Post-training feature pruning for fundus images classification

Supplementary Material

Compactness and Separability Metrics

To evaluate how GFP changes the structure of the latent feature space, we compute three standard metrics of cluster compactness and separability on the final feature representations (Table 4).

Setup and notation. Let $z_i \in \mathbb{R}^L$ denote the flattened feature vector of sample i produced by the backbone, and let $y_i \in \{1, \dots, C\}$ be its class label. For each class c, define the index set

$$\mathcal{D}_c = \{\, i \mid y_i = c \,\}, \quad N_c = |\mathcal{D}_c|, \quad N = \sum_{c=1}^{C} N_c.$$

The class mean and global mean in feature space are

$$\mu_c = \frac{1}{N_c} \sum_{i \in \mathcal{D}_c} z_i, \qquad \mu = \frac{1}{N} \sum_{i=1}^{N} z_i,$$

and $\|\cdot\|$ denotes the Euclidean norm.

Intra-class variance (lower is better). Intra-class variance measures the average within-class scatter of features (i.e., how tightly samples cluster around their class mean) [4]. We define

$$\mathrm{Var}_{\mathrm{intra}} = \frac{1}{C} \sum_{c=1}^{C} \frac{1}{N_c} \sum_{i \in \mathcal{D}_c} \|z_i - \mu_c\|^2. \tag{13}$$

Lower values indicate that samples from the same class are more compactly clustered in the latent space.

Fisher Discriminant Ratio (higher is better). The Fisher Discriminant Ratio (FDR) [13] compares the between-class variance of the class means to the within-class variance. Using the intra-class variance above as

$$\sigma^2_{\mathrm{within}} = \mathrm{Var}_{\mathrm{intra}}, \tag{14}$$

we define the between-class variance as

$$\sigma^2_{\mathrm{between}} = \frac{1}{C} \sum_{c=1}^{C} \|\mu_c - \mu\|^2, \tag{15}$$

and the multi-class Fisher Discriminant Ratio as

$$\mathrm{FDR} = \frac{\sigma^2_{\mathrm{between}}}{\sigma^2_{\mathrm{within}}} = \frac{\frac{1}{C} \sum_{c=1}^{C} \|\mu_c - \mu\|^2}{\frac{1}{C} \sum_{c=1}^{C} \frac{1}{N_c} \sum_{i \in \mathcal{D}_c} \|z_i - \mu_c\|^2}. \tag{16}$$

Higher FDR values indicate that class means are far apart relative to their internal scatter, i.e., better class separability.

Silhouette score (higher is better). The Silhouette score [31] jointly captures cluster cohesion and separation. For each sample z_i, define

$$a(i) = \frac{1}{N_{y_i} - 1} \sum_{\substack{j \in \mathcal{D}_{y_i} \\ j \neq i}} \|z_i - z_j\|, \tag{17}$$

the average distance to other samples in the *same* class, and

$$b(i) = \min_{c \neq y_i} \frac{1}{N_c} \sum_{j \in \mathcal{D}_c} \|z_i - z_j\|, \tag{18}$$

the minimum average distance to samples in any *other* class. The silhouette value for sample i is

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i),\, b(i)\}}, \tag{19}$$

which lies in $[-1, 1]$, and the overall Silhouette score is

$$\mathrm{Silhouette} = \frac{1}{N} \sum_{i=1}^{N} s(i). \tag{20}$$

Values close to 1 indicate that samples are well matched to their own class and far from other classes; values near 0 indicate overlap between clusters; and negative values suggest potential misassignment. Thus, higher Silhouette scores correspond to more coherent and better separated class clusters in feature space.

Threshold-Dependent Metrics

While AUROC and AUPRC provide threshold-independent summaries of ranking quality, clinical deployment typically requires operating at a fixed decision threshold, e.g., for screening versus referral. To assess GFP under such operating points, we additionally report threshold-dependent metrics—Accuracy (ACC), Precision (Prec), Recall (Rec), Specificity (Spec), and F1 score—on all datasets. Results for DDR, Messidor-2, and PAPILA are shown in Table 6, and those for ODIR and RETINA in Table 7. For each model and dataset, a single threshold value of 0.5 is applied per class to convert probabilistic outputs into binary predictions, and the metrics are computed from the resulting confusion matrix.

