, I. (2021). Learning transferable visual models from natural language supervision. *arXiv:2103.00020*. https://arxiv.org/abs/2103.00020

Zhang, Y., Jiang, H., Miura, Y., Manning, C. D., & Langlotz, C. P. (2022). Contrastive learning of medical visual representations from paired images and text. *Proceedings of the 3rd Machine Learning for Healthcare Conference (MLHC)*, *182*, 2–25. https://proceedings.mlr.press/v182/zhang22a.html

He, K., Chen, X., Xie, S., Li, Y., Dollár, P., & Girshick, R. (2022). Masked autoencoders are scalable vision learners. *CVPR*. https://openaccess.thecvf.com/content/CVPR2022/html/He_Masked_Autoencoders_Are_Scalable_Vision_Learners_CVPR_2022_paper.html

Georgievskaya, A., Tlyachev, T., Danko, D., et al. (2025). How artificial intelligence adopts human biases: The case of cosmetic skincare industry. *AI and Ethics*, *5*, 105–115. https://doi.org/10.1007/s43681-023-00378-2

Aram HUVIS Co., L. (2025). Advanced AI scalp skin analysis solutions – aram huvis. Retrieved September 13, 2025, from https://www.aramhuvis.com/

Stable Diffusion v1-5. (2022). Stable diffusion v1.5 [Hugging Face model card. Canonical mirror of the formerly hosted `runwayml/stable-diffusion-v1-5`.].

von Platen, P., Patil, S., Lozhkov, A., Cuenca, P., Lambert, N., Rasul, K., Davaadorj, M., Nair, D., Paul, S., Berman, W., Xu, Y., Liu, S., & Wolf, T. (2022). Diffusers: State-of-the-art diffusion models.

Karras, T., Aittala, M., Laine, S., Härkönen, E., Hellsten, J., Lehtinen, J., & Aila, T. (2022). Elucidating the design space of diffusion-based generative models. *Advances in Neural Information Processing Systems (NeurIPS)*.

Lu, C., Zhou, Y., Bao, F., Chen, J., Li, C., & Zhu, J. (2022). Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *Advances in Neural Information Processing Systems (NeurIPS)*.

Verzino, G. (2021). Why balancing classes is over-hyped [Towards Data Science, Medium].

Arabboev, M., Begmatov, S., Rikhsivoev, M., Nosirov, K., & Saydiakbarov, S. (2024). A comprehensive review of image super-resolution metrics: Classical and ai-based approaches. *Acta IMEKO*, *13*(1), 1–8. https://doi.org/10.21014/actaimeko.v13i1.1679

Mosley, L., Halcrow, P., & Wang, Y. (2022). On the importance of f1 score for imbalanced datasets. *Expert Systems with Applications*, *205*, 117889. https://doi.org/10.1016/j.eswa.2022.117889

Chicco, D., & Jurman, G. (2020). The advantages of the matthews correlation coefficient (mcc) over f1 score and accuracy in binary classification evaluation. *BMC Genomics*, *21*(1), 6. https://doi.org/10.1186/s12864-019-6413-7

Yang, Y., & Loog, M. (2018). A benchmark and comparison of active learning for logistic regression. *Pattern Recognition*, *83*, 401–415. https://doi.org/10.1016/j.patcog.2018.05.005

Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on image data augmentation for deep learning. *Journal of Big Data*, *6*(1), 60. https://doi.org/10.1186/s40537-019-0197-0

Yu, K., et al. (2023). Deep learning with test-time augmentation for radial endobronchial ultrasound (rebus) images. *Scientific Reports*, *13*(1), 10924. https://doi.org/10.1038/s41598-023-37941-0

Maalouf, M. (2011). Logistic regression in data analysis: An overview. *International Journal of Data Analysis Techniques and Strategies*, *3*(3), 281–299. https://doi.org/10.1504/IJDATS.2011.041335

Yan, S., & colleagues. (2024). Panderm: Towards a general-purpose foundation model for dermatology. *arXiv preprint*.

Wightman, R. (2023). *Pytorch image models (timm) documentation*. Retrieved September 14, 2025, from https://huggingface.co/docs/timm

Wang, H., Wang, Y., Zhou, Z., Ji, X., Gong, D., Zhou, J., Li, W., & Liu, T. (2018). Cosface: Large margin cosine loss for deep face recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 5265–5274.

Deng, J., Guo, J., Xue, N., & Zafeiriou, S. (2019). Arcface: Additive angular margin loss for deep face recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 4690–4699.

