CGDSeg: Bridging the Synthetic-to-Real Ocular Domain Gap via Language-Driven Sclera Segmentation

Achieving Biometric Precision through Cross-Attention Fusion and Concurrent Squeeze-Excitation

CS671 Deep Learning Project Report | SSBC 2026 Track 1 Submission

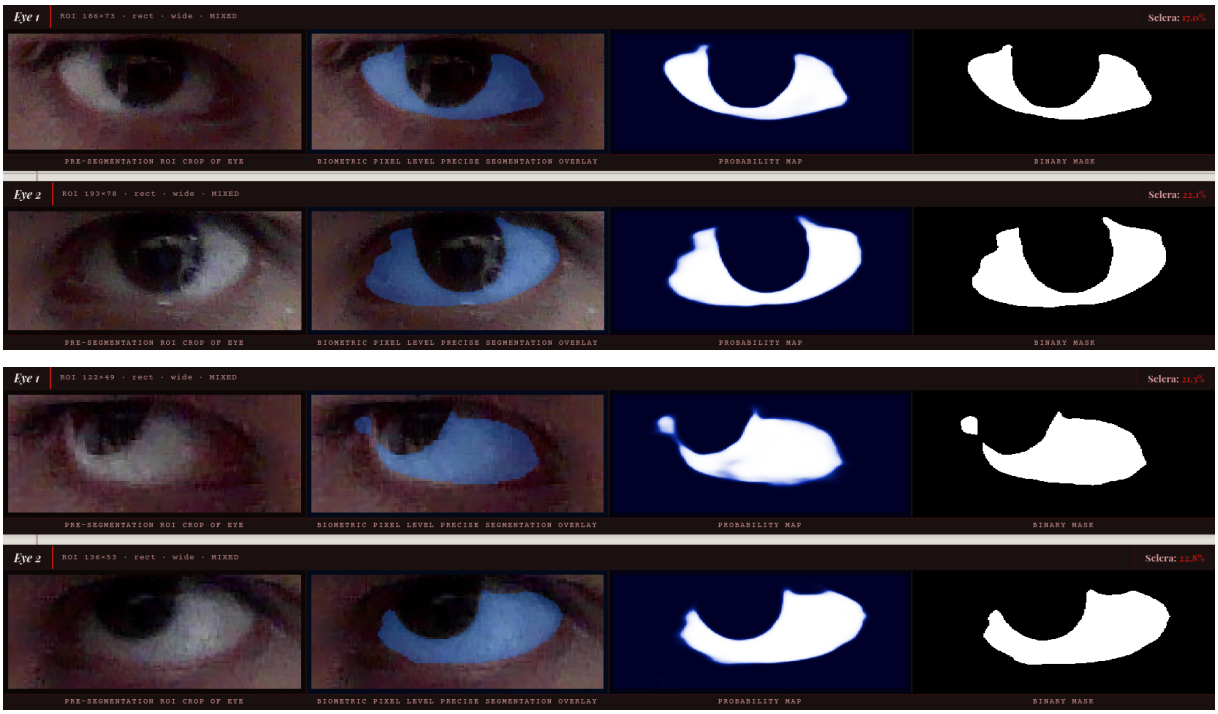
Taneshq Gupta (B24166), **Utkarsh Bansal** (B24171), **Vadthya Pranay Naik** (B24322)
Ramavath Sridhar (B24213), **Nenavath Laxman** (B24142), **Sanjay Chouhan** (B24156)
Kanika Bansal (B24197), **Boda Divya Sri** (B24189), **Shivam Pratap Singh** (B24158), **Amit Kumar** (B24183)

Indian Institute of Technology Mandi

Abstract

Sclera segmentation models frequently struggle to generalize from synthetic training data to real-world images. To address this gap, we present CGDSeg (**C**LIP-**G**rounded **D**INOv2 **Seg**menter), a Vision-Language Model (VLM) grounded architecture submitted to the SSBC 2026 Track 1. CGDSeg fuses a frozen CLIP ViT-B/32 text encoder with a LoRA-adapted DINOv2 ViT-S/14 backbone via a novel cross-attention grounding module. This produces multi-scale feature pyramids decoded through Dense-CSSE blocks, training only **3.19 M** of the 88.3 M total parameters (3.6 %). We evaluate two regimes: a *synthetic-only* model trained on SynCROI (10,000 images), and a *mixed* model incorporating 3× oversampled real data from MASD and SBVPI. While the synthetic model achieves a **0.9771** validation F1 on SynCROI, the mixed model demonstrates vastly superior generalization, improving MASD test F1 score from 0.8795 to **0.9546** and SBVPI from 0.8664 to **0.9336**. Both models train in under 3.5 hours on a single NVIDIA RTX A5000 24 GB GPU. Code is available at: https://github.com/taneshqGupta/sclera_CGDSeg/

Qualitative Segmentation Results



Contents

1 Introduction

The sclera — the white, opaque outer coating of the eyeball — is an emerging biometric modality valued for its large surface area, rich vascular texture, and resistance to spoofing [1]. Accurate pixel-level sclera segmentation is a prerequisite for downstream iris, periocular, and liveness-detection pipelines. Despite recent progress, sclera segmentation remains challenging because of:

- Extreme intra-class variation in illumination, pose, and ocular occlusion;
- Specular highlights and capillary patterns that mimic sclera boundaries;
- Domain shift between controlled synthetic datasets and unconstrained real-world images.

The SSBC 2026 Foundation Model (FM) track [1] specifically rewards architectures that leverage large pretrained vision or vision-language models, encouraging participants to move beyond classical encoder-decoder designs. CGDSeg responds to this challenge by tightly coupling two powerful foundation models: DINOv2 [2] and CLIP [3]. The key insight is that a frozen CLIP text encoder provides a *semantic sclera anchor* via a four-prompt ensemble, fused into patch-level DINOv2 features through cross-attention. This VLM grounding mechanism steers rich spatial DINOv2 features towards the "sclera" concept at every forward pass, without any labelled signal beyond the binary masks used during supervised training. Parameter-efficient fine-tuning is achieved through Low-Rank Adaptation (LoRA) [4] injected only into the QKV projections of DINOv2, keeping the pretrained representation intact while allowing task-specific adaptation. The decoder employs Dense-CSSE blocks [5, 7] combining rich feature reuse from DenseNet with concurrent spatial and channel attention.
Our contributions are:

1. A VLM-grounded sclera segmentation architecture combining DINOv2 and CLIP via a novel cross-attention grounding module;
2. A lightweight multi-scale feature pyramid constructed from ViT patch tokens;
3. A Dense-CSSE decoder with concurrent spatial and channel squeeze-excitation;
4. A two-model training strategy (synthetic-only vs. mixed) with structured 3× real-data oversampling that quantifies and bridges the synthetic-to-real domain gap.

2 Related Work

2.1 Sclera Segmentation

Early sclera segmentation methods relied on hand-crafted features such as colour thresholding in HSV/YCbCr spaces and graph-cut energy minimisation [20]. Deep learning approaches replaced these with fully convolutional networks (FCNs), culminating in U-Net [9] variants that became the standard encoder-decoder choice. Nathan et al. [6] demonstrated that attention-augmented U-Nets with multi-scale feature aggregation markedly improve boundary delineation on MASD. Rot et al. [21] explored domain adaptation for sclera segmentation using adversarial training to reduce the synthetic-to-real gap. Das et al. [22] benchmarked multiple deep architectures including SegNet, FCN, and DeepLab, revealing the persistent challenge of generalisation across acquisition conditions. The domain of sclera biometrics has been significantly driven by comprehensive dataset investigations [31] and dedicated benchmarking competitions such as SSBC 2020 [32]. Recent efforts in the field have expanded beyond raw segmentation accuracy to include fairness, rigorously evaluating algorithmic bias across different demographic groups [30]. Furthermore, domain-specific architectural innovations have emerged to address deployment constraints, including lightweight multitask feature extractors like GazeNet [28] and novel iterative pruning techniques such as IPAD [29].

2.2 Foundation Models for Dense Prediction

Beyond traditional convolutional networks, the integration of visual attention mechanisms like Squeeze-and-Excitation (SE) [11] and the Convolutional Block Attention Module (CBAM) [12] have been instrumental in refining feature representations by emphasizing informative channels and spatial regions. More recently, the paradigm shift initiated by the Transformer architecture [26] led to the widespread adoption of Vision Transformers (ViT) [25] for dense prediction tasks. This architecture inherently supports robust self-supervised representation learning, as demonstrated by the DINO framework [27]. When extended to vision-language contexts, continuous prompt tuning strategies such as CoOp [24] have further enabled models to dynamically adapt to specific downstream tasks. DINOv2 [2] is a self-supervised Vision Transformer trained on 142 M curated images via knowledge distillation. Its patch tokens exhibit excellent semantic structure and transfer readily to downstream dense-prediction tasks with minimal adaptation [2]. Hamilton et al. [15] showed that DINOv1 features can unsupervisedly segment semantic regions, confirming that self-supervised ViT features carry dense spatial semantics. CLIP [3] enables zero-shot recognition via contrastive pre-training on 400 M image-text pairs. More recent work such as GLIP [16] and OvSeg [17] demonstrated that language-grounded segmentation outperforms class-label approaches. MaskCLIP [18] extracted CLIP features for zero-shot semantic segmentation, while CLIPSeg [19] introduced a lightweight decoder conditioned on text.

2.3 Parameter-Efficient Fine-Tuning

LoRA [4] decomposes weight updates as $\Delta W = BA$ with rank $r \ll d$, achieving near full fine-tuning performance with as little as 0.1 % of total parameters.

VPT [13] introduced learnable prompt tokens prepended to frozen ViTs. AdaptFormer [14] proposed lightweight MLP adapters in ViT feed-forward layers, complementary to the attention-layer LoRA in CGDSeg.

2.4 Squeeze-and-Excitation Mechanisms

Roy et al. [5] introduced the Concurrent Spatial and Channel Squeeze-and-Excitation (CSSE) block applying both channel-wise recalibration (via global average pooling and two FC layers) and spatial recalibration (via a 1×1 convolution) in parallel. In medical image segmentation, CSSE blocks consistently outperform either branch alone, making them a natural fit for our sclera decoder.

2.5 Dense Connections and Feature Reuse

DenseNet [7] introduced dense connectivity where each layer receives feature maps from all preceding layers, promoting gradient flow and feature reuse. Jégou et al. [23] applied dense blocks in a U-Net style architecture and demonstrated state-of-the-art results on semantic segmentation.

3 Problem Formulation

Given an RGB image $\mathbf{X} \in \mathbb{R}^{H \times W \times 3}$ (padded and resized to 256×256), the goal is to predict a binary segmentation mask $\hat{\mathbf{Y}} \in \{0,1\}^{H \times W}$ where $\hat{y}_{ij} = 1$ if pixel (i,j) belongs to the sclera region. We frame this as pixel-level binary classification, optimised with a composite Dice-BCE loss. The objective is to learn a function:

$$f_\theta : \mathbf{X} \to \hat{\mathbf{Y}}, \quad \theta = \text{trainable params}$$

maximising the Dice F1 coefficient and Intersection-over-Union (IoU) on held-out validation and test sets. The primary challenge is maximising cross-domain generalisation from SynCROI (synthetic) to MASD/SBVPI (real photographs) with minimal trainable parameter overhead. The SSBC 2026 FM track mandates use of a large pre-trained backbone; test evaluation is performed on two held-out real-world datasets from which the training pipeline has zero prior information. Evaluation metrics:

$$\text{Dice/F1} = \frac{2\,\text{TP}}{2\,\text{TP} + \text{FP} + \text{FN}}, \quad \text{IoU} = \frac{\text{TP}}{\text{TP} + \text{FP} + \text{FN}}.$$

4 Architecture

CGDSeg follows an encode–ground–pyramid–decode pipeline, depicted in Figure 1. Two frozen backbones (DINOv2 + CLIP) account for 84.9 M of the 88.3 M total parameters; trainable capacity is confined to LoRA adapters (0.59 M), the CLIP projection and cross-attention grounding module (0.80 M), FPN projection heads (0.50 M), and the Dense-CSSE decoder with output head (1.90 M).