Across the three DR and GS detection datasets (DDR, Messidor-2, PAPILA), GFP often improves the balance of sensitivity and specificity for ViT and CoAtNet. For example, ViT + GFP achieves the highest ACC and F1 on DDR (82.63% and 80.33%), and CoAtNet + GFP reaches

Table 6. Performance comparison on DDR, Messidor-2, and PAPILA. Metrics reported: Accuracy (ACC), Precision (Prec), Recall (Rec), Specificity (Spec), and F1 score.

Backbone / Method	DDR (DR det.)					Messidor-2 (DR det.)					PAPILA (GS det.)				
	ACC	Prec	Rec	Spec	F1	ACC	Prec	Rec	Spec	F1	ACC	Prec	Rec	Spec	F1
EfficientNetV2	81.51	92.17	68.87	94.15	78.83	**80.84**	74.58	81.48	80.39	**77.88**	77.03	64.00	66.67	82.00	65.31
EfficientNetV2 + MP	**81.59**	91.88	**69.29**	93.88	**79.00**	79.69	72.36	**82.41**	77.78	77.06	**79.73**	**68.00**	70.83	**84.00**	**69.39**
EfficientNetV2 + GFP	79.76	**95.74**	62.27	**97.23**	75.46	66.28	**100**	18.52	**100**	31.25	63.51	46.51	**83.33**	54.00	59.70
TRAM	76.72	**94.11**	57.00	**96.44**	71.00	70.50	59.63	88.89	57.52	71.38	79.73	90.91	41.67	98.00	57.14
LTMP	82.15	92.78	69.72	94.57	79.61	73.56	67.89	68.52	77.12	68.20	71.62	**100**	12.50	**100**	22.22
ViT	81.40	93.07	67.86	94.95	78.49	**80.08**	**78.57**	71.30	**86.27**	74.76	**82.43**	92.31	**50.00**	98.00	**64.86**
ViT + MP	81.14	92.58	67.70	94.57	78.82	**80.08**	**78.57**	71.30	**86.27**	74.76	**82.43**	92.31	**50.00**	98.00	**64.86**
ViT + GFP	**82.63**	92.57	**70.94**	94.31	**80.33**	**80.08**	76.92	**74.07**	84.31	**75.47**	78.38	83.33	41.67	96.00	55.56
CoAtNet	88.03	**99.45**	76.48	**99.57**	86.46	84.67	80.91	82.41	86.27	81.65	79.73	**90.91**	41.67	**98.00**	57.14
CoAtNet + MP	88.93	99.06	78.61	99.26	**87.66**	79.69	72.00	**83.33**	77.12	77.25	81.08	85.71	50.00	96.00	63.16
CoAtNet + GFP	**89.01**	99.13	**78.71**	99.31	87.75	**86.21**	**83.96**	82.41	**88.89**	**83.18**	**85.14**	88.24	**62.50**	96.00	**73.13**

Table 7. Performance comparison on ODIR and RETINA. Metrics reported: Accuracy (ACC), Precision (Prec), Recall (Rec), Specificity (Spec), and F1 score.

Backbone / Method	ODIR (multi-label)					RETINA (multi-class)				
	ACC	Prec	Rec	Spec	F1	ACC	Prec	Rec	Spec	F1
EfficientNetV2	**93.76**	63.64	**44.72**	98.90	**49.84**	76.92	52.14	53.05	83.58	52.00
EfficientNetV2 + MP	93.71	63.37	44.18	98.92	49.20	76.38	51.12	51.39	83.04	50.78
EfficientNetV2 + GFP	93.39	**63.89**	43.85	**99.07**	48.61	**78.57**	**52.89**	**54.72**	**84.57**	**53.33**
TRAM	92.67	64.98	33.70	98.63	38.00	86.26	72.27	62.18	87.69	63.78
LTMP	93.06	70.03	**49.22**	97.61	**54.71**	85.17	66.76	67.64	88.53	67.01
ViT	**93.57**	76.37	47.01	98.64	51.40	90.11	85.65	75.07	91.28	77.20
ViT + MP	**93.57**	**77.16**	46.40	**98.72**	51.24	88.46	75.90	72.50	90.72	72.79
ViT + GFP	93.56	60.72	48.37	98.33	51.93	89.56	78.18	77.43	92.01	77.22
CoAtNet	**94.46**	**80.46**	49.84	**98.27**	55.19	**91.76**	**84.78**	**82.98**	**93.76**	**82.11**
CoAtNet + MP	94.23	74.14	50.04	97.96	**55.24**	**91.76**	**84.78**	**82.98**	**93.76**	**82.11**
CoAtNet + GFP	94.27	76.13	**54.18**	97.35	55.19	91.21	83.98	81.32	93.22	81.04