TorchVision Contributors. (2023a). *Torchvision.transforms.randomresizedcrop*. Retrieved September 14, 2025, from https://pytorch.org/vision/stable/transforms.html#torchvision.transforms.RandomResizedCrop

TorchVision Contributors. (2023b). *Torchvision.transforms.randomhorizontalflip*. Retrieved September 14, 2025, from https://pytorch.org/vision/stable/transforms.html#torchvision.transforms.RandomHorizontalFlip

Chen, J., Zhang, H., & Sun, L. (2024). A survey on class imbalance in computer vision datasets. *ACM Computing Surveys*.

Loshchilov, I., & Hutter, F. (2019). Decoupled weight decay regularization. *International Conference on Learning Representations (ICLR)*. https://arxiv.org/abs/1711.05101

PyTorch Contributors. (2023). *Crossentropyloss*. Retrieved September 15, 2025, from https://docs.pytorch.org/docs/stable/generated/torch.nn.CrossEntropyLoss.html

Müller, R., Kornblith, S., & Hinton, G. (2020). When does label smoothing help? https://arxiv.org/abs/1906.02629

Pascanu, R., Mikolov, T., & Bengio, Y. (2013). On the difficulty of training recurrent neural networks. https://arxiv.org/abs/1211.5063

Prechelt, L. (1998). Early stopping – but when? *Neural Networks: Tricks of the Trade*, 55–69.

TorchVision Contributors. (2023c). *Torchvision.transforms.fivecrop*. Retrieved September 14, 2025, from https://pytorch.org/vision/stable/transforms.html#torchvision.transforms.FiveCrop

Guo, C., Pleiss, G., Sun, Y., & Weinberger, K. Q. (2017). On calibration of modern neural networks. *International Conference on Machine Learning (ICML)*, 1321–1330.

Colorimetry — part 4: CIE 1976 $l^*a^*b^*$ colour space. (2019). International Organization for Standardization and International Commission on Illumination (CIE). https://www.iso.org/standard/74166.html

A Appendix

A.1 Preliminary material

Self-Supervised Learning. Self-supervised learning (SSL) is a learning paradigm where a model is trained on unlabeled data by solving a surrogate task that provides supervision inherently. In essence, the model creates its own training labels from the data. This approach has been pivotal in the development of foundation models, including those in our study, because it enables learning from vast quantities of images without the need for manual annotations. In computer vision, several types of self-supervised objectives have proven effective. We cite:

- Contrastive Learning. The model learns to associate multiple views of the same image while distinguishing them from views of other images. Methods like MoCo He et al., 2022 fall in this category.
- Knowledge Distillation / Redundancy Reduction. This approach is exemplified by DINO Caron et al., 2021.
- Masked Image Modeling. Inspired by masked language modeling in NLP (e.g., BERT), methods like MAE (Masked Autoencoder, He et al., 2022) randomly mask patches of the input image and task the network to reconstruct the missing content.
- Multi-modal Alignment. A form of self-supervision where two different modalities of the same underlying data are used. CLIP Radford et al., 2021 is a prominent example that trains an image encoder and a text encoder jointly to align image and text embeddings for the same item. In vision, this can be seen as self-supervised from the standpoint that no manual label is given—only the pairing of image and caption.

Logistic regression. Logistic regression models the conditional probability of a binary label $y \in \{0, 1\}$ given a feature vector $x \in \mathbb{R}^d$ via the logistic link:

$$p_\theta(y = 1 \mid x) \;=\; \sigma(z) \;=\; \frac{1}{1 + \exp(-z)}, \qquad z \;=\; w^\top x + b,$$

with parameters $\theta = (w, b)$. Parameters are learned by minimizing the (regularized) negative log-likelihood over n examples $\{(x_i, y_i)\}_{i=1}^n$:

$$\mathcal{L}(\theta) \;=\; -\frac{1}{n} \sum_{i=1}^{n} \Big[y_i \log p_\theta(y_i{=}1 \mid x_i) + (1 - y_i) \log\Big(1 - p_\theta(y_i{=}1 \mid x_i)\Big) \Big] \;+\; \lambda \|w\|_2^2,$$

optionally with class weights to address imbalance. The decision rule is $\hat{y} = \mathbb{1}\{p_\theta(y{=}1 \mid x) \geq t\}$ with a tunable threshold $t \in (0, 1)$. For K classes, the softmax generalization uses class scores $z_k = w_k^\top x + b_k$ and

$$p_\theta(y{=}k \mid x) \;=\; \frac{\exp(z_k)}{\sum_{j=1}^{K} \exp(z_j)}, \quad \mathcal{L}(\theta) \;=\; -\frac{1}{n} \sum_{i=1}^{n} \log p_\theta(y_i \mid x_i) \;+\; \lambda \sum_{k=1}^{K} \|w_k\|_2^2.$$

Logistic regression is convex in θ, trained with (stochastic) gradient methods, and yields well-calibrated scores under the model assumptions.

