A Severity Score for Retinopathy of Prematurity

Peng Tian
pengtian@ece.neu.edu
Northeastern University
Boston, MA

Yuan Guo
guo.yu@husky.neu.edu
Northeastern University
Boston, MA

Jayashree Kalpathy-Cramer
kalpathy@nmr.mgh.harvard.edu
Massachusetts General Hospital
Charlestown, MA

Susan Ostmo
ostmo@ohsu.edu
Oregon Health & Science University
Portland, OR

John Peter Campbell
campbelp@ohsu.edu
Oregon Health & Science University
Portland, OR

Michael F. Chiang
chiangm@ohsu.edu
Oregon Health & Science University
Portland, OR

Jennifer Dy
jdy@ece.neu.edu
Northeastern University
Boston, MA

Deniz Erdoğmuş
erdogmus@ece.neu.edu
Northeastern University
Boston, MA

Stratis Ioannidis
ioannidis@ece.neu.edu
Northeastern University
Boston, MA

ABSTRACT

Retinopathy of Prematurity (ROP) is a leading cause for childhood blindness worldwide. An automated ROP detection system could significantly improve the chance of a child receiving proper diagnosis and treatment. We propose a means of producing a continuous severity score in an automated fashion, regressed from both (a) diagnostic class labels as well as (b) comparison outcomes. Our generative model combines the two sources, and successfully addresses inherent variability in diagnostic outcomes. In particular, our method exhibits an excellent predictive performance of both diagnostic and comparison outcomes over a broad array of metrics, including AUC, precision, and recall.

KEYWORDS

Retinopathy of Prematurity; learning from comparisons; Bradley-Terry model; classification

ACM Reference Format:
Peng Tian, Yuan Guo, Jayashree Kalpathy-Cramer, Susan Ostmo, John Peter Campbell, Michael F. Chiang, Jennifer Dy, Deniz Erdoğmuş, and Stratis Ioannidis. 2019. A Severity Score for Retinopathy of Prematurity. In *The 25th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD '19), August 4–8, 2019, Anchorage, AK, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3292500.3330713

1 INTRODUCTION

Retinopathy of Prematurity (ROP) is a rapidly-progressive retinal neovascular disease affecting premature infants. It is a leading cause of childhood blindness worldwide, and the societal impacts of infant-acquired visual loss are enormous [25, 26, 31]. The disease incidence is increasing, especially in middle-income countries as survival rates among premature infants rise throughout the world [5, 22]. Most cases of blindness from ROP are avoidable with early diagnosis and treatment, however in many regions of the world access to trained ophthalmologists is severely limited. An automated ROP detection system incorporated into a telemedicine application could significantly improve the chance of a child receiving proper diagnosis and treatment. This could reduce the incidence of blindness without a proportionate increase in the need for human resources, which takes many years to develop.

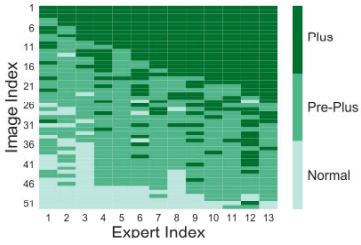


Figure 1: Diagnostic class labels (Plus, Pre-plus, Normal) generated by 13 experts for 52 ROP images. Each column corresponds to an expert, and each row to an image. The color of a cell indicates the diagnostic label attributed by the expert—darker corresponds to most severe (Plus). Images are indexed in increasing order of the number of experts that diagnose them as "Plus". We observe that diagnostic outcomes vary significantly across experts.

Plus disease, defined as arterial tortuosity and venous dilation of the retinal vessels greater than a standard published photograph, is the most important prognostic factor in ROP [18]. The disease classification was originally a binary variable: Plus vs Normal [18]. Recently however an intermediate class Pre-plus was introduced [30]. Based on recent papers suggesting that clinicians have a systematic bias in terms of where to draw the line between categorical classifications along a continuum, it has been proposed to replace such discrete (2 or 3 level) classifications with a continuous severity score [11, 32, 34].

In this paper, we propose a method of producing a continuous severity score in an automated fashion. One method to obtain an automated severity score is to regress an expert obtained diagnosis against image-based disease features that can be manually or