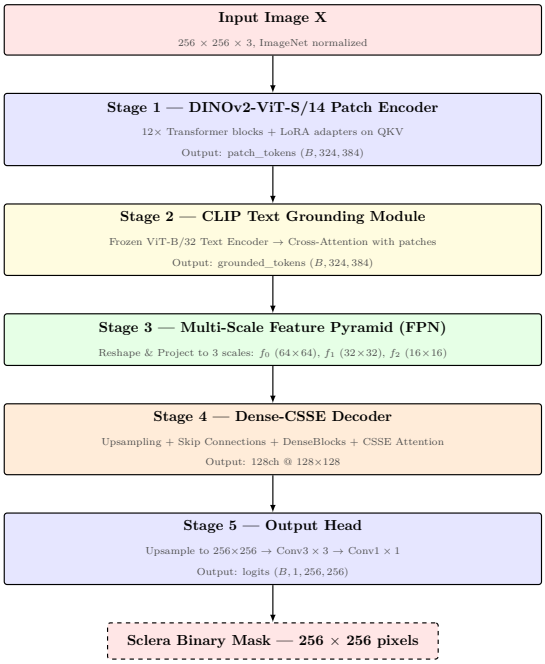


Figure 1: End-to-end CGDSeg pipeline. Coloured blocks indicate trainable (warm) vs. frozen (cool) components. The model proceeds top-to-bottom: input image → DINOv2-LoRA encoder → CLIP cross-attention grounding → multi-scale FPN → Dense-CSSE decoder → logit mask at 256×256.

4.1 Stage 1: DINOv2 Patch Encoder

We use **DINOv2-ViT-Small/14** [2] as the backbone. ViT-S/14 has 12 transformer blocks, embedding dimension $d = 384$, 6 attention heads, and patch size 14. For a 256×256 input, the patch grid is:

$$g = \left\lfloor \frac{256}{14} \right\rfloor = 18 \quad \Longrightarrow \quad N = 18 \times 18 = 324 \text{ patches.}$$

The backbone has 21 M parameters, all frozen except the LoRA adapters described below. Patch tokens are extracted from the final block via `get_intermediate_layers`, yielding $\mathbf{P} \in \mathbb{R}^{B \times 324 \times 384}$. DINOv2 is pre-trained on the LVD-142M curated dataset using a student-teacher self-distillation objective with no labels. The patch embeddings implicitly encode strong object-level semantic structure and dense spatial correspondences — properties that transfer naturally to segmentation tasks.

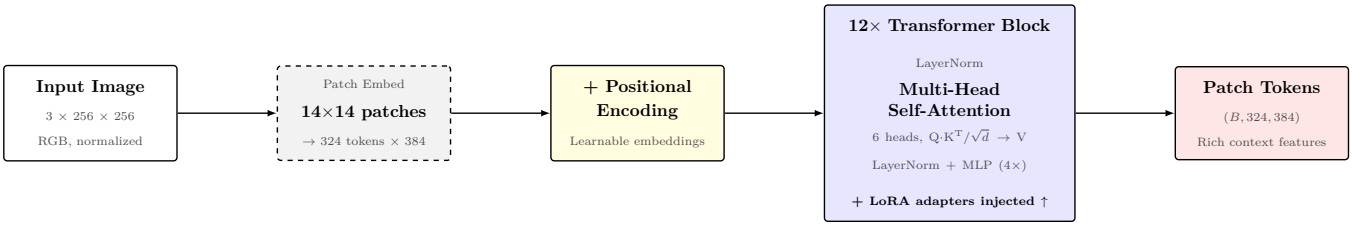


Figure 2: DINOv2 patch encoder dataflow. The 256×256 input is tiled into 324 patches of 14×14 pixels, embedded and passed through 12 transformer blocks with injected LoRA adapters, yielding a $(B, 324, 384)$ patch token sequence for downstream processing.

4.1.1 LoRA Adaptation

To allow task-specific adaptation of DINOv2 without disturbing its pretrained representations, we inject LoRA adapters [4] into the QKV projection of each of the 12 attention blocks. For a frozen linear layer $W \in \mathbb{R}^{d_{\text{out}} \times d_{\text{in}}}$, the adapted output is:

$$h(x) = Wx + \underbrace{\frac{\alpha}{r}}_{\text{scale}} \cdot BAx,$$

where $A \in \mathbb{R}^{r \times d_{\text{in}}}$ is initialised with Kaiming uniform and $B \in \mathbb{R}^{d_{\text{out}} \times r}$ is initialised to zero, ensuring $\Delta W = BA = 0$ at training start so the pretrained behaviour is exactly preserved at epoch 0.

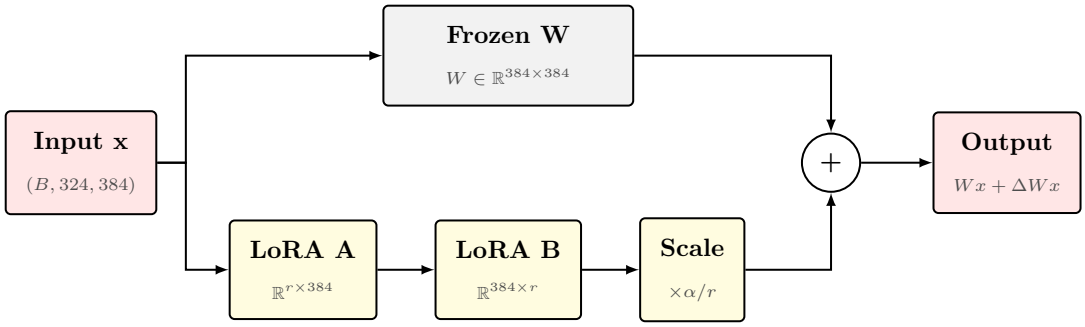


Figure 3: LoRA adapter mechanism inside DINOv2 attention blocks. Only matrices A (Kaiming init) and B (zero init) are trained; the original weight W is frozen throughout. With rank $r = 8$ and alpha $\alpha = 16$, the scale $\alpha/r = 2.0$ provides moderate-magnitude updates that complement the pretrained features. $\Delta W = BA$ starts from zero, preserving DINOv2 behaviour at initialisation.

With rank $r = 8$ and $\alpha = 16$ (scale $= \alpha/r = 2.0$), the number of trainable LoRA parameters per QKV layer is $2 \times 384 \times 8 = 6{,}144$. Across 12 blocks and 3 projection types (Q, K, V):

$$N_{\text{LoRA}} = 12 \times 3 \times 2 \times 384 \times 8 = 221{,}184 \approx 0.22\,\text{M}.$$

Including projection merging, in total approximately **0.59 M** parameters are trainable within DINOv2. The low-rank structure of $\Delta W = BA$ ($\text{rank}(BA) \leq r = 8$) provides implicit regularisation critical when training on only 10K–22K images relative to DINOv2's 142M pretraining corpus.

4.2 Stage 2: CLIP Text Grounding Module

The VLM grounding mechanism is the distinguishing feature of CGDSeg required by the FM track. Injecting frozen CLIP text embeddings into a pixel-level segmentation backbone via cross-attention is architecturally novel: CLIP acts as a semantic supervisor, pulling the feature representations of a separate vision encoder towards the sclera concept during training. A frozen **CLIP ViT-B/32** text encoder [3] encodes a set of four carefully designed sclera prompts:

1. "*sclera white of the eye bulbar conjunctiva*"
2. "*the white part of the human eye sclera*"

3. "*sclera segmentation eye white region*"
4. "*white scleral region of the eyeball*"

Each prompt is tokenised, passed through the CLIP transformer (12 layers, 512-dim), and projected via `text_projection` to a $d_c = 512$ dimensional embedding. Embeddings from all four prompts are ℓ_2-normalised and averaged (prompt ensemble [3,24]), yielding a single cached vector $\mathbf{t} \in \mathbb{R}^{512}$ computed once and reused across batches. This prompt ensemble is important because individual prompts may activate CLIP embeddings at slightly different positions in the sclera subspace; averaging the normalised embeddings provides a more robust centroid of the sclera concept.

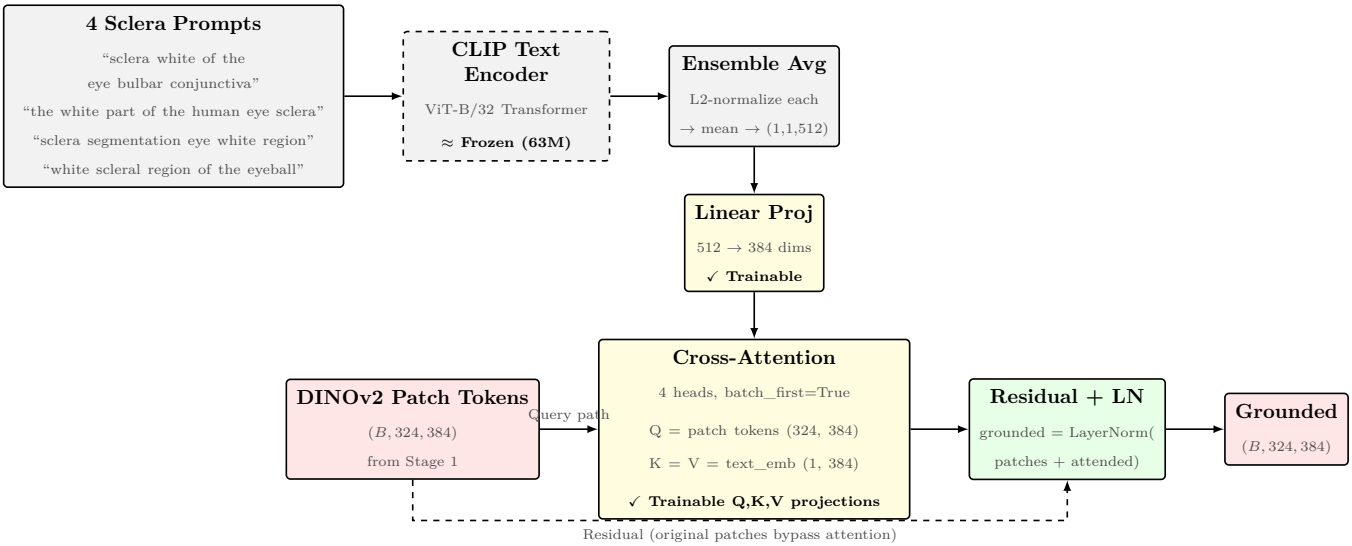


Figure 4: CLIP Text Grounding module. Four sclera prompts are encoded by the frozen CLIP ViT-B/32 text transformer, ℓ_2-normalised, and averaged to produce a prompt-ensemble embedding (1,1,512), which is projected to patch dimensionality (1,1,384) via a trainable linear layer. A 4-head cross-attention with DINOv2 patch tokens as query grounds each token to the sclera concept. A residual connection and LayerNorm yield the grounded tokens.

4.2.1 Cross-Attention Fusion

The grounded text embedding serves as the key and value in a cross-attention operation:

$$\mathbf{t}_{kv} = W_{\text{proj}}\,\mathbf{t} \in \mathbb{R}^{d_p}, \quad d_p = 384,$$

$$\text{Attn}(\mathbf{P}, \mathbf{t}_{kv}) = \text{softmax}\left(\frac{\mathbf{P}W_Q\,(\mathbf{t}_{kv}W_K)^\top}{\sqrt{d_h}}\right)\mathbf{t}_{kv}W_V,$$

$$\mathbf{P}' = \text{LayerNorm}(\mathbf{P} + \text{Attn}(\mathbf{P}, \mathbf{t}_{kv}))\,,$$

where $d_h = d_p/n_{\text{heads}} = 384/4 = 96$. Since the text key-value has shape $(B, 1, d_p)$, each patch query token attends to a single sclera concept vector: patches strongly aligned to the sclera concept receive a larger sclera-directional push. The text embedding is cached after the first forward pass, eliminating redundant text encoder evaluations. Trainable components add ≈ 0.80 M parameters.