Convolutional neural networks (CNNs). CNNs learn hierarchical representations from images by composing convolution, nonlinearity, and (optionally) pooling. A 2D convolutional layer with C_{in} input channels and C_{out} filters applies shared kernels $W^{(c)} \in \mathbb{R}^{k\times k\times C_{\text{in}}}$ and biases b_c to produce feature maps

$$Y_c(i,j) \;=\; \phi\!\left(\sum_{u=1}^{k}\sum_{v=1}^{k}\sum_{r=1}^{C_{\text{in}}} W^{(c)}_{u,v,r}\, X_{i+u-1,\,j+v-1,\,r} \;+\; b_c\right), \qquad c = 1,\dots,C_{\text{out}},$$

where ϕ is a pointwise nonlinearity (e.g., ReLU). Stride and padding control resolution; optional pooling (e.g., max/average) aggregates locally to build translation invariance. Stacking layers grows the receptive field and yields increasingly abstract features. After several blocks, a classifier head (global average pooling and/or fully connected layer) outputs class scores z_k, trained end-to-end by minimizing cross-entropy

$$\mathcal{L}(\theta) \;=\; -\frac{1}{n}\sum_{i=1}^{n} \log \frac{\exp(z_{k_i}(x_i))}{\sum_j \exp(z_j(x_i))}$$

with regularization techniques such as weight decay, batch normalization, and dropout. Parameter sharing in convolutions confers strong inductive bias (locality and translation equivariance), making CNNs data-efficient for vision tasks relative to dense networks.

Multi–Layer Perceptron (MLP) head. An MLP is a feed-forward network that maps an input vector $h_0 \in \mathbb{R}^{d_0}$ to an output h_L through a stack of affine layers interleaved with pointwise nonlinearities:

$$h_{\ell+1} \;=\; \phi\big(W_\ell h_\ell + b_\ell\big), \qquad \ell = 0,\dots,L-1,$$

where W_ℓ and b_ℓ are learnable parameters and ϕ is a non-linear activation. In our classification head we use a compact, regularised MLP often termed a *StrongHead*: a LayerNorm–Linear projection to width 1024 with GELU activation and dropout ($p = 0.35$), followed by two *pre-norm residual* MLP blocks, and a cosine (angular) classifier. A pre-norm residual block applies LayerNorm before the block MLP and adds a skip connection

$$h \leftarrow h + \mathrm{MLP}\big(\mathrm{LN}(h)\big),$$

which improves optimisation stability at small data scale. The cosine classifier normalises both features and class weights and uses a learnable scale s,

$$z_k \;=\; s\,\frac{h}{\|h\|}\cdot\frac{w_k}{\|w_k\|} \;=\; s\,\cos\big(\angle(h,w_k)\big),$$

producing logits z_k whose decision boundaries depend on angles rather than raw norms, typically yielding better calibration and robustness.

GELU activation. The Gaussian Error Linear Unit (GELU) gates inputs according to their probability of being positive under a standard normal variable:

$$\mathrm{GELU}(x) \;=\; x\,\Phi(x),$$

where Φ is the CDF of $\mathcal{N}(0,1)$. A common smooth approximation is

$$\mathrm{GELU}(x) \;\approx\; \tfrac{1}{2}x\Big(1+\tanh\Big(\sqrt{2/\pi}\,\big(x+0.044715\,x^3\big)\Big)\Big).$$

Unlike ReLU, GELU is differentiable everywhere and softly suppresses small negative inputs while preserving large positive ones. This smooth gating has proved effective in modern vision and language encoders, leading to stable optimisation and strong performance when paired with LayerNorm and residual MLP blocks as above.