KDD '19, August 4–8, 2019, Anchorage, AK, USA

ACM ISBN 978-1-4503-6201-6/19/08…$15.00
https://doi.org/10.1145/3292500.3330713

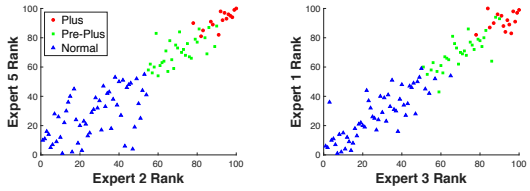


Figure 2: Comparison of image ranks generated by different experts via comparisons. Each expert was asked to compare different images w.r.t. their relative ROP severity. These pairwise comparisons were used to rank the images using the Elo algorithm [20], and ranks across different experts are displayed in each axis. The average correlation coefficient among all pairs of five experts is 0.934 and the standard deviation is 0.017. For each image, we also indicate the Reference Standard Diagnosis (RSD) label (c.f. Section 4). The average correlation coefficient among all pairs of five experts is 0.934 and the standard deviation is 0.017. We observe that resulting ranks are in agreement across experts.

automatically extracted. However, it is well established that there is significant inter-expert variability in ROP diagnosis [15, 42, 46]. This is illustrated in Fig. 1, which shows the diagnostic class labels provided by 13 experts on 52 ROP images. A wide variation across different experts is clearly evident.

In contrast, it was recently demonstrated that, though experts disagree on diagnostic class labels, they are more likely to agree on the *relative* severity of ROP between two images [32]. This is illustrated in Fig. 2. Five experts were presented two images and were asked to choose the one in which ROP was more severe. The average correlation coefficient among all pairs of five experts is 0.934 and the standard deviation is 0.017, which shows that the relative ordering of images induced by expert comparisons is strongly correlated among experts.

Based on these observations, the main goal of this paper is to produce *an automatically generated severity score, based on manually and automatically extracted image features trained from both (a) diagnostic class labels as well as (b) comparison outcomes.* In doing so, we face the challenge of coming up with a single generative model for both class and comparison labels, as well as addressing the inherent noise present in diagnostic labels.

We make the following contributions:

- We propose a multi-expert generative model that combines both diagnostic class *and* comparison labels (i.e., relative severity orderings of image pairs by experts). Our model is flexible, naturally encompassing a variety of loss functions.
- We compare three variants of this model and show that introducing *expert biases* successfully addresses inherent variability in diagnostic class labels. This is true despite the fact that we produce a unique, expert-independent severity score per image, and expert biases are used only in training and not in testing.
- We extensively evaluate the predictive performance of our severity score with respect to a broad array of metrics. We show that when predicting plus labels (comparison labels), it is able to achieve a 0.937 AUC (0.903 AUC) on manually segmented images and a 0.864 AUC (0.813 AUC) on automatically segmented images.
- We show that our produced score has excellent predictive power with respect to Reference Standard Diagnosis (RSD) labels, generated via a rigorous consensus process between three experts. Surprisingly, when training our score on comparison and (noisy) diagnostic class labels, we can predict RSD labels *with higher accuracy than a score trained on RSD labels directly.*
- We also show that our methodology, and our resulting score, is robust to different objectives selected during training.

The remainder of this paper is organized as follows. Section 2 discusses relevant work. Section 3 describes our methodology and introduces our generative model combining diagnostic and comparison labels. Section 4 reports our experimental results, evaluated on not only an ROP dataset, but also two other real-world dataset.

2 RELATED WORK

Several papers propose automated means of measuring ROP features such as diameter, dilation, curvature, and tortuosity. Heneghan et al. [27] segment out vascular structures in retinal images via a morphological filtering stage and predict "Plus" disease based on the width, as well as the tortuosity of the segmented vessels. Gelman et al. [21] and Wallace et al. [47] propose indices for tortuosity and dilation, and establish a correlation between these indices and ROP diagnostic labels via univariate analysis. Wilson et al. [50] use tortuosity and dilation statistics, e.g., mean, median, maximum, and minimum, to represent each vessel, and compute the Spearman correlation coefficient [44] with expert-graded tortuosity and dilations. Cabrera et al. [10] use the features extracted by Wallace et al. [47], and predict "Plus" disease via linear regression. Pour et al. [37] segment images via a local Radon transform [38], extract tortuosity and dilation features, and use k-nearest neighbor [1], Support Vector Machine (SVM) [17] and multilayer perceptrons [19] to predict "Plus" disease. Rajashekar et al. [40] segment vessels by the gaussian matched filter responses [13], extract tortuosity indices as well as the number of vessels, and use a two-class linear discriminant function to predict Aggressive Posterior Retinopathy of Prematurity (APROP) [30], a severe form of ROP, which is characterized by fast progression. Appaji et al. [3] segment the vessels by canny's edge detector [12], extract tortuosity indices, and classify APROP by a 10-layer neural network. Several of these methods require manual tracing of the vessels within a retina image [10, 21, 47, 50]. We follow the feature extraction procedure of [4], which generates 156 features including the indices by [3, 21, 47, 50]. Ataer-Cansızoğlu [4] shows that this outperforms prior feature extraction methods in "Plus" prediction via a SVM classifier; the author also provides an automatic technique for vessel segmentation. We differ from all of the above methods in not relying only on diagnostic class labels, but in extending our training method to incorporate pairwise comparison labels as well.