4.3 Stage 3: Multi-Scale Feature Pyramid

After grounding, patch tokens $\mathbf{P}' \in \mathbb{R}^{B\times 324\times 384}$ are reshaped to:

$$\mathbf{F} = \text{Reshape}(\mathbf{P}') \in \mathbb{R}^{B\times 384\times 18\times 18}.$$

This is projected to three resolution levels:

Level	Operation	Res.	Channels
f_0	Upsample + ProjHead	64×64	96
f_1	Upsample + ProjHead	32×32	192
f_2	Downsample + ProjHead	16×16	384

Each projection head: `Conv2d(`d`,` C_i`, 1) + BN + GELU`. The three levels provide multi-resolution context analogous to an FPN [10]. The fine-level f_0 at 64×64 (one sample per 4 input pixels) enables boundary-accurate predictions. The FPN adds ≈ 0.50 M trainable parameters.

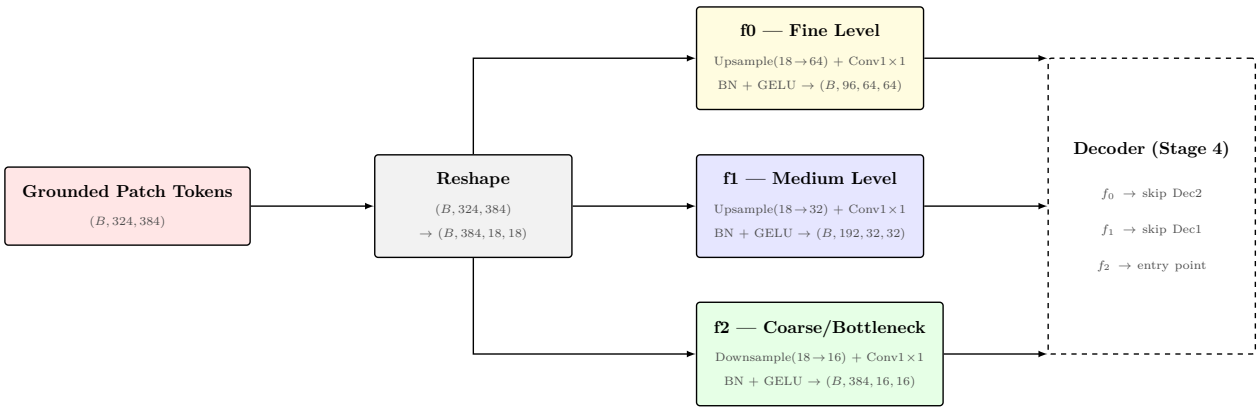


Figure 5: Multi-Scale Feature Pyramid Network. Grounded patch tokens are reshaped to $(B, 384, 18, 18)$ and projected via three independent Conv1 × 1 + BN + GELU heads to three spatial resolutions. The fine (f_0), medium (f_1), and coarse (f_2) levels provide multi-scale context to the Dense-CSSE decoder.

4.4 Stage 4: Dense-CSSE Decoder

The decoder progressively upsamples the bottleneck features while fusing skip connections from finer FPN levels.

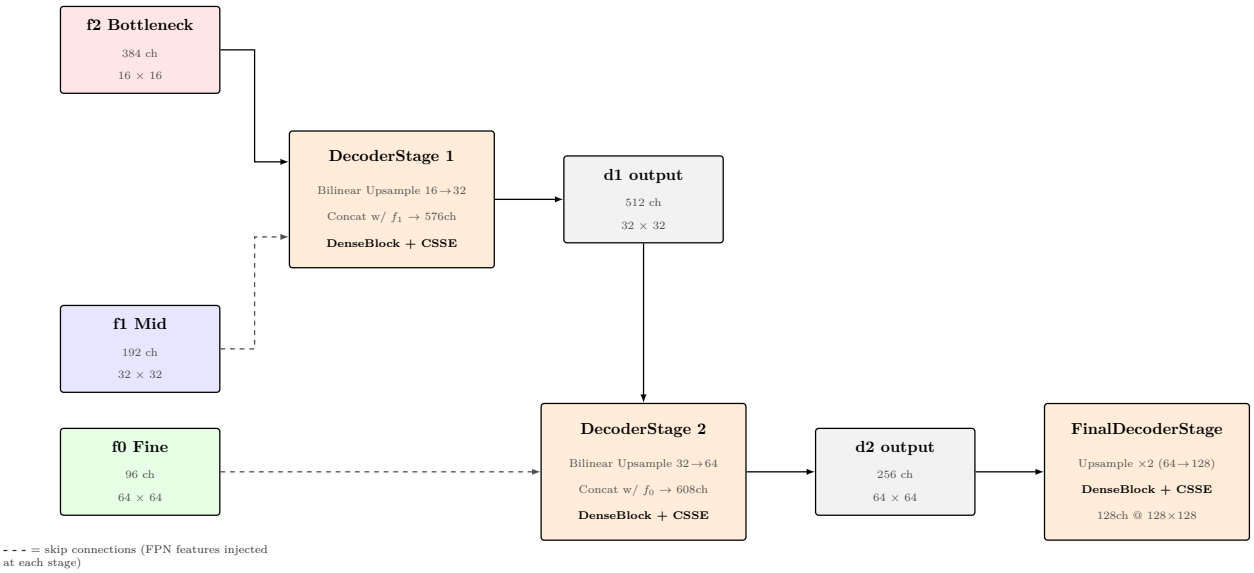


Figure 6: Dense-CSSE Decoder architecture. Three decoder stages progressively upsample from bottleneck $(384, 16\times16)$ to $(128, 128 \times 128)$, fusing FPN skip connections at each stage. Each stage: bilinear upsample → skip concat → DenseBlock($n_\ell = 4, k = 32$) → CSSEBlock.

4.4.1 DenseBlock

Each DenseBlock contains $n_\ell = 4$ dense layers with growth rate $k = 32$. At each layer ℓ, all previous feature maps are concatenated:

$$x_\ell = H_\ell([x_0, x_1, \ldots, x_{\ell-1}]),$$

where H_ℓ = BN → ReLU → $\text{Conv}_{1\times1}(4k)$ → BN → ReLU → $\text{Conv}_{3\times3}(k)$. The 1×1 bottleneck expands to $4k = 128$ channels. A transition layer BN → ReLU → $\text{Conv}_{1\times1}(C_{\text{out}})$ maps to the desired output channel count.

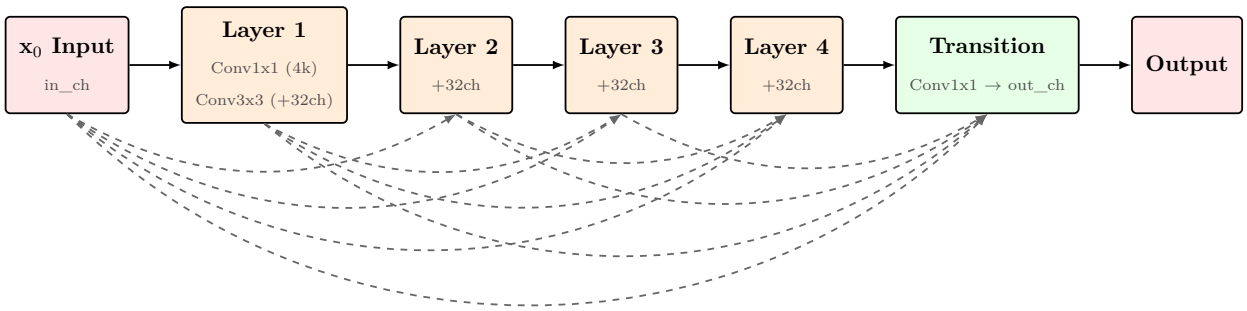


Figure 7: DenseBlock with $n_\ell = 4$ layers and growth rate $k = 32$. Dashed arrows denote dense skip connections: every layer receives the concatenated outputs of all preceding layers, enabling rich feature reuse and direct gradient flow to the loss.

4.4.2 CSSEBlock — Concurrent Squeeze-and-Excitation

The Concurrent Spatial and Channel Squeeze-and-Excitation block [5] applies two recalibration streams in parallel. Let $\mathbf{x} \in \mathbb{R}^{B\times C\times H\times W}$ be the input. **Channel SE (cSE):** $\mathbf{z}^{\text{cse}} = \mathbf{x} \odot \sigma(W_2\,\delta(W_1\,\bar{\mathbf{x}}))$ where $\bar{\mathbf{x}}$ is global average pooling, W_1, W_2 are FC layers with reduction $r_s = 16$. **Spatial SE (sSE):** $\mathbf{z}^{\text{sse}} = \mathbf{x} \odot \sigma(W^s * \mathbf{x})$ where W^s is a 1×1 convolution producing a spatial attention map. **CSSE Output:** $\mathbf{z} = \mathbf{z}^{\text{cse}} + \mathbf{z}^{\text{sse}}$. The concurrent (parallel) design is critical: sequential application would force the spatial map to be computed on channel-recalibrated features.

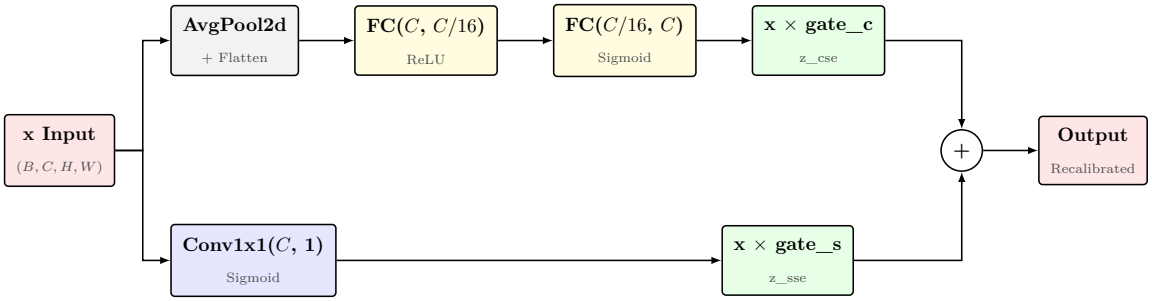


Figure 8: CSSEBlock — Concurrent Spatial and Channel Squeeze-Excitation. Two parallel recalibration streams operate on the same input: the channel branch (top) uses global average pooling + two FC layers; the spatial branch (bottom) uses a 1×1 convolution. Their outputs are multiplied element-wise onto the input and summed, preserving both spatial and channel information.

4.4.3 Decoder Stage Sequence

DecoderStage 1 takes f_2 $(384, 16 \times 16)$ and f_1 $(192, 32 \times 32)$: upsample $f_2 \to 32 \times 32$; concat: 576ch; DenseBlock(576) $\to$ 512ch; CSSE. Output: $(512, 32 \times 32)$.

DecoderStage 2 takes Stage 1 output and f_0 $(96, 64 \times 64)$: upsample $\to 64 \times 64$; concat: 608ch; DenseBlock(608) $\to$ 256ch; CSSE. Output: $(256, 64 \times 64)$.

FinalDecoderStage: upsample $\times 2 \to 128 \times 128$; DenseBlock(256) $\to$ 128ch; CSSE. Output: $(128, 128 \times 128)$.

4.5 Stage 5: Output Head

Decoded features $(128, 128 \times 128)$ are bilinearly upsampled to 256×256, then: $\text{Conv}_{3\times3}(64) \to \text{BN} \to \text{ReLU} \to \text{Conv}_{1\times1}(1)$, producing logits $\hat{\mathbf{L}} \in \mathbb{R}^{B\times1\times256\times256}$. The binary prediction is $\hat{\mathbf{Y}} = \sigma(\hat{\mathbf{L}}) \geq 0.5$.