A.2 Dataset specification

Classified skin conditions report

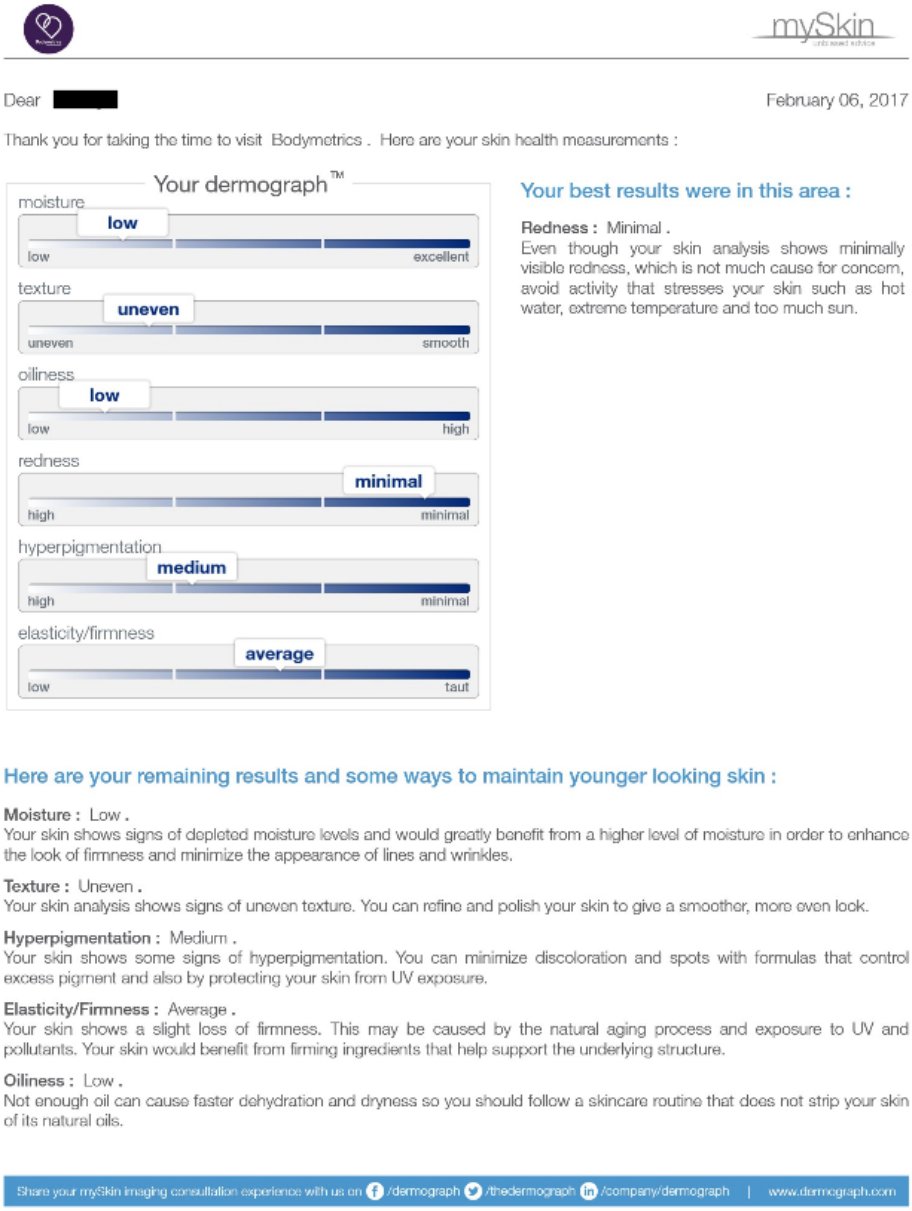


Figure A.1: Sample of output report produced by the mySkin UK software for skin classification.

Figure A.1 show an example of the report produced by the mySkin software. Each skin condition is remapped into a three level class and, based on its specifics. High level information is provided regarding each level's meaning. This poses as a base for recommendation of cure/products. In our study the feature *Redness* is excluded for total absence of samples in the high-level class.

Skin tone assignment rule

This work refers to the skin tone groups as classified by the CIE L*a*b* (CIELAB) colour space as a perceptually oriented, device-independent representation of appearance. In CIELAB, L^* encodes lightness on a 0–100 scale (black to white), while a^* and b^* describe opponent chromatic axes (green→red and blue→yellow, respectively). Because Euclidean distances in this space correlate more closely with perceived differences than in RGB, simple rules become meaningful for complexion analysis. Following our assignment rule, very light skin is characterised by high L^* values and very dark skin by lower L^*; human skin typically occupies the quadrant with positive a^* (pinkish) and positive b^* (yellowish), with practical thresholds in our dataset around $a^* \in [5, 25]$ and $b^* \in [5, 30]$. This choice enables a transparent mapping from measured colour to skin-tone categories while remaining grounded in a standardised colourimetric framework ("Colorimetry — Part 4: CIE 1976 L*a*b* colour space", 2019).