A significant body of research has recently focused on the related problem of detecting diabetic retinopathy. Techniques used include deep neural networks [23, 35, 39], SVMs [49] and a hybrid model combining Gaussian mixture model, SVM and a multimodel-mediod approach [2]. We depart from these methods both in incorporating pairwise comparisons but also in focusing on ROP rather than diabetic retinopathy. Closer to us, Brown et al. [8] train a Convolutional Neural Network (CNN) to regress consensus ROP diagnostic

labels. We differ from them by a) directly training on noisy diagnostic labels, b) extracting interpretable ROP related features, and c) incorporating the training model with comparison labels.

The generative model we use to combine diagnostic and comparison labels is motivated by the Bradley-Terry model [7, 28, 41], which is a special case of the (more general) Plackett-Luce model [36]. We depart from classic Bradley-Terry by regressing scores from image features, as well as by incorporating class labels. In addition, we show that expert dependent biases can improve the resulting (expert-independent) comparison score generation.

To predict whether a user rates an item, Rendle et al. [41] learn pairwise comparisons via a ridge-regularized logistic regression model. We differ by incorporating class labels in training. Similar to our setting, several works combine regression and ranking, as expressed via comparison labels in diverse tasks, including click prediction [43], image classification [14, 48], and Post-Traumatic Stress Disorder [9]. All of the above works incorporate a penalty capturing the relative ranking among features into a traditional regression loss function; in all cases, the ranking is automatically generated (rather than obtained via experts). For example, Sculley [43] combines a regression loss (e.g., squared loss) and ranking loss (e.g., logistic loss), with comparison labels extracted from inter-class comparisons, and shows that such a penalty is useful in datasets with extremely unbalanced classes. Chen et al. [14], Brown et al. [9], and Wang et al. [48] also combine pointwise labels with pairwise labels extracted from additional external features. We depart from the above works by (a) considering different loss functions, motivated by Bradley-Terry, (b) obtaining labels from multiple experts rather than extracting them from pointwise labels or external features, and (c) proposing a multi-expert model, directly tackling variability among experts not present in this prior work. In the singe expert settings, our SVM-SVM model is equivalent to [14] and [9], c.f. Section 3.3.2.

3 BACKGROUND AND PROBLEM FORMULATION

Our aim is to generate a continuous severity score that combines *both* class *and* comparison labels. We introduce our problem setup and notation in this section.

3.1 Notation and Preliminaries

We consider a dataset of N images, indexed by $i \in \{1, 2, \cdots, N\}$, and M experts, indexed by $e \in \{1, 2, \cdots, M\}$, that label the images. Every image i is represented as d-dimensional feature vector $\boldsymbol{x_i} \in \mathbb{R}^d$. There are two label types: *class labels*, corresponding to a diagnosis by an expert, and *comparison labels*, capturing the outcome of a comparison between two images by an expert.

More specifically, we denote the class label set as D_d: this set consists of tuples of the form (i, e, y_i^e) where $y_i^e \in \{-1, +1\}$ is the class label given by expert e to image i. Without loss of generality (W.l.o.g), we assume binary class labels (indicating, e.g., a 'Plus' vs. not 'Plus' disease diagnosis). Similarly, we denote the comparison label set as D_c: this consists of tuples of the form $(i, j, e, y_{(i,j)}^e)$, where $y_{(i,j)}^e \in \{-1, +1\}$ is the comparison label given by expert e when comparing images i and j with respect to their relative severity. We follow the convention that $y_{(i,j)}^e = +1$ if and only if the severity of image i is higher than the severity of image j. Both our model, as well as the datasets we have collected, contain *no ties*: expert e always selects a comparison label in $\{+1, -1\}$ when presented with two images. We summarize our notations in Table 1.

Table 1: Summary of Notation

N	number of images
M	number of experts
i, j	image indices
e	expert index
$\boldsymbol{x_i}$	feature vector of image i
y_i^e	class label by e for i
$y_{(i,j)}^e$	comparison outcome between i and j by expert e
D_d	dataset of class labels
D_c	dataset of comparison labels
s_i	Bradley-Terry score for image i
$\boldsymbol{\beta}$	parameter vector/model in $\mathbb{R}^d$
b	global bias
b^e	expert-dependent bias
$\boldsymbol{\beta_e}$	expert-dependent model
α	balance parameter in [0, 1]
λ	regularization parameter in $\mathbb{R}_+$

Note that, in the absence of the comparison labels D_c, regressing the class labels D_d is a standard supervised learning problem. Before presenting our joint generative model for both class labels and comparison labels, we discuss how comparison labels can be regressed through the Bradley-Terry model [7, 29].