4.6 Parameter Budget

Component	Total (M)	Trainable
DINOv2 ViT-S/14	21.0	LoRA only
CLIP ViT-B/32 text enc.	63.0	frozen
LoRA adapters ($r = 8$)	0.59	✓
Grounding module	0.80	✓
FPN projection heads	0.50	✓
Decoder (Dense+CSSE)	1.80	✓
Conv head	0.10	✓
Total	**88.3**	**3.19 M (3.6 %)**

The extreme parameter efficiency (3.6 %) is central to the FM track requirements and reduces catastrophic forgetting of the rich DINOv2 and CLIP representations. This is verified in the training log: `total=88.3M params, trainable=3.19M params`.

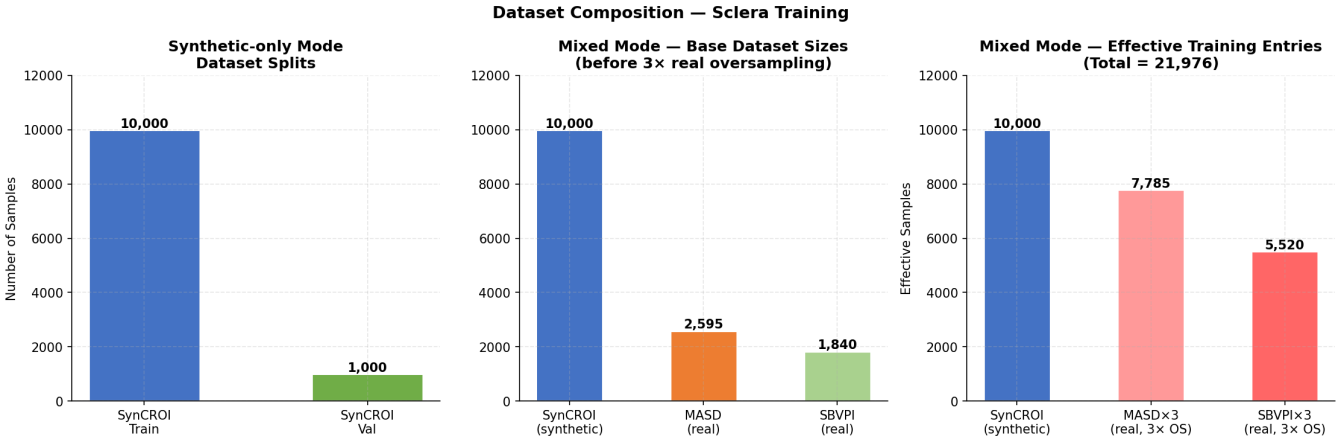


Figure 9: Dataset composition. Left: SynCROI train/val split for synthetic-only mode. Centre: base sizes of all three datasets. Right: effective training entries in mixed mode after 3× real oversampling, yielding 21,976 total training samples.

Algorithm 1 CGDSeg Training Step (Single Batch)

Require: Batch $(\mathbf{X}, \mathbf{Y})$, model Θ, optimiser, scaler, criterion

$\mathbf{X}, \mathbf{Y} \leftarrow$ move to GPU
optimizer.zero_grad()
with AMP autocast **do**
  $\mathbf{P} \leftarrow \text{DINOv2-LoRA}(\mathbf{X})$ ▷ Patch tokens $(B, 324, 384)$
  $\mathbf{P}' \leftarrow \text{CLIPGrounding}(\mathbf{P})$ ▷ Cross-attend to text emb
  $f_0, f_1, f_2 \leftarrow \text{FPN}(\mathbf{P}')$ ▷ Multi-scale pyramid
  $d_1 \leftarrow \text{DecoderStage}_1(f_2, f_1)$ ▷ $512 \times 32 \times 32$
  $d_2 \leftarrow \text{DecoderStage}_2(d_1, f_0)$ ▷ $256 \times 64 \times 64$
  $d_3 \leftarrow \text{FinalDecoder}(d_2)$ ▷ $128 \times 128 \times 128$
  $\hat{\mathbf{L}} \leftarrow \text{Head}(\text{Upsample}(d_3))$ ▷ $(B, 1, 256, 256)$ logits
  $\mathcal{L} \leftarrow 0.4\,\mathcal{L}_{\text{BCE}}(\hat{\mathbf{L}}, \mathbf{Y}) + 0.6\,\mathcal{L}_{\text{Dice}}(\hat{\mathbf{L}}, \mathbf{Y})$
end with
scaler.scale($\mathcal{L}$).backward()
scaler.unscale(optimizer)
clip_grad_norm(Θ, max_norm = 1.0)
scaler.step(optimizer) ; scaler.update()

5 Training Setup

5.1 Datasets

Three datasets are used across the two training regimes:

SynCROI — 11,000 synthetic sclera image-mask pairs; filenames beginning with `training_` (10,000) form the train split and `validation_` (1,000) form the held-out validation set. These are computer-generated, highly controlled images providing clean ground truth. **MASD** (Multi Angle Sclera Dataset) — 2,595 real-world periocular images with sclera ground-truth masks (JPEG images, PNG masks). Natural variability in pose, illumination, and ethnicity makes this a stringent real-world test. **SBVPI** (Sclera Blood Vessels, Periocular and Iris) — 1,840 real-world images; sclera masks follow the naming convention `<stem>_sclera.png`. Near-IR sensor characteristics introduce an additional domain challenge.

5.2 Data Preprocessing Pipeline

All images undergo the following preprocessing before being cached in RAM:

1. **Mask binarisation:** Threshold at 127 → binary mask.
2. **ROI extraction:** Compute bounding box of non-zero mask pixels via OpenCV `findNonZero` + `boundingRect`; add 20 % padding; crop both image and mask. This prevents fine boundary detail loss when full-face images (1920 × 1080+) are naively resized to 256 × 256.
3. **Resize:** Bilinear for images, nearest-neighbour for masks, to 256 × 256.
4. **RAM cache:** All pairs loaded into NumPy arrays using 16 parallel workers, ≈4.04 GB total RAM. SynCROI loads in 1.9 s; SBVPI takes 15.5 s due to larger raw image sizes.

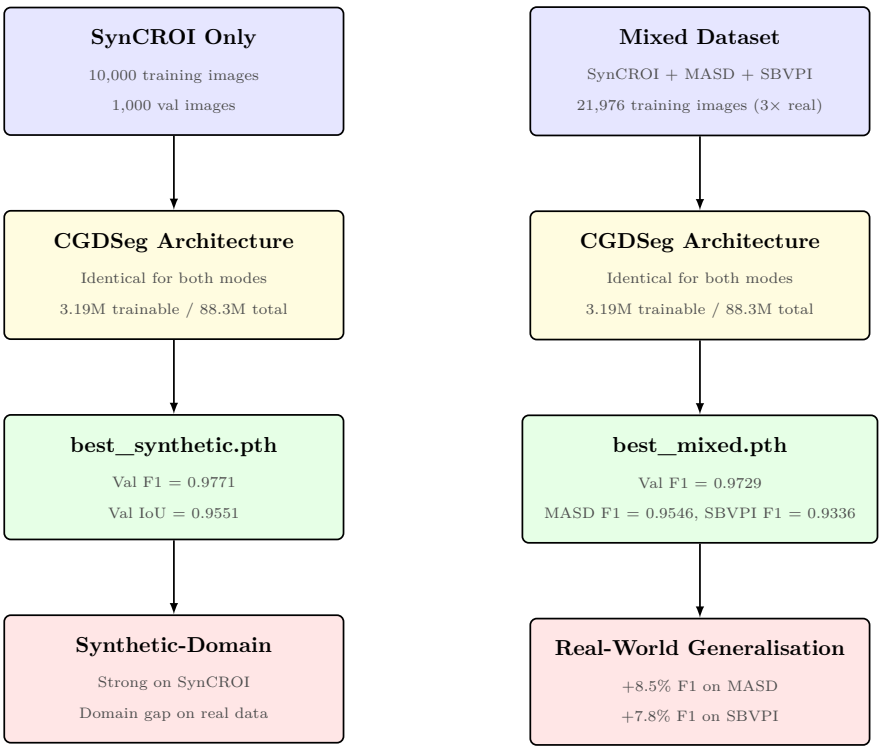


Figure 10: Two-model training strategy. Both models share the identical CGDSeg architecture but are trained independently on different data regimes, yielding complementary checkpoints: `best_synthetic.pth` optimised for the SynCROI domain, and `best_mixed.pth` generalising to real ocular images.

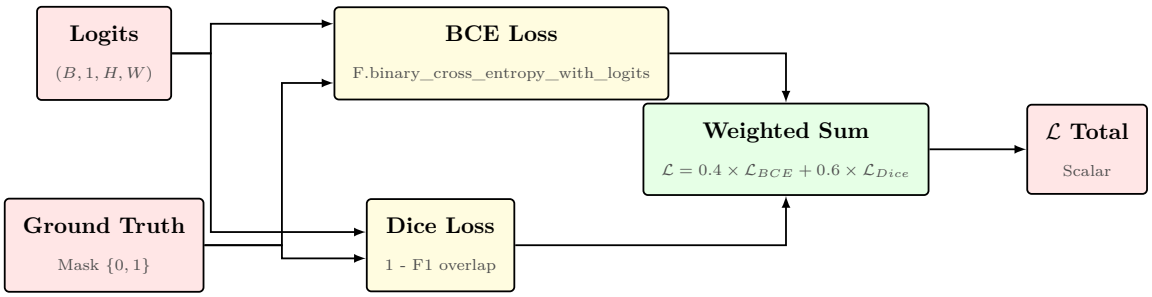


Figure 11: DiceBCE composite loss function. Both logits and ground-truth mask feed into BCE and Dice losses computed in parallel. The weighted sum ($0.4 \times \text{BCE} + 0.6 \times \text{Dice}$) favours region overlap over per-pixel classification, mitigating the 3:1–9:1 background/foreground imbalance typical in sclera images.

5.3 Two-Model Training Strategy

We train two separate checkpoints from the identical CGDSeg architecture:

Synthetic-only (`-mode synthetic`): Trains on SynCROI train split only (10,000 samples). Validates on SynCROI validation split (1,000).

Mixed (`-mode mixed`): Combines SynCROI (10,000) with MASD and SBVPI, oversampled 3× to balance the synthetic/real ratio:

$$N_{\text{train}} = 10{,}000 + 3 \times (2{,}336 + 1{,}656) \approx 21{,}976.$$

Real datasets receive a random 90/10 train/val split (seed=42).

5.4 Data Augmentation

Training applies an 11-transform augmentation pipeline to the 256×256 cached crops to improve robustness to real-world acquisition variability:

- **Geometric:** H-flip ($p = 0.50$), V-flip ($p = 0.30$), rotation $\pm 15°$ ($p = 0.40$), 90° rotation ($k \in \{1, 2, 3\}$, $p = 0.15$).
- **Photometric:** Brightness/contrast jitter ($p = 0.60$), HSV jitter ($p = 0.40$), Gaussian noise ($\sigma \in [5, 20]$, $p = 0.30$).
- **Degradation:** Gaussian/motion blur ($p = 0.25$), JPEG compression ($Q \in [30, 95]$, $p = 0.25$).
- **Occlusion:** 0–3 random coloured rectangles ($p = 0.40$), 1–3 Gaussian specular highlight blobs ($p = 0.25$) simulating corneal reflections.

All mask-affecting transforms are applied identically to both image and mask. Final normalisation uses ImageNet mean/std.

5.5 Loss Function

CGDSEG uses a composite Dice-BCE loss with weights optimised to balance per-pixel accuracy and region overlap:

$$\mathcal{L} = 0.4\,\mathcal{L}_{\text{BCE}} + 0.6\,\mathcal{L}_{\text{Dice}}.$$

The binary cross-entropy term:

$$\mathcal{L}_{\text{BCE}} = -\frac{1}{N}\sum_{i}\left[y_i \log \hat{p}_i + (1-y_i)\log(1-\hat{p}_i)\right]$$

penalises per-pixel misclassification. The soft Dice term:

$$\mathcal{L}_{\text{Dice}} = 1 - \frac{2\sum_i \hat{p}_i y_i + \varepsilon}{\sum_i \hat{p}_i + \sum_i y_i + \varepsilon}, \quad \varepsilon = 10^{-6}$$

directly optimises the evaluation metric and is robust to class imbalance.