the best ACC and F1 on Messidor-2 (86.21% and 83.18%) and PAPILA (85.14% and 73.13%). These gains indicate that GFP can improve diagnostic consistency at a fixed operating point while using substantially fewer feature dimensions.

However, GFP does not uniformly improve all metrics on all backbones. In several cases, pruning leads to *reduced sensitivity (Recall)* when aggressive feature removal occurs. For example, EfficientNetV2 + GFP shows strong precision and specificity (often 95–100%) on DDR and Messidor-2, but its Recall drops substantially (e.g., from 68.87% to 62.27% on DDR and from 81.48% to 18.52% on Messidor-2), illustrating a trade-off toward conservative predictions. Similar degradations also appear in ViT + GFP on PAPILA, where Recall decreases (50.00% to 41.67%), and in GFP on ODIR, where F1 is slightly reduced relative to MP or the original backbone.

For the multi-label (ODIR) and multi-class (RETINA) tasks, pruning leads to more mixed outcomes. GFP preserves or slightly improves ACC and Spec for several models (e.g., EfficientNetV2 on RETINA), but does not consistently outperform MP or the original models in F1 or Recall for ViT and CoAtNet. These results highlight that while GFP offers strong threshold-independent improvements (AUROC/AUPRC), threshold-dependent performance can show mild degradation when the pruned features contain threshold-critical information (e.g., disease-rare features that influence Recall).

Inference time comparison

Although GFP significantly reduces the dimensionality of the final feature vector, it is important to assess whether this reduction translates into measurable inference-time im-

Table 8. Inference-time comparison with and without GFP. For each dataset, we report the number of flattened features and the average per-image inference time (ms).

Backbone / Method	DDR		Messidor-2		PAPILA		ODIR		RETINA	
	#Feat.	Time (ms)	#Feat.	Time (ms)	#Feat.	Time (ms)	#Feat.	Time (ms)	#Feat.	Time (ms)
EfficientNetV2	2152	1.8907	2152	3.2119	2152	19.3387	2152	2.4757	2152	7.7208
EfficientNetV2 + GFP	2104	1.8782	1824	3.1240	1292	19.2259	2005	2.4384	2061	7.5806
ViT	768	2.5139	768	4.0434	768	19.9308	768	3.3733	768	8.4905
ViT + GFP	256	2.5086	416	4.0035	32	19.1072	736	3.3554	56	8.4835
CoAtNet	1024	2.7056	1024	4.3178	1024	20.0955	1024	3.5153	1024	8.7723
CoAtNet + GFP	103	2.6932	32	4.1708	30	19.3447	656	3.5005	410	8.6485

provements. We therefore compare the per-image latency of each backbone before and after pruning across five datasets, as reported in Table 8. For each dataset, we measure (i) the number of retained flattened features and (ii) the average inference time on a single NVIDIA GPU A6000 using identical preprocessing, batch size, and hardware settings. Across all backbones, GFP removes a substantial portion of the flattened features—up to 96% for CoAtNet on PAPILA (1024 $\rightarrow$ 30) and up to 95% for ViT on PAPILA (768 $\rightarrow$ 32). However, the observed inference-time reductions are modest, typically on the order of 0.01–0.15 ms. This is expected because the backbone dominates the computational cost of forward propagation; the final linear layer accounts for less than 1% of total FLOPs in modern CNN, ViT, and hybrid architectures. As a result, shrinking the input dimension to the classifier head yields limited system-level speedup despite drastically reducing the size of the representation.