3.2 A Generative Model for Pairwise Comparisons

The Bradley-Terry model [7, 29] is a generative model for probabilistic comparison outcomes. It is a special case for the more general Plackett-Luce model [36]. Formally, the Bradley-Terry model postulates that every item i in a labelled comparison dataset akin to D_c is parametrized by a (non-random) inherent score s_i. Then, all labelling events in D_c are independent of each other, and their marginal probabilities are:

$$P(y_{(i,j)}^e = +1) = \frac{s_i}{s_i + s_j}, \qquad \forall (i, j, e, y_{(i,j)}^e) \in D_c. \tag{1}$$

In this work, we extend the Bradley-Terry model by introducing a common reparametrization of the scores s_i. In particular, we assume that there exists a common parameter vector/model $\boldsymbol{\beta} \in \mathbb{R}^d$ and a global bias $b \in \mathbb{R}$ such that:

$$s_i = e^{\boldsymbol{\beta}^T \boldsymbol{x}_i + b}, \qquad \forall i \in \{1, 2, \cdots, N\}. \tag{2}$$

Under this assumption, given D_c, the maximum a posteriori (MAP) estimation of $\boldsymbol{\beta}$ amounts to minimizing the following negative log-likelihood function:

$$L_c(\boldsymbol{\beta}; D_c) = \sum_{(i,j,e,y_{(i,j)}^e) \in D_c} \log\left(1 + e^{-y_{(i,j)}^e (\boldsymbol{\beta}^T (\boldsymbol{x_i} - \boldsymbol{x_j}))}\right). \tag{3}$$

Note that this is equivalent to logistic regression on the *difference* $\boldsymbol{x}_i - \boldsymbol{x}_j$ between feature vectors $\boldsymbol{x}_i, \boldsymbol{x}_j$. The loss (3) does *not* depend on the bias term b: indeed, in contrast to class labels and standard regression, biases do not have an effect on pairwise comparisons.

3.3 Combining Class and Comparison Labels

To produce a combined generative model, we assume that given parameter vector/model $\boldsymbol{\beta} \in \mathbb{R}^d$, both class and comparison labels

are independent. Moreover, we further assume that the class labels are generated by a logistic model parametrized *by the same model* $\boldsymbol{\beta}$ *as comparison labels*, i.e., $P(y_i^e = +1) = \frac{e^{\boldsymbol{\beta}^T \boldsymbol{x}_i + b}}{1+e^{\boldsymbol{\beta}^T \boldsymbol{x}_i + b}}$ for all $(i, e, y_i^e) \in D_d$. Given D_d alone (but not the comparisons D_c) the MAP estimation of $\boldsymbol{\beta}$ amounts to minimizing:

$$L_d(\boldsymbol{\beta}; D_d) = \sum_{(i,e,y_i^e)\in D_d} \log(1 + e^{-y_i^e(\boldsymbol{\beta}^T \boldsymbol{x_i} + b)}), \tag{4}$$

which is the classic logistic regression. Using Eq. (3) and Eq. (4), the MAP estimation of $\boldsymbol{\beta}$ given *both* D_d and D_c amounts to minimizing:

$$\begin{aligned} L(\boldsymbol{\beta}; D_d, D_c) = &\sum_{(i,e,y_i^e)\in D_d} \log(1 + e^{-y_i^e(\boldsymbol{\beta}^T \boldsymbol{x_i} + b)}) \\ &+ \sum_{(i,j,e,y^e_{(i,j)})\in D_c} \log(1 + e^{-y^e_{(i,j)}(\boldsymbol{\beta}^T(\boldsymbol{x_i} - \boldsymbol{x_j}))}). \end{aligned} \tag{5}$$

Motivated by Eq. (5), we regress $\boldsymbol{\beta}$ through:

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^d, b\in\mathbb{R}} \alpha L_d(\boldsymbol{\beta}, D_d) + (1-\alpha) L_c(\boldsymbol{\beta}, D_c) + \lambda\|\boldsymbol{\beta}\|_1, \tag{6}$$

where the balance parameter $\alpha \in [0, 1]$ and the regularization parameter $\lambda \in \mathbb{R}_+$ are trained through cross-validation. For $\alpha = 0.5$, Eq. (6) is precisely MAP estimation, assumping a Laplace prior on parameter vector $\boldsymbol{\beta} \in \mathbb{R}^d$. Intuitively, the balance parameter α allows us to establish a tradeoff between class labels (D_d) and comparison labels (D_c): our objective can choose to penalize the training error w.r.t. one dataset more than the other. Note that Eq. (5) is a convex optimization problem, which can be solved via standard techniques (e.g., Newton's method [6]).