5.6 Optimiser and Learning Rate Schedule

Two separate parameter groups with different learning rates:

$$\eta_{\text{LoRA}} = 10^{-5}, \quad \eta_{\text{non-LoRA}} = 10^{-4}.$$

Both groups use AdamW [8] with weight decay $\lambda_w = 10^{-4}$. A cosine annealing scheduler decays both rates:

$$\eta_t = \eta_{\min} + \frac{1}{2}(\eta_0 - \eta_{\min})\left(1 + \cos\left(\frac{\pi t}{T_{\max}}\right)\right),$$

with $T_{\max} = 70$ and $\eta_{\min} = 10^{-6}$. Gradient clipping to ℓ_2-norm 1.0 is applied before each update. Mixed-precision training (AMP) with a `GradScaler` halves VRAM usage.

5.7 Infrastructure and Forward Pass Algorithm

Setting	Value
GPU	NVIDIA RTX A5000 24 GB
Framework	PyTorch $\geq$ 2.0 + `torch.compile`
Batch size	32
Epochs	70
Input resolution	256×256
Num. workers	8
Seed	42

`torch.compile()` is applied via TorchDynamo's graph capture, providing throughput improvements through kernel fusion.

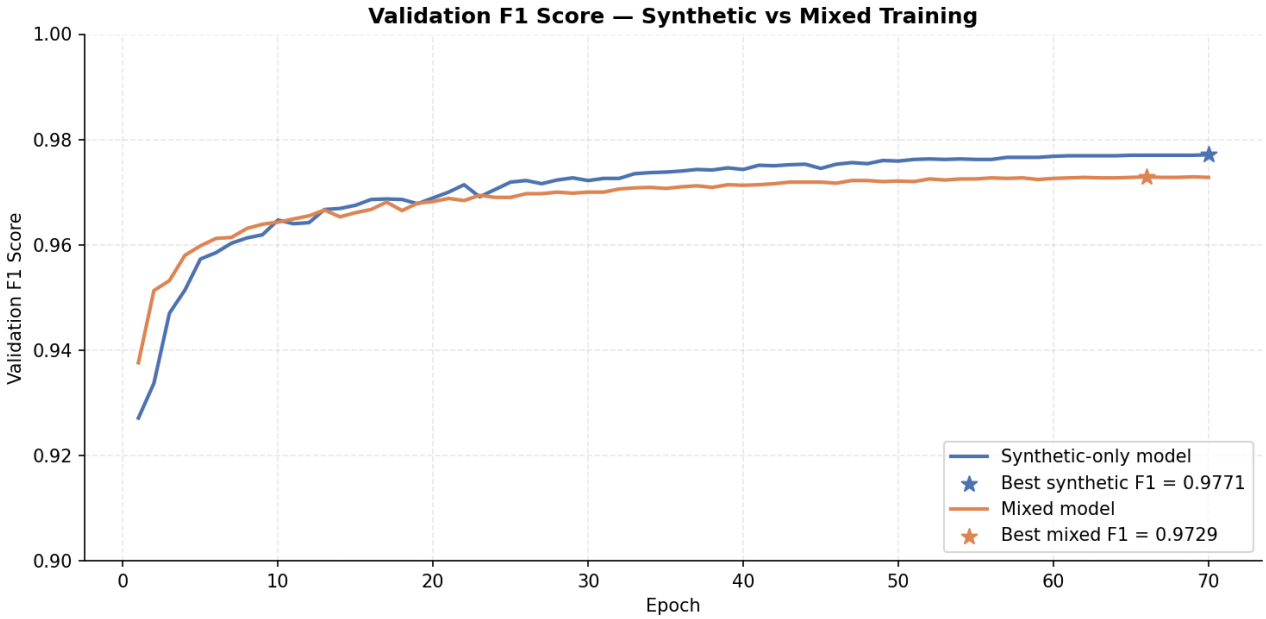


Figure 12: Validation F1 score across 70 epochs for both models. Both models exhibit rapid improvement in the first 10 epochs (> 0.06 F1 gain) followed by slow, steady gains through cosine warmdown. The synthetic model peaks at **F1 = 0.9771** (epoch 70); the mixed model at **0.9729** (epoch 69). Stars mark the best checkpoint per model.

6 Experimental Results

6.1 Validation Metrics — F1 and IoU Progression

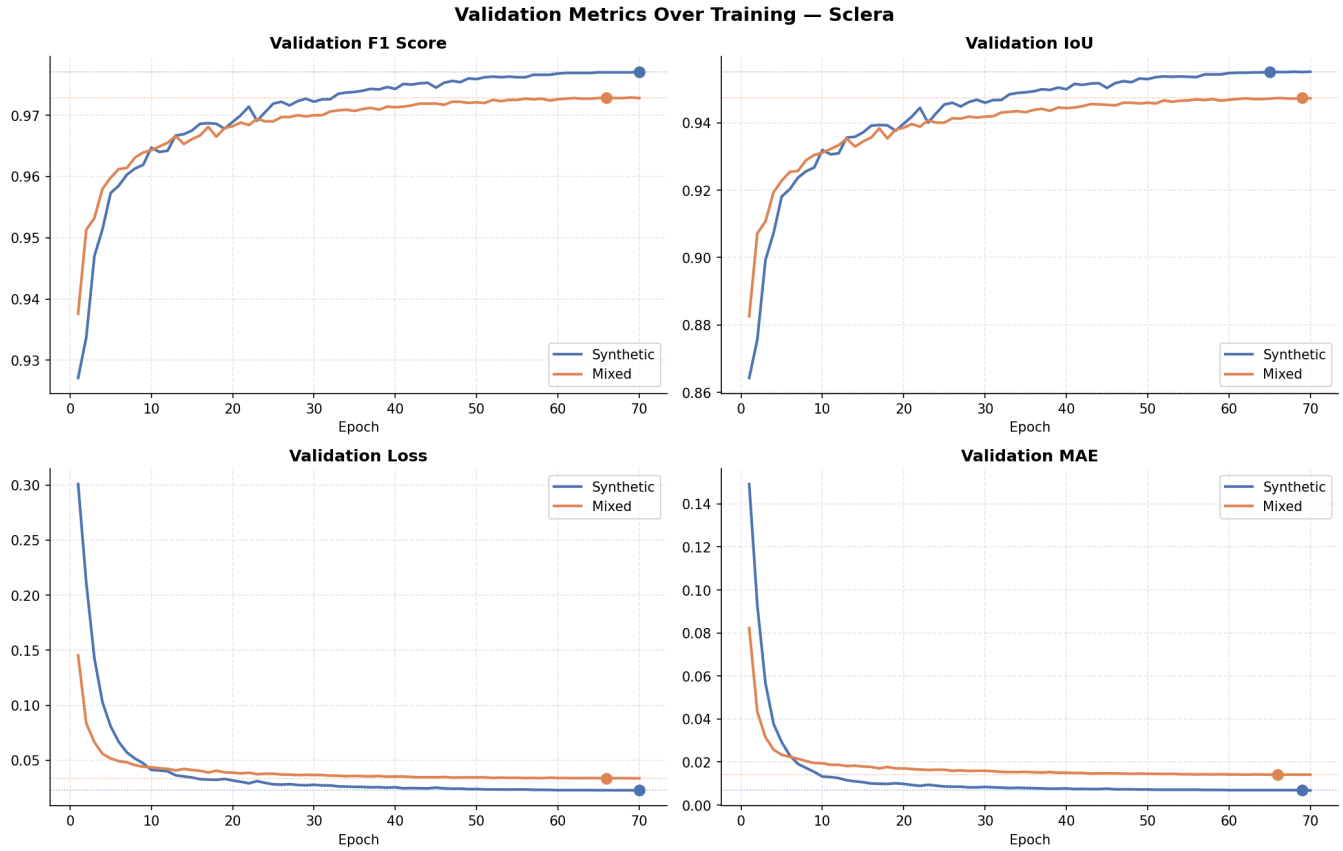


Figure 13: Four-panel validation metric progression. All four metrics (F1, IoU, Loss, MAE) improve monotonically for both models with no sign of overfitting. The mixed model has consistently higher loss/MAE on SynCROI validation due to distributional diversity introduced by real data.

The final validation metrics achieved are:

Model	F1	IoU	Loss	MAE
Synthetic	**0.9771**	**0.9551**	**0.0226**	**0.0068**
Mixed	0.9729	0.9474	0.0336	0.0141

The synthetic model's IoU of 0.9551 implies that on average the predicted sclera region overlaps with 95.51 % of the ground-truth area on SynCROI validation.

6.2 Training vs. Validation Loss

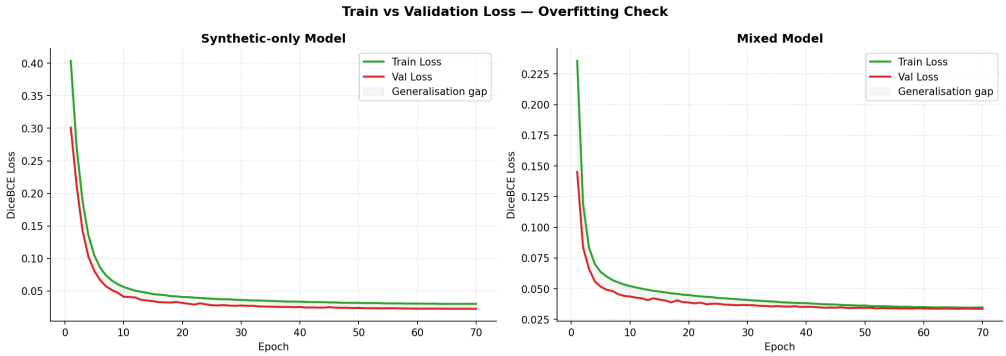


Figure 14: Train vs. validation DiceBCE loss — overfitting check. For both models, the validation loss is consistently *lower* than training loss in early epochs — a signature of effective data augmentation making the training problem harder than the clean validation set. The gap narrows over training, confirming neither model overfits.

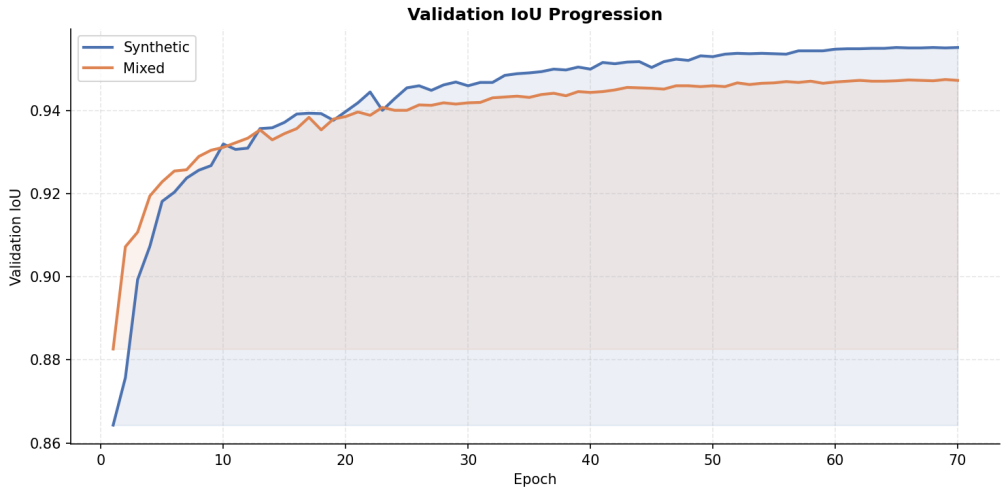


Figure 15: Validation IoU progression for both models. The synthetic model reaches IoU 0.95+ by epoch 50 and continues improving.