Train-Test split algorithm

Algorithm 1: Label–aware patient split with minimum per–bin test coverage

Input : Wide CSV $\mathcal{D}$ with columns `patient_id`, feature image paths x_f, and labels $y_f \in \{-1, 0, 1\}$ for features $f \in \mathcal{F}$; mapping `feature2imgcol`; test fraction τ; seed s; file–existence check flag; minimum support threshold m.

Output : Disjoint *train* and *test* row sets; selected test patient IDs.

1. Preprocessing.
Set RNG with seed s. Optionally drop rows where image files are missing. For each feature $f \in \mathcal{F}$, coerce y_f to integer when present.

2. Build per–patient and global bin counts.
For each row $r \in \mathcal{D}$ with patient p: for each feature $f \in \mathcal{F}$, if $y_f(r)$ is defined, update $\texttt{patient\_bins}[p][\![f, y_f]\!]$ and $\texttt{global\_bins}[\![f, y_f]\!]$.

3. Define minimum test coverage per bin.
For each bin $b = (f, c)$ in `global_bins`, set requirement $\texttt{req}[b] \leftarrow \mathbb{1}\{\texttt{global\_bins}[b] \geq m\}$ (i.e., ensure at least one test example if total support is $\geq m$).

4. Greedy cover of required bins (patient selection).
Let P be the list of unique patients, shuffled once with RNG. Initialize empty set $\mathcal{T}$ (test patients) and counter $\texttt{covered}[b] \leftarrow 0$.
Define patient gain
$\text{gain}(p) = \sum_b \mathbb{1}\{\texttt{req}[b] = 1 \wedge \texttt{covered}[b] < \texttt{req}[b] \wedge \texttt{patient\_bins}[p][b] > 0\}$.
while *some* b *has* $\texttt{covered}[b] < \texttt{req}[b]$ **do**
  pick $p^\star = \arg\max_{p \in P} \text{gain}(p)$
  if $\text{gain}(p^\star) \leq 0$ **then**
    break (insufficient support to meet all requirements).
  add $p^\star$ to $\mathcal{T}$; for every b with $\texttt{patient\_bins}[p^\star][b] > 0$ and $\texttt{covered}[b] < \texttt{req}[b]$, set $\texttt{covered}[b] \leftarrow \texttt{covered}[b] + 1$; remove $p^\star$ from P.

5. Fill to target test size.
Let $n_{\text{target}} \leftarrow \max(1, \lceil \tau \cdot |\text{unique patients}| \rceil)$.
Shuffle remaining P and add patients to $\mathcal{T}$ until $|\mathcal{T}| = n_{\text{target}}$.

6. Materialize split and checks.
Assign a row to *test* iff its $\texttt{patient\_id} \in \mathcal{T}$; otherwise to *train*. Assert no patient overlap. Optionally save CSVs and report per–feature, per–class counts.

Notes. The greedy cover enforces at least one test example for each (f, c) with sufficient support ($\geq m$) while keeping a patient-wise split and approximate test fraction τ.

Table A.1: Train counts per feature and class (rowsum = 61).

Feature	**-1**	**0**	**1**	**Total**
Texture	6	39	16	61
Hyperpigmentation	6	36	19	61
Oiliness	58	1	2	61
Moisture	42	18	1	61
Elasticity	4	42	15	61

Table A.2: Test counts per feature and class (rowsum = 16).

Feature	**-1**	**0**	**1**	**Total**
Texture	2	10	4	16
Hyperpigmentation	2	8	6	16
Oiliness	14	2	0	16
Moisture	10	6	0	16
Elasticity	2	6	8	16

A.3 Prompts example

Prompt construction and tokenization. For Stable Diffusion v1.5 `img2img`, we construct each positive prompt as *base (feature-specific) + class descriptor + a few feature additions* (e.g., sharp microtexture for Texture, recoil vs. oil-shine for Elasticity). To keep the appendix compact, Table A.3 lists only the class-conditioned descriptors, and Table A.4 lists the negative prompts (per feature). At generation time, the combined positive and negative prompts are trimmed by the model tokenizer to a maximum of 75 tokens. The exact, pre-truncation strings used for every image are logged in our run artifacts (see `oversample_log_2.csv`).