Nonetheless, two consistent trends emerge. First, latency never increases after pruning, confirming that GFP introduces no computational overhead. Second, whenever the classifier layer constitutes a non-negligible fraction of total computation (e.g., in smaller backbones or when processing high-resolution inputs), GFP provides small but measurable improvements—for example, 3.21 ms $\rightarrow$ 3.12 ms on Messidor-2 with EfficientNetV2, and 4.32 ms $\rightarrow$ 4.17 ms on CoAtNet. These reductions, although minor in absolute scale, demonstrate that GFP can improve efficiency without altering the backbone or affecting predictive performance. Overall, the inference-time analysis confirms that the primary benefits of GFP lie in compactness, interpretability, and generalization, while its computational footprint remains negligible and, in most cases, slightly advantageous.

Running-Time Analysis of the GFP Algorithm

To validate the theoretical time complexity of the proposed Greedy Feature Pruning (GFP) algorithm, we measure its running time on the DDR training set across a wide range of group sizes $n \in \{1, 2, 4, 8, 16, 32, 64, 128, 256\}$. Because GFP partitions the flattened feature vector into $S = L/n$ groups and evaluates each group once during solo scoring and once during the greedy build-up, the computational cost is dominated by the number of groups. As derived in Section 3, the overall runtime satisfies

$$T(n) \propto \frac{1}{n},$$

implying linear scaling with respect to the inverse group size.

Table 9. Running time (seconds) of the GFP algorithm on the DDR training set for different group sizes n and backbones.

Group size n	GFP runtime on DDR (s)		
	ViT	CoAtNet	EfficientNetV2
1	109.94	155.43	438.75
2	54.72	76.94	219.54
4	27.73	38.56	109.40
8	14.17	19.16	55.55
16	6.99	9.49	27.89
32	3.51	4.85	13.96
64	1.86	2.34	6.79
128	0.88	1.24	3.41
256	0.45	0.82	1.84

Table 9 summarizes the measured runtimes for ViT, CoAtNet, and EfficientNetV2. As expected, the runtime decreases monotonically with increasing n. EfficientNetV2 exhibits the highest running time because it produces the largest flattened feature vector (2152 dimensions), whereas ViT and CoAtNet are faster due to their lower-dimensional representations.

To confirm the predicted linear scaling, we plot $\log_{\frac{1}{2}} T$ against $\log_{\frac{1}{2}}(1/n)$ in Fig. 5. A log–log coordinate system is used to avoid compression of the data points at large n, where runtimes become very small. Under the theoretical

model $T(n) = \alpha(1/n)$, this plot should form a straight line with slope 1. All three backbones follow this trend closely, empirically validating that GFP scales linearly with $1/n$ across several orders of magnitude.

These results confirm that GFP is computationally efficient and scales predictably across architectures. In practice, choosing moderate or large group sizes greatly reduces pruning time while maintaining pruning quality, making GFP practical for real-world medical imaging workflows.

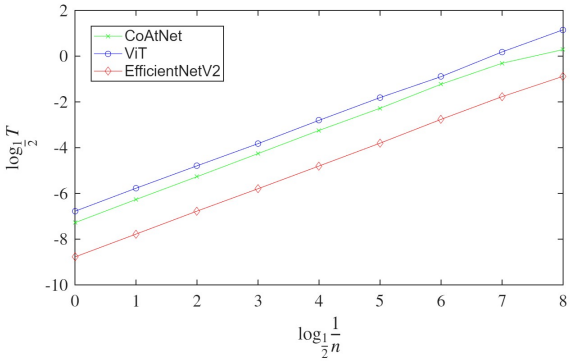

Figure 5. Relationship between $\log_{\frac{1}{2}} T$ and $\log_{\frac{1}{2}}(1/n)$ for different backbones on the DDR dataset. The near-linear trend confirms the theoretical scaling $T(n) \propto 1/n$.