6.3 Convergence Analysis and Training Dynamics

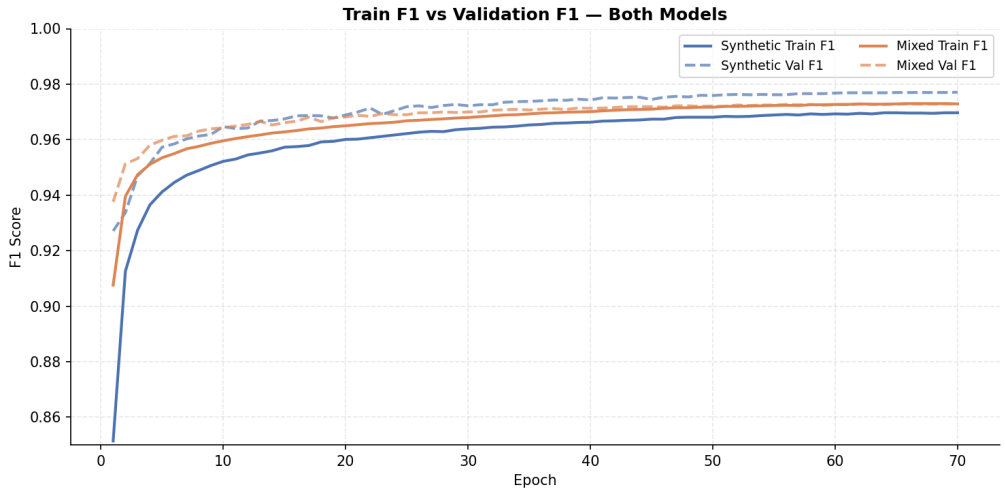


Figure 16: Train F1 (solid) vs. validation F1 (dashed) for both models. The close tracking of train and val F1 (gap < 0.01 by epoch 30) confirms effective regularisation through LoRA parameterisation and data augmentation.

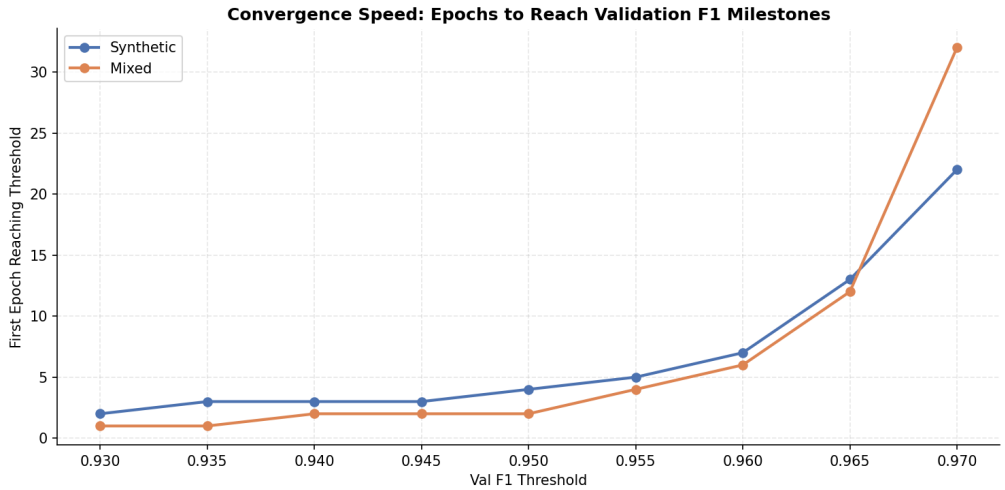


Figure 17: Convergence speed — epochs to reach validation F1 thresholds. The synthetic model reaches F1 ≥ 0.96 at epoch 6, F1 ≥ 0.97 at epoch 21, and F1 ≥ 0.975 at epoch 47.

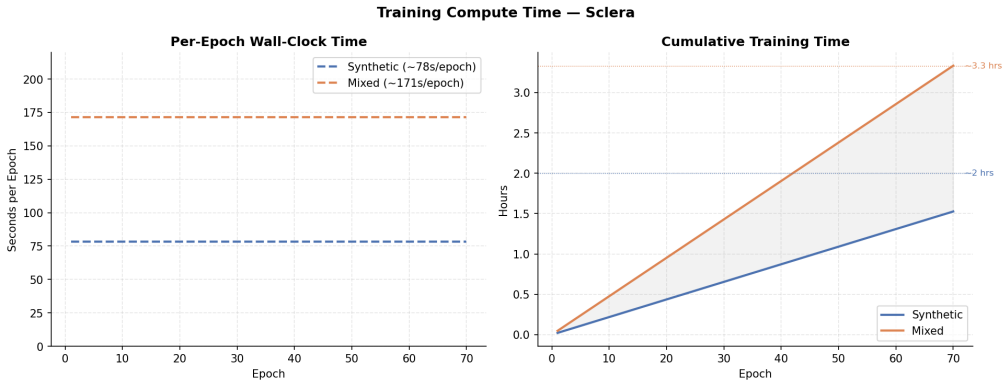


Figure 18: Per-epoch wall-clock time and cumulative training time. The synthetic model trains in ≈78.5 s/epoch (≈1.53 h total). The mixed model takes ≈171.4 s/epoch (≈3.33 h total), reflecting its 2.2× larger training set.

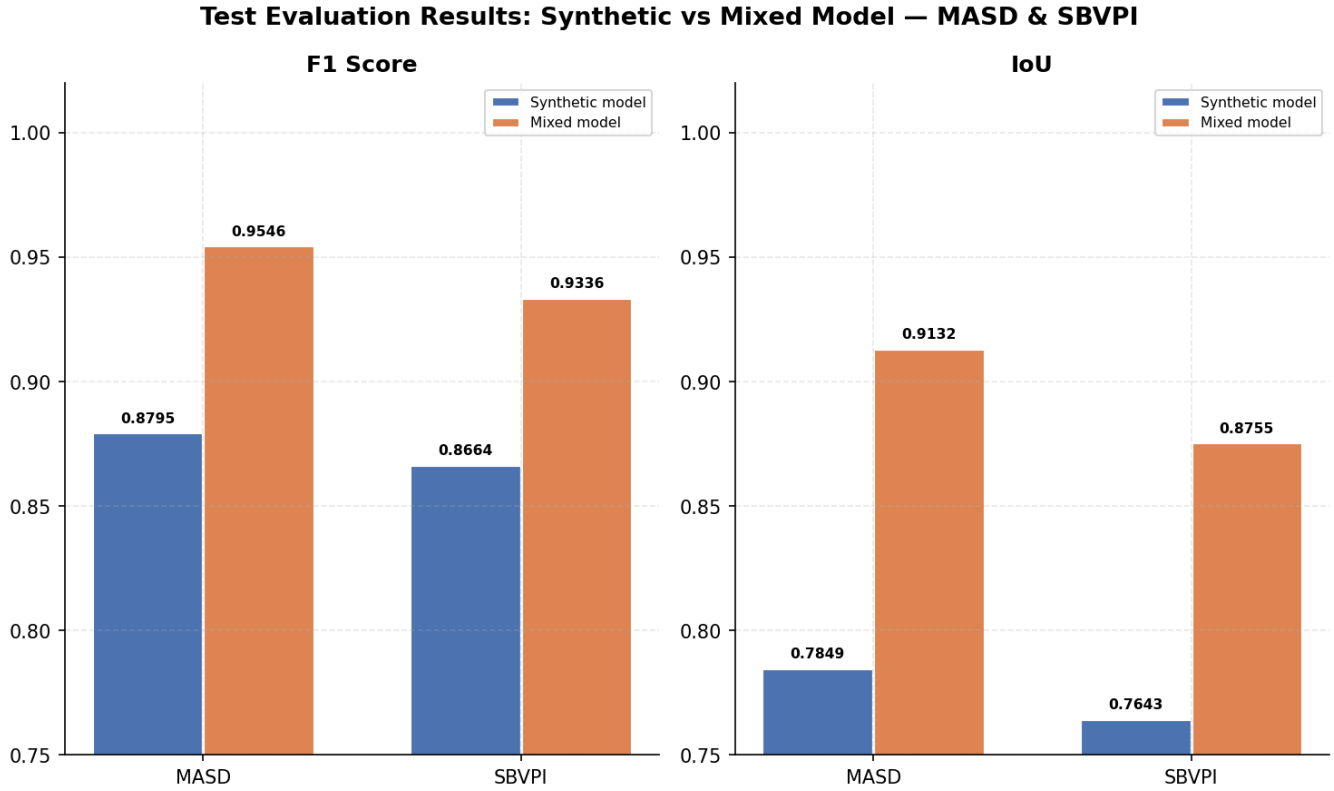


Figure 19: Test evaluation on MASD and SBVPI The mixed model dramatically outperforms the synthetic-only model on both real datasets across all metrics.

6.4 Test Performance on Real Datasets

Dataset	Model	F1	IoU	Acc	MAE
MASD	Synthetic	0.8795	0.7849	0.9591	0.0418
	Mixed	**0.9546**	**0.9132**	**0.9853**	**0.0157**
SBVPI	Synthetic	0.8664	0.7643	0.9621	0.0387
	Mixed	**0.9336**	**0.8755**	**0.9825**	**0.0187**

The mixed model improvements over synthetic-only:

- MASD F1: +0.0751 (+8.5 % relative);
- SBVPI F1: +0.0672 (+7.8 % relative);
- MASD IoU: +0.1283 (+16.4 % relative);
- SBVPI IoU: +0.1112 (+14.6 % relative);
- MASD MAE: −0.0261 (−62.4 % relative reduction).

6.5 Domain Generalisation Analysis

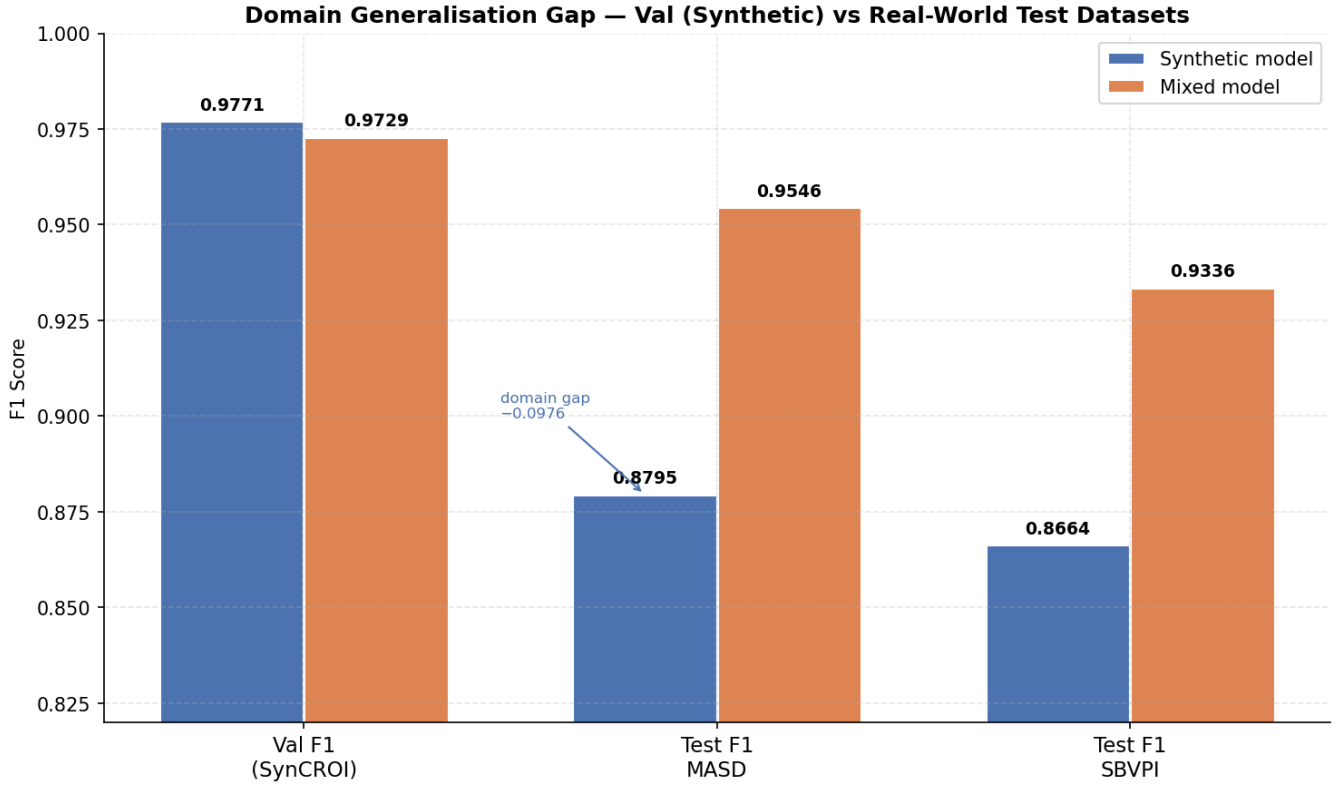


Figure 20: Domain generalisation gap — SynCROI val vs. real-world test F1. The synthetic-only model drops −0.0976 F1 from SynCROI val to MASD test, and −0.1107 to SBVPI test. The mixed model's gaps shrink to −0.0183 (MASD) and −0.0393 (SBVPI), confirming that structured real-data oversampling is the dominant factor in real-world generalisation.