3.3.1 Generating Predictions. For all of the above choices of α and λ, the regressed score of an image is given by: $s_i = \exp(\boldsymbol{\beta}^T \boldsymbol{x_i} + b)$, for all $i \in \{1, 2, \cdots, N\}$. As such, the predicted class label for an image i is determined by : $\hat{y}_i^e = \text{sign}(s_i - 1)$, while the predicted outcome of the comparison event between images i and j by expert e is determined by whether $s_i \geq s_j$, i.e., $\hat{y}^e_{(i,j)} = \text{sign}(s_i - s_j)$.

3.3.2 Extensions. More generally, we can replace the logistic penalty function in Eq. (6) by other loss functions. For example, using SVMs, training $\boldsymbol{\beta}$ amounts to:

$$\begin{aligned} \text{Minimize :} \quad & \alpha \sum_{(i,e,y_i^e)\in D_d} \zeta_i^e + (1-\alpha) \sum_{(i,j,e,y^e_{(i,j)})\in D_c} \zeta^e_{(i,j)} + \lambda\|\boldsymbol{\beta}\|_1 \\ \text{subject to:} \quad & y_i^e(\boldsymbol{\beta}^T \boldsymbol{x}_i + b) \geq 1 - \zeta_i^e, \\ & \zeta_i^e \geq 0, \qquad \forall (i, e, y_i^e) \in D_d, \\ & y^e_{(i,j)}(\boldsymbol{\beta}^T(\boldsymbol{x}_i - \boldsymbol{x}_j)) \geq 1 - \zeta^e_{(i,j)}, \\ & \zeta^e_{(i,j)} \geq 0, \quad \forall (i, j, e, y^e_{(i,j)}) \in D_c, \end{aligned}$$

The objective is a weighted combination of linear SVMs; as such, the problem can be solved in a standard fashion, by incorporating constraints in the objective and using a Quasi-Newton method [6]. We consider additional combinations in our experiments Sec. 4, e.g., a logistic loss for class labels and SVM for comparison labels, etc.

3.4 Modelling Multiple Experts

We consider three model variants to measure and account for the discrepancy in class labels among experts:

Figure 3: Comparison Between Manual and Automatic Image Segmentations. An original retina image is shown on the left. A manually segmented mask (middle) is drawn by an ROP expert. The automatic segmented mask (right) is noisier than the manual mask.

Global Model (GM). In the global model, there exists a universal $\boldsymbol{\beta}$ and a universal bias b that satisfy Eq. (2), namely:

$$s_i = \exp(\boldsymbol{\beta}^T \boldsymbol{x}_i + b), \qquad \forall i \in \{1, 2, \cdots, N\}. \tag{7}$$

This presumes a commonality in the behavior among experts and does not account for discrepancies between them.

Global Model with Expert Bias (GMEB). Our second model is:

$$s_i = s_i^e = \exp(\boldsymbol{\beta}^T \boldsymbol{x}_i + b^e), \qquad \forall i \in \{1, 2, \cdots, N\}. \tag{8}$$

Compared to GM, GMEB assumes a common behavior among experts (via the universal model $\boldsymbol{\beta}$), but also allows for a variability in biases (b^e for each expert e). This is intuitively appealing, and partially motivated by Fig. 1 and Fig 2. Indeed, Eq. (8) implies that expert-dependent scores induce the same relative ordering, as in Fig 2, while allowing different behavior in class labelling events (as in Fig. 1). The expert bias has a bearing only in class labels; to see this, note that $s_i^e \geq s_j^e$ if and only if $\boldsymbol{\beta}^T \boldsymbol{x}_i \geq \boldsymbol{\beta}^T \boldsymbol{x}_j$, and, hence, comparison labels by expert e *do not depend* on b^e.

Expert Model (EM). For comparison purposes, we also consider:

$$s_i = s_i^e = \exp(\boldsymbol{\beta_e}^T \boldsymbol{x}_i + b^e), \qquad \forall i \in \{1, 2, \cdots, N\}. \tag{9}$$

Assuming that $\boldsymbol{\beta}_e$ are distinct, this model postulates a complete disagreement among experts. A scenario in which such a model exhibits improved predictive power over GM and GMEB poses a significant obstacle in our analysis: such an outcome would indicate that we *cannot* produce a single discriminative score per image that is consistent with all experts. As we will see, however, this model tends to overfit the training data when compared to GMEB (c.f. Sec. 4.4). This gives us a strong indication that GMEB better captures discrepancies as well as commonalities between experts.