Feature	Negative prompt
Moisture	face, eyes, nose, mouth, hair, portrait, selfie, text, logo, watermark, arrows, grid, legend, scale bar, numbers, excessive blur, heavy noise, oversaturated, sweat droplets, water droplets, oily glare, lotion smears, blue illumination, blue tint.
Oiliness	face, eyes, nose, mouth, hair, portrait, selfie, text, logo, watermark, arrows, grid, legend, scale bar, numbers, excessive blur, heavy noise, oversaturated, heatmap, colormap, contour overlay, uniform blue wash, blue fog, posterization, overexposed, blown highlights, hotspot, bloom, lens flare, vignette, flashlight beam, banding, tiling, haze, overglow.
Elasticity	face, eyes, nose, mouth, hair, portrait, selfie, text, logo, watermark, arrows, grid, legend, scale bar, numbers, excessive blur, heavy noise, oversaturated, oily glare, sweat droplets, deep wrinkles, surgical tape marks.
Texture	face, eyes, nose, mouth, hair, portrait, selfie, text, logo, watermark, arrows, grid, legend, scale bar, numbers, excessive blur, heavy noise, oversaturated, flat uniform surface.
Hyperpigmentation	face, eyes, nose, mouth, hair, portrait, selfie, text, logo, watermark, arrows, grid, legend, scale bar, numbers, excessive blur, heavy noise, oversaturated, blue tint, solid blue blobs, filled blue shapes, heatmap, colormap.

Table A.4: Negative prompts concatenated as in code (base list plus feature-specific additions). These remain constant across classes for a given feature.

Bases used: Moisture — neutral non-blue clinical close-up; Oiliness — blue-illumination dermatology imaging; Elasticity/Texture/Hyperpigmentation — clinical macro close-up with high-detail microtexture.

Feature	Class	Prompt description
Moisture	−1	Low moisture; dehydrated; matte; fine lines and micro-cracks pronounced; narrow highlights.
	0	Average moisture; balanced hydration; soft highlights; clear microtexture.
	1	High moisture; plump; diffuse sheen; broader low-contrast highlights; fine lines reduced; keep input skin tone.
Oiliness	−1	Low oiliness; matte; sparse faint blue micro-specular points ($< 5\%$ area); microtexture clearly visible.
	0	Balanced oil; soft luster; scattered small blue highlights (10–20% area); microtexture visible.
	1	High oiliness; glossy; coherent blue specular regions (30–50% area) but pores and lines still visible.
Elasticity	−1	Low elasticity; slack microfolds; creases persist; broad dull highlights.
	0	Average elasticity; balanced micro-relief; moderate highlights.
	1	High elasticity; taut microtexture; minimal creasing; tighter brighter highlights.
Texture	−1	Rough texture; visible pores and micro ridges.
	0	Moderately even texture with some fine pores.
	1	Smooth even texture; refined pores.
Hyperpigmentation	−1	Severe hyperpigmentation; dense clustered melanin microspots; thin semi-transparent blue contours around spots.
	0	Moderate hyperpigmentation; noticeable microspots with mild clustering; moderate thin blue contours.
	1	Minimal hyperpigmentation; few small, widely separated microspots; few thin blue contours.

Table A.3: Feature-specific prompt descriptors used to guide Stable Diffusion v1.5 `img2img` augmentation across classes. The *Class* column uses numeric alignment for consistent placement of −1, 0, and 1.

A.4 Computational Resources

All experiments were conducted in Google Colab Pro+ environments using Jupyter notebooks. The computational resources varied depending on the stage of the pipeline: CPU-only execution was used for lightweight preprocessing, while GPU acceleration (NVIDIA Tesla T4 with 16GB VRAM) was employed for model training and generative augmentation tasks.

In particular:

- Data preprocessing: executed on CPU, sufficient for handling metadata parsing, dataset structuring, and augmentation pipeline preparation.
- Stable Diffusion: implemented using a T4 GPU to generate synthetic skin images, leveraging the GPU's parallelism for efficient latent diffusion sampling.
- Google Derm Foundation Model: restricted to CPU inference, as the released implementation does not currently support GPU acceleration.
- PanDerm and DINOv2 (frozen embeddings): executed on a T4 GPU for batch embedding extraction, ensuring reasonable runtime efficiency.
- PanDerm and DINOv2 (fine-tuning): trained on a T4 GPU. Training times varied approximately from 2 to 6 hours.

This configuration proved sufficient for both training and inference phases given the small dataset, enabling experimentation with multiple foundation models while keeping runtimes within practical limits for iterative development.