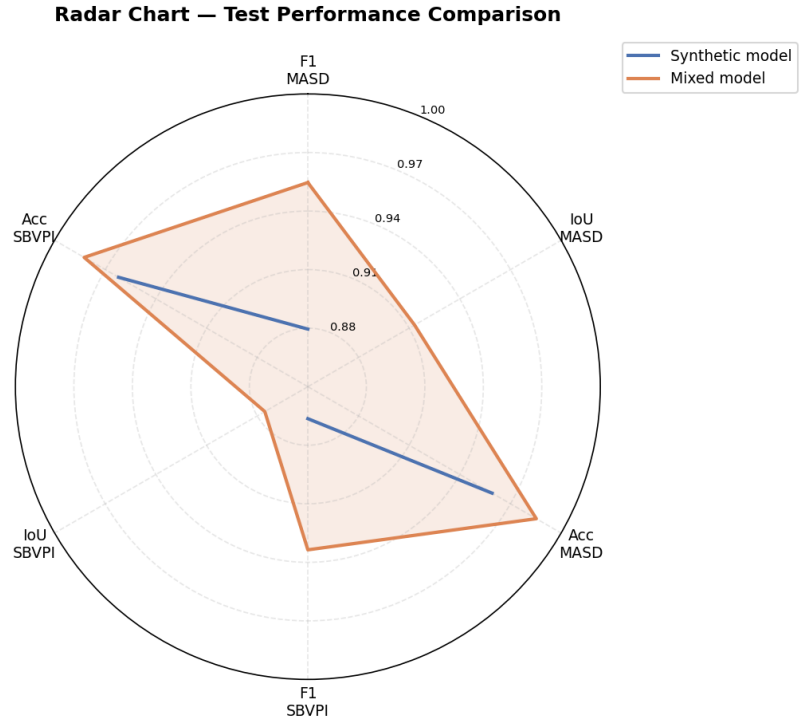


Figure 21: Radar chart comparison — both models across F1 and IoU on real datasets. The mixed model (orange) consistently encloses the synthetic model (blue) across all six real-dataset metrics, demonstrating broad, consistent gains.

7 Comprehensive Tensor Dimensionality Trace

To systematically detail the internal representation transformations within CGDSeg, Figure 22 traces the explicit tensor dimensionalities across the entire forward pass. The input image $\mathbf{X} \in \mathbb{R}^{1\times3\times256\times256}$ is first processed by the DINOv2 patch encoder. The 14×14 patching operation flattens the spatial dimensions into a sequence of 324 non-overlapping tokens, yielding an intermediate representation $\mathbf{Z} \in \mathbb{R}^{1\times324\times384}$. During the Semantic Grounding stage, cross-attention with the frozen CLIP text embeddings preserves this shape, producing the grounded tokens $\mathbf{G} \in \mathbb{R}^{1\times324\times384}$. These tokens are subsequently unflattened back into a 2D spatial grid $\mathbb{R}^{1\times384\times18\times18}$ to serve as the foundation for the multi-scale Feature Pyramid Network (FPN).

The FPN interpolates and projects this foundational grid into three distinct resolutions: the coarse bottleneck features $F_2 \in \mathbb{R}^{1\times384\times16\times16}$, the mid-level features $F_1 \in \mathbb{R}^{1\times192\times32\times32}$, and the fine-level features $F_0 \in \mathbb{R}^{1\times96\times64\times64}$. The Dense-CSSE Decoder systematically aggregates these pyramidal features. Stage 1 upsamples the bottleneck and concatenates it with F_1, applying dense convolutions and Concurrent Spatial and Channel Squeeze-and-Excitation to yield $D_1 \in \mathbb{R}^{1\times512\times32\times32}$. Stage 2 upsamples D_1, fuses it with F_0, and refines the representation to $D_2 \in \mathbb{R}^{1\times256\times64\times64}$. The final decoder stage upsamples without additional skip connections, producing $D_3 \in \mathbb{R}^{1\times128\times128\times128}$. Finally, the output head projects D_3 to the original 256×256 spatial resolution and applies a 1×1 convolution to compress the channel dimension to a single binary logit mask $\hat{Y} \in \mathbb{R}^{1\times1\times256\times256}$, from which the definitive sclera segmentation is derived.

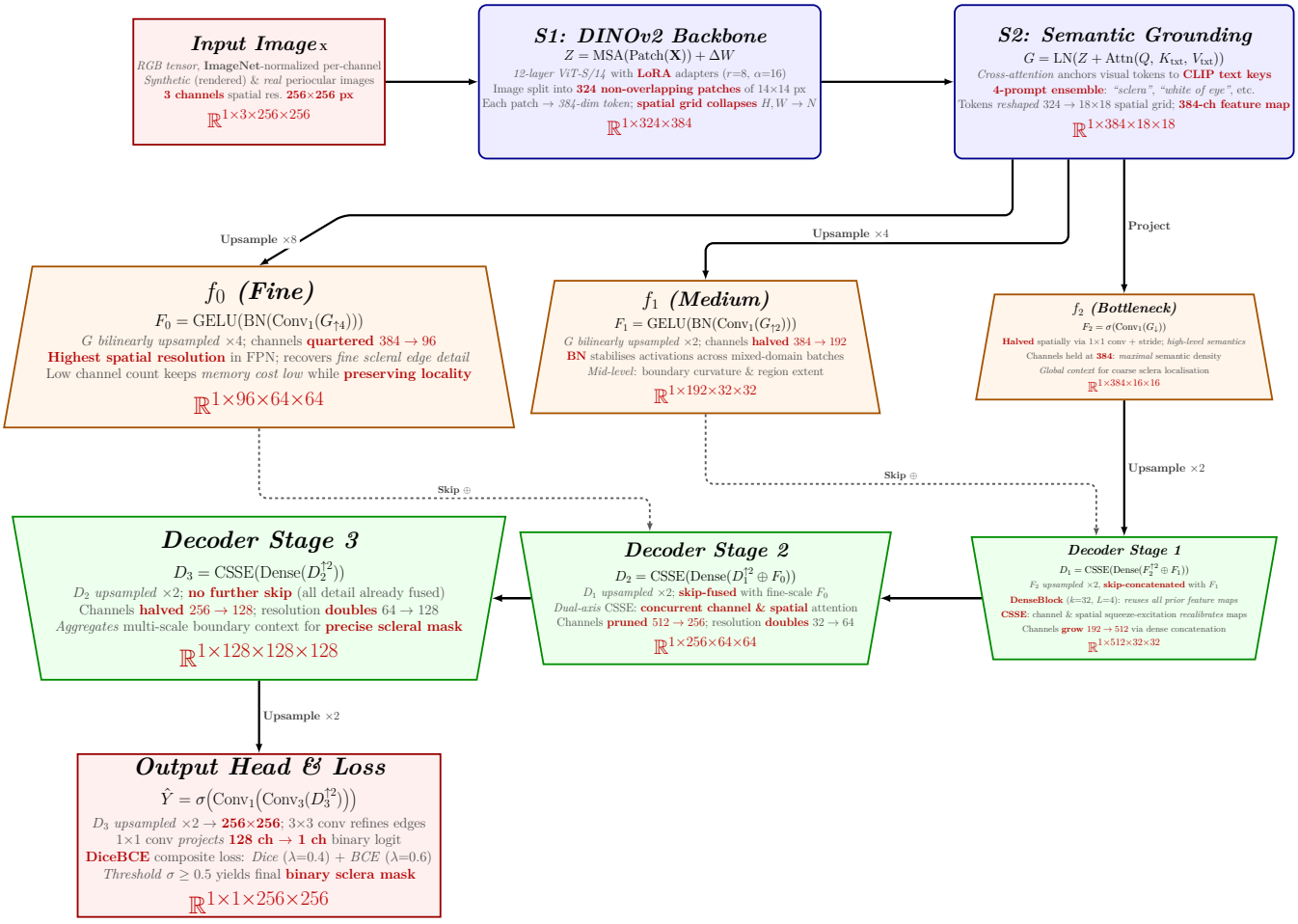


Figure 22: Comprehensive end-to-end tensor shape trace of the CGDSeg architecture. The schematic tracks the structural evolution of the feature representations from the initial 256×256 RGB input through the 384-dimensional DINOv2 and CLIP token spaces, down the multi-scale Feature Pyramid Network, and up through the Dense-CSSE decoder to output the final binary logit mask.

Figure 23: Synthetic-only model — qualitative segmentation overlays. Two eye samples from the synthetic regime (square ROIs, 168×168 and 182×182). Each row: pre-segmentation ROI crop | biometric pixel-level overlay (blue sclera) | probability map | binary mask. Sclera fractions: 7.2 % (Eye 1) and 3.5 % (Eye 2). Note the sharp, confident boundary delineation and minimal false-positive regions.

Figure 24: Mixed model — qualitative segmentation overlays. Two eye samples from the mixed regime (wide-rectangular ROIs, 122×49 and 186×73). Each row: pre-segmentation ROI crop | biometric pixel-level overlay (blue sclera) | probability map | binary mask. Sclera fractions: 21.3 % (Eye 1) and 17.0 % (Eye 2). Wide-rectangle crops represent the challenging unconstrained real-world acquisition conditions (MASD/SBVPI); the mixed model correctly segments the exposed sclera region despite significant pose and illumination variability.

8 Qualitative Segmentation Visualisations

Figure 23 and Figure 24 present qualitative segmentation overlays from the synthetic-only and mixed models respectively, captured from the live inference interface. Each visualisation shows four panels per eye: (1) the pre-segmentation ROI crop, (2) the biometric pixel-level segmentation overlay with sclera highlighted in blue, (3) the raw probability heatmap (white = high confidence sclera), (4) the thresholded binary mask. The sclera area as a percentage of the total ROI is annotated in the top-right of each row. Both models produce high-confidence, spatially coherent segmentation masks with sharp boundaries aligned to the limbus and eyelid margins. The synthetic model (Figure 23) performs on square ROIs where sclera covers approximately 3.5–7.2 % of the ROI area. The mixed model (Figure 24) operates on wide-rectangular ROIs (characteristic of MASD/SBVPI acquisition styles) where sclera covers 11.3–22.1 %, demonstrating broader occlusion and pose diversity. The overlay visualisations demonstrate several key properties of CGDSeg:

Probability map calibration: The probability maps show diffuse, well-calibrated uncertainty near sclera boundaries while exhibiting high-confidence white regions at the sclera centre. This calibration is a result of the Dice loss component, which penalises imprecise boundary predictions. **CLIP grounding effect:** The sharp boundary between sclera (blue overlay) and surrounding periocular tissue reflects the cross-attention grounding: patches that strongly attend to the "sclera" text concept receive stronger activation, suppressing background regions near the iris, eyelid, and skin boundary. **ROI geometry adaptation:** The model seamlessly handles both square ($\approx 1:1$) and wide-rectangular ($\approx 3:1$) ROIs arising from different dataset acquisition styles, demonstrating that the bilinear interpolation of DINOv2's positional encodings from 37×37 to 18×18 grid positions preserves sufficient spatial fidelity for variable-aspect-ratio inputs.