4 EXPERIMENTS

4.1 ROP Dataset

Dataset Collection. Retinal images in our ROP dataset were obtained by a trained photographer using a published protocol [16] following a wide angle digital fundus camera (RetCam camera, Natus Medical Incorporated). These images were then uploaded to the database without any protected health information (PHI). The images were reviewed in a masked and anonymized fashion by study experts. This study was approved by the Institutional Review Board at Oregon Health & Science University and followed the tenets of the Declaration of Helsinki. Written informed consent was obtained from parents of all infants for imaging and study participation.

Image Feature Extraction. We follow the image feature extraction methodology of Ataer-Cansızoğlu [4]. We describe our feature extraction method in detail in Appendix A in the supplement. In

Table 2: Dataset Summary

Dataset	N	d	M	Class Labels	Comparison Labels
ROP	100	156	13	1,300	29,705
FAC	3,520	1024	1	3,520	4,964
Netflix	201	30	53	8,411	442,208

short, this consists of the following three steps. (1) *Segmentation*: We convert the colorful image to a binary mask denoting whether the pixel is on a vessel. (2) *Tracing*: We find the vessel centerline and the vascular tree structure. (3) *Feature Extraction*: We extract 156 features, including tortuosity and dilation statistics, described in detail in the supplement. We use two kinds of image segmentation methods. The first is *manual segmentation*: an ROP expert manually draws the binary segmented mask. The second is *automatic segmentation*: we use the algorithm based on eigenvalue analysis by Ataer-Cansızoğlu [4]. Fig. 3 gives an example of manual and automatic segmentation for an ROP image. Automatic segmentation is noisier, which degrades the quality of extracted features.

Label Generation. We use a dataset of 100 colored retinal images independently graded by 13 ROP experts as Plus, Pre-Plus, or Normal. The resulting dataset D_d comprises 1300 class labels, which we binarize as described in Section 3.1. Five experts additionally performed pairwise comparisons for all 4950 pairs of 100 images. Every expert labelled every pair at least once and certain pairs of images were labelled multiple times by the same expert. When presented with two images, experts always select one as having the most severe presence of the disease (i.e.,there are no ties). The resulting dataset D_c consists of 29, 705 comparison labels.

In addition to these labels, we also have access to consensus /reference standard labels, generated via the following rigorous process involving three experts. First, each of the 100 images above received a clinical diagnosis by the physician monitoring the patient. Subsequently, three medical experts also independently labelled these images, and the majority label was compared to the physician diagnosis. If these two differed, a consensus was reached after discussion between all experts. We refer to such labels as Reference Standard Diagnosis (RSD) labels. This dataset comprises 100 images consisting of 15 Plus, 31 Pre-Plus and 54 Normal labels.

We binarize class labels by using both Plus vs. not Plus and Normal vs. not Normal as possible binary diagnostic labels in our experiments. This is because the three diagnostic label classes (Plus, Pre-Plus and Normal) are ordered by degree of severity, and this ordering information would be lost in, e.g., multinomial regression over the three classes. We confirmed this via preliminary experimentation, in which we observed that binarization outperforms multinomial regression over the three classes[1].

4.2 Additional Datasets

To evaluate the generality of our method, we also conduct experiments on two other real-world datasets, the Netflix and the Filter Aesthetic Comparison (FAC) dataset. We describe these two additional datasets in Appendix C in the supplement. Table 2 summarizes the number of images/movies (N), dimensionality (d), the number of experts (M), the number of class labels and the number of comparison labels for all three datasets

[1]Our code is available at https://github.com/neu-spiral/SeverityScoreForROP

4.3 Experimental Setup

Cross-validation. We evaluate different scores using 5-fold cross-validation. In constructing folds, we ensure that no images in the training set appear in the test set. We also conduct experiments using standard cross-validation: for all experiments and metrics, we observed AUCs higher than the ones we report here. The balance parameter α ranges from 0 to 1 with 0.1 interval and the regularization parameter λ ranges from 10^{-6} to 10^4.

Performance Metrics. We use the area under the curve (AUC), precision, recall, accuracy, and correlation coefficient (CC) to evaluate the performance of different models. For all metrics except AUC, we compute the 95% confidence intervals by repeating measurements 10 times with different cross-validation partitioning, and then determining the standard deviation. For AUC, we conduct the experiment 10 times and report the average. The 95% confidence interval of AUC is 1.96 times the standard deviation σ_A, which is computed as follows [24]:

$$\sigma_A^2 = \frac{1}{mn}\left(A(1-A) + (m-1)(P_{xxy} - A^2) + (n-1)(P_{xyy} - A^2)\right),$$

where A is the AUC, $P_{xxy} = A/(2-A)$, $P_{xyy} = 2A^2/(1+A)$, and m, n are the number of positive and negative samples, respectively.

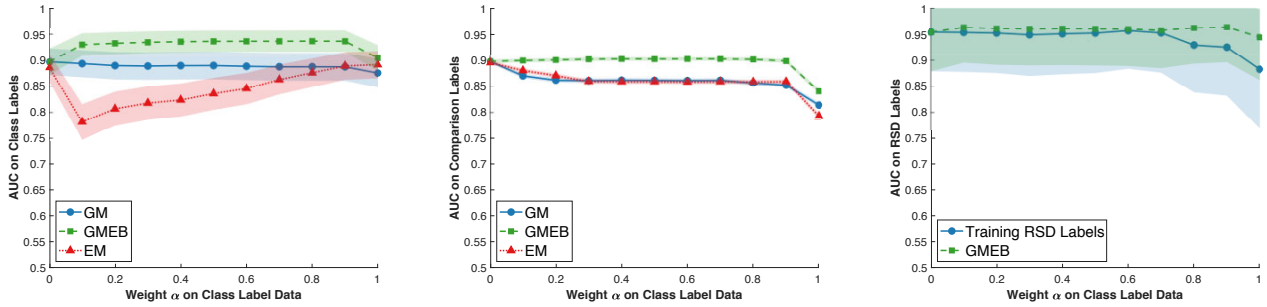
All metrics except CC are classic. CC measures the correlation between the rank of our produced severity scores and an expert consensus rank. We compute the latter by applying the Elo algorithm [20] to all pairwise comparison labels. We describe the Elo algorithm in detail in Appendix B in the supplement. We report the results for all metrics on test folds.

4.4 Comparison of Different Expert Models

Our three expert models under SVM penalties are compared in Fig. 4 for manually segmented features. These figures show the AUC of class label, comparison label and RSD label predictions as a function of balance parameter α. Note that because expert-specific biases b_e are used in Eq. (8), GMEB generates expert-specific scores in predicting class labels. However, expert biases are trained but *not used* in predicting comparison labels in GMEB. Similarly, RSD labels are not expert-specific, and expert biases b_e are trained but *not used* in predicting RSD labels. Finally, EM generates expert specific scores.

We observe several important trends. First, as shown in Fig. 4a, GMEB outperforms GM and EM. Second, also in Fig. 4a, we see that GM attains its optimal AUC at $\alpha = 0$. Hence, surprisingly, it is optimal for GM to completely ignore class labels *even when tasked with predicting class labels*. This is precisely due to the inherent noise in class labels on account of expert disagreement (c.f. in Fig. 1), which GM does not model. In contrast, GMEB attains an optimal AUC at $\alpha = 0.8$, which indicates that it successfully leverages both class labels and comparison labels. EM performs worse than GMEB and GM because it overfits. We observe the same trends for automatically segmented features in Fig. 10 in the Supplement.

We also observe the same trends in Table 3, which shows the AUC of class label and comparison label predictions under different models and different penalty combinations, for both manually and automatically segmented images. Universally, GMEB outperforms the other two models.



(a) Plus vs Not Plus of Manually Segmented Result on Diagnostic Labels

(b) Plus vs Not Plus of Manually Segmented Result on Comparison Labels

(c) Plus vs Not Plus of Manually Segmented Result on RSD Labels

Figure 4: Comparison between Expert representation models on manually segmented features under SVM penalties. The shaded area indicates 95% confidence interval. Note that GMEB uses expert specific scores in predicting diagnostic labels, however, expert biases *are not used* in predicting comparison and RSD labels. EM uses expert specific scores. We observed several important trends. First, GMEB outperforms GM and EM. Second, we see that GM attains its optimal AUC at $\alpha = 0$. Hence, surprisingly, it is optimal for GM to completely ignore class labels even when tasked with predicting class labels. In contrast, GMEB attains an optimal AUC at $\alpha = 0.8$, which indicates that it successfully leverages both class labels and comparison labels. Then, EM performs worse than GMEB and GM because it overfits. Finally, GMEB model trained with class label outperforms the model trained with RSD labels even in predicting RSD labels.

Table 3: Multi-Expert AUC Results on Manually and Automatically Segmented Features

Model		Manually Segmented Features				Automatically Segmented Features			
		Plus vs Not Plus		Normal vs Not Normal		Plus vs Not Plus		Normal vs Not Normal	
		Class Labels (α)	Comparison Labels (α)	Class Labels (α)	Comparison Labels (α)	Class Labels (α)	Comparison Labels (α)	Class Labels (α)	Comparison Labels (α)
Log-Log	GM	0.900 ± 0.025 (0)	0.903 ± 0.004 (0)	0.918 ± 0.016 (1)	0.903 ± 0.004 (0)	0.818 ± 0.031 (0.2)	0.810 ± 0.006 (0)	0.827 ± 0.024 (0.8)	0.805 ± 0.006 (0)
	GMEB	**0.937 ± 0.022 (0.1)**	**0.903 ± 0.004 (0)**	0.947 ± 0.014 (0.6)	**0.904 ± 0.004 (0.7)**	**0.864 ± 0.029 (0.9)**	**0.813 ± 0.006 (0.9)**	0.843 ± 0.024 (0.9)	0.805 ± 0.006 (0.7)
	EM	0.899 ± 0.025 (0.9)	0.896 ± 0.004 (0)	0.918 ± 0.016 (0)	0.895 ± 0.004 (0)	0.804 ± 0.033 (0.9)	0.801 ± 0.006 (0)	0.806 ± 0.025 (0.9)	0.795 ± 0.006 (0)
Log-SVM	GM	0.897 ± 0.025 (0)	0.898 ± 0.004 (0)	0.916 ± 0.016 (1)	0.896 ± 0.004 (0)	0.817 ± 0.033 (1)	0.800 ± 0.006 (0)	0.821 ± 0.024 (0.9)	0.795 ± 0.006 (0.9)
	GMEB	0.935 ± 0.022 (0.6)	0.902 ± 0.004 (0.4)	0.943 ± 0.014 (0.8)	0.902 ± 0.004 (0.8)	0.857 ± 0.029 (1)	0.808 ± 0.006 (0.9)	0.832 ± 0.024 (0.9)	0.797 ± 0.006 (0.8)
	EM	0.897 ± 0.025 (1)	0.896 ± 0.004 (0)	0.916 ± 0.016 (0)	0.894 ± 0.004 (0)	0.800 ± 0.033 (0.9)	0.802 ± 0.006 (0)	0.800 ± 0.025 (1)	0.794 ± 0.006 (0)
SVM-Log	GM	0.900 ± 0.025 (0)	0.903 ± 0.004 (0)	0.915 ± 0.016 (0)	0.903 ± 0.004 (0)	0.820 ± 0.031 (0.1)	0.810 ± 0.006 (0)	0.832 ± 0.024 (0.9)	0.805 ± 0.006 (0)
	GMEB	0.937 ± 0.020 (0.5)	0.903 ± 0.004 (0)	**0.949 ± 0.014 (0.3)**	0.903 ± 0.004 (0)	0.861 ± 0.029 (0.7)	0.811 ± 0.006 (0.8)	**0.844 ± 0.022 (0.8)**	**0.807 ± 0.006 (0.7)**
	EM	0.892 ± 0.025 (1)	0.896 ± 0.004 (0)	0.918 ± 0.016 (0)	0.895 ± 0.004 (0)	0.793 ± 0.033 (0.9)	0.801 ± 0.006 (0)	0.805 ± 0.024 (0.9)	0.795 ± 0.006 (0)
SVM-SVM	GM	0.897 ± 0.025 (0)	0.898 ± 0.004 (0)	0.904 ± 0.018 (0.8)	0.896 ± 0.004 (0)	0.814 ± 0.033 (0.6)	0.798 ± 0.006 (0)	0.830 ± 0.024 (0.9)	0.797 ± 0.006 (0.8)
	GMEB	0.937 ± 0.020 (0.8)	0.903 ± 0.004 (0.6)	0.946 ± 0.014 (0.9)	0.902 ± 0.004 (0.8)	0.859 ± 0.029 (0.7)	0.810 ± 0.006 (0.9)	0.839 ± 0.022 (0.9)	0.802 ± 0.006 (0.9)
	EM	0.892 ± 0.025 (1)	0.896 ± 0.004 (0)	0.916 ± 0.016 (0)	0.894 ± 0.004 (0)	0.791 ± 0.033 (0.9)	0.802 ± 0.006 (0)	0.799 ± 0.025 (0.9)	0.794 ± 0.006 (0)

Log-SVM represents the logistic loss for class labels and SVM loss for comparison labels. Similarly, SVM-Log represents the SVM loss for class labels and logistic loss for comparison labels.

Recall that the expert bias is not used in predicting comparison labels in GMEB. Nevertheless, GMEB outperforms GM in comparison label AUC. This indicates that accounting for expert biases when training leads to an improved score when testing, *even if trained biases are not used in testing*. This is further supported by Table 4: in these experiments, we train GMEB on comparison and class labels and use $\boldsymbol{\beta}$ (but not b_e) to predict RSD labels. Surprisingly, in many cases, *prediction based on GMEB outperforms regressing RSD labels from RSD labels*! This too offers a strong indication that GMEB indeed captures–and negates–inherent properties of the expert discrepancies. Even though class labels are noisy, GMEB (a) accounts for this noise correctly, and (b) has access to more labels (class labels) compared to regressing from RSD labels directly, thereby leading to better predictions.