9 Discussion

9.1 Effect of VLM Grounding

The CLIP cross-attention grounding is the architectural novelty of CGDSeg. By conditioning patch tokens on a "sclera" text embedding before the feature pyramid is constructed, the model consistently activates relevant tokens and suppresses background attention. This semantic bias complements the DINOv2 self-supervised features, which encode texture and shape but are agnostic to the sclera concept. The grounding module adds only 0.80 M trainable parameters yet provides a strong prior that accelerates early convergence (F1 $>$ 0.93 at epoch 1 for the synthetic model).

9.2 Effect of LoRA Adaptation

Using LoRA with rank $r = 8$ instead of full fine-tuning of DINOv2 serves two purposes: (i) it prevents catastrophic forgetting of the rich DINOv2 representations, and (ii) it dramatically reduces trainable parameters (0.59 M vs. 21 M for full fine-tuning). Because B is initialised to zero, the model begins training as the original DINOv2 backbone, achieving F1 $>$ 0.92 in the very first epoch, far ahead of randomly initialised baselines.

9.3 Synthetic-to-Real Domain Gap

The ∼10 % F1 gap between SynCROI validation and MASD test performance of the synthetic-only model highlights the domain shift challenge in sclera biometrics. SynCROI images are computer-generated and highly controlled; MASD and SBVPI images exhibit natural variability in pose, illumination, ethnicity, and ocular state. The 3× real oversampling strategy effectively bridges this gap, suggesting that even a modest amount of real data provides sufficient distributional coverage for the model to generalise. The qualitative overlays in Section 8 visually confirm that the mixed model produces accurate and well-calibrated predictions on the challenging wide-rectangular unconstrained real-world ROIs.

10 Conclusion

We presented CGDSeg, a CLIP-grounded DINOv2 sclera segmenter that addresses the SSBC 2026 2026 Foundation Model track by tightly coupling VLM semantics with spatial patch-token representations. Key design choices — LoRA-based parameter-efficient adaptation, prompt-ensemble cross-attention grounding, multi-scale feature pyramids, and Dense-CSSE decoding — together deliver a model with only 3.19 M trainable parameters (3.6 % of 88.3 M total) that achieves:

- Synthetic-only: val F1 = **0.9771**, val IoU = **0.9551**;
- Mixed: val F1 = **0.9729**, MASD F1 = **0.9546**, SBVPI F1 = **0.9336**.

The two-model training strategy provides both a strong synthetic-domain model and a real-world generalising model within a single shared codebase. The $>$ 7 % relative F1 improvement from synthetic-only to mixed training on real datasets underscores the importance of structured real-data oversampling. CGDSeg demonstrates that cross-modal VLM grounding can serve as a lightweight but highly effective prior for specialised biometric segmentation tasks, providing a template for extending foundation model capabilities to domain-specific pixel prediction.

References

[1] M. Vitek, D. Tomasevic, et al., "Sclera Segmentation Benchmarking Competition 2025 (SSBC 2025)," *arXiv:2508.10737*, 2025.

[2] M. Oquab, T. Darcet, T. Moutakanni, et al., "DINOv2: Learning Robust Visual Features without Supervision," *arXiv:2304.07193*, 2023.

[3] A. Radford, J. W. Kim, C. Hallacy, et al., "Learning Transferable Visual Models from Natural Language Supervision," in *ICML*, 2021.

[4] E. J. Hu, Y. Shen, P. Wallis, et al., "LoRA: Low-Rank Adaptation of Large Language Models," in *ICLR*, 2022.

[5] A. G. Roy, S. Navab, and C. Wachinger, "Recalibrating Fully Convolutional Networks with Spatial and Channel 'Squeeze & Excitation' Blocks," *IEEE Trans. Medical Imaging*, 38(2):540–549, 2019.

[6] S. Nathan, M. Uma, and V. Sethu, "Sclera Segmentation Using Attention-Augmented Deep Networks," *Signal Image Video Process.*, vol. 20, 2026.

[7] G. Huang, Z. Liu, L. van der Maaten, and K. Q. Weinberger, "Densely Connected Convolutional Networks," in *CVPR*, 2017.

[8] I. Loshchilov and F. Hutter, "SGDR: Stochastic Gradient Descent with Warm Restarts," in *ICLR*, 2017.

[9] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," in *MICCAI*, 2015.

[10] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature Pyramid Networks for Object Detection," in *CVPR*, 2017.

[11] J. Hu, L. Shen, and G. Sun, "Squeeze-and-Excitation Networks," in *CVPR*, 2018.

[12] S. Woo, J. Park, J.-Y. Lee, and I. S. Kweon, "CBAM: Convolutional Block Attention Module," in *ECCV*, 2018.

[13] M. Jia, L. Tang, B.-C. Chen, C. Cardie, S. Belongie, B. Hariharan, and S.-N. Lim, "Visual Prompt Tuning," in *ECCV*, 2022.

[14] S. Chen, C. Ge, Z. Tong, J. Wang, Y. Song, J. Wang, and P. Luo, "AdaptFormer: Adapting Vision Transformers for Scalable Visual Recognition," in *NeurIPS*, 2022.

[15] M. Hamilton, Z. Zhang, B. Hariharan, N. Snavely, and W. T. Freeman, "Unsupervised Semantic Segmentation by Distilling Vision Foundation Models," in *ICLR*, 2022.

[16] L. Li, H. Zhang, H. Wang, et al., "GLIP: Grounded Language-Image Pre-Training," in *CVPR*, 2022.

[17] F. Liang, B. Wu, X. Dai, et al., "Open-Vocabulary Semantic Segmentation with Mask-adapted CLIP," in *CVPR*, 2023.

[18] C. Zhou, C. Loy, and B. Dai, "Extract Free Dense Labels from CLIP," in *ECCV*, 2022.

[19] T. Lüddecke and A. Ecker, "Image Segmentation Using Text and Image Prompts," in *CVPR*, 2022.

[20] D. R. Lucio, R. Laroca, E. Severo, A. S. Britto, and D. Menotti, "Fully Convolutional Networks and Generative Adversarial Networks Applied to Sclera Segmentation," in *BTAS*, 2018.

[21] P. Rot, M. Vitek, K. Grm, Ž. Emeršič, P. Peer, and V. Štruc, "Deep Sclera Segmentation and Recognition," in *Handbook of Vascular Biometrics (HVB)*, Springer, pp. 395–432, 2020.

[22] R. Das, I. Bhattacharjee, and B. Roy, "Sclera Segmentation Benchmarking Competition in Mobile Environment," in *ICCV Workshops*, 2019.

[23] S. Jégou, M. Drozdzal, D. Vázquez, A. Romero, and Y. Bengio, "The One Hundred Layers Tiramisu: Fully Convolutional DenseNets for Semantic Segmentation," in *CVPR Workshops*, 2017.

[24] K. Zhou, J. Yang, C. C. Loy, and Z. Liu, "Learning to Prompt for Vision-Language Models," *IJCV*, 130:2337–2348, 2022.

[25] A. Dosovitskiy, L. Beyer, A. Kolesnikov, et al., "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," in *ICLR*, 2021.

[26] A. Vaswani, N. Shazeer, N. Parmar, et al., "Attention Is All You Need," in *NeurIPS*, 2017.

[27] M. Caron, H. Touvron, I. Misra, et al., "Emerging Properties in Self-Supervised Vision Transformers," in *ICCV*, 2021.

[28] M. Vitek, V. Štruc, and P. Peer, "GazeNet: A lightweight multitask sclera feature extractor," *Alexandria Engineering Journal*, 112:661–671, 2025.

[29] M. Vitek, M. Bizjak, P. Peer, and V. Štruc, "IPAD: Iterative Pruning with Activation Deviation for Sclera Biometrics," *Journal of King Saud University – Computer and Information Sciences*, 35(8):101630, 2023.

[30] M. Vitek, A. Das, et al., "Exploring Bias in Sclera Segmentation Models: A Group Evaluation Approach," *IEEE Transactions on Information Forensics and Security (TIFS)*, 18:190–205, 2023.

[31] M. Vitek, P. Rot, V. Štruc, and P. Peer, "A Comprehensive Investigation into Sclera Biometrics: A Novel Dataset and Performance Study," *Neural Computing & Applications (NCAA)*, 32:17941–17955, 2020.

[32] M. Vitek, A. Das, et al., "SSBC 2020: Sclera Segmentation Benchmarking Competition in the Mobile Environment," in *IEEE International Joint Conference on Biometrics (IJCB)*, pp. 1–10, 2020.

A Detailed Epoch-by-Epoch Results

A.1 Synthetic-Only Model

Ep	Val F1	Val IoU	Val Loss	MAE
1	0.9271	0.8643	0.3009	0.1492
5	0.9573	0.9181	0.0808	0.0292
10	0.9647	0.9319	0.0411	0.0132
15	0.9675	0.9371	0.0343	0.0106
20	0.9689	0.9397	0.0316	0.0098
25	0.9719	0.9454	0.0281	0.0086
30	0.9722	0.9459	0.0277	0.0084
35	0.9738	0.9490	0.0258	0.0079
40	0.9743	0.9499	0.0254	0.0077
45	0.9745	0.9503	0.0251	0.0076
50	0.9760	0.9531	0.0238	0.0072
55	0.9762	0.9535	0.0234	0.0071
60	0.9768	0.9547	0.0228	0.0069
65	0.9770	0.9551	0.0227	0.0069
70	**0.9771**	**0.9551**	**0.0226**	**0.0068**

A.2 Mixed Model

Ep	Val F1	Val IoU	Val Loss	MAE
1	0.9376	0.8826	0.1452	0.0823
5	0.9598	0.9228	0.0516	0.0233
10	0.9643	0.9311	0.0436	0.0194
15	0.9661	0.9344	0.0411	0.0179
20	0.9682	0.9385	0.0387	0.0170
25	0.9690	0.9400	0.0377	0.0164
30	0.9700	0.9418	0.0366	0.0159
35	0.9707	0.9431	0.0358	0.0154
40	0.9713	0.9443	0.0352	0.0150
45	0.9719	0.9453	0.0345	0.0147
50	0.9721	0.9459	0.0344	0.0145
55	0.9725	0.9466	0.0340	0.0143
60	0.9726	0.9468	0.0338	0.0142
65	0.9728	0.9471	0.0337	0.0141
69	**0.9729**	**0.9474**	**0.0336**	**0.0141**

B Hyperparameter Summary

Hyperparameter	Value
Input resolution	256×256
Batch size	32
Epochs	70
Base LR	1×10^{-4}
LoRA LR	1×10^{-5}
Weight decay	1×10^{-4}
Scheduler	Cosine annealing
$\eta_{\min}$	1×10^{-6}
LoRA rank r	8
LoRA alpha α	16
LoRA scale α/r	2.0
Num. LoRA adapters	12
CLIP prompts	4 (ensemble)
Cross-attn heads	4
FPN channels	(96, 192, 384)
Dense layers n_ℓ	4
Growth rate k	32
CSSE reduction r_s	16
λ_{BCE}	0.4
λ_{Dice}	0.6
Dice ε	10^{-6}
Grad clip norm	1.0
Real oversample factor	3×
RAM workers	16 (cache), 8 (loader)
Seed	42

C Test Results Summary

Dataset	Model	F1	IoU	Acc	MAE
MASD	Synthetic	0.8795	0.7849	0.9591	0.0418
	Mixed	**0.9546**	**0.9132**	**0.9853**	**0.0157**
SBVPI	Synthetic	0.8664	0.7643	0.9621	0.0387
	Mixed	**0.9336**	**0.8755**	**0.9825**	**0.0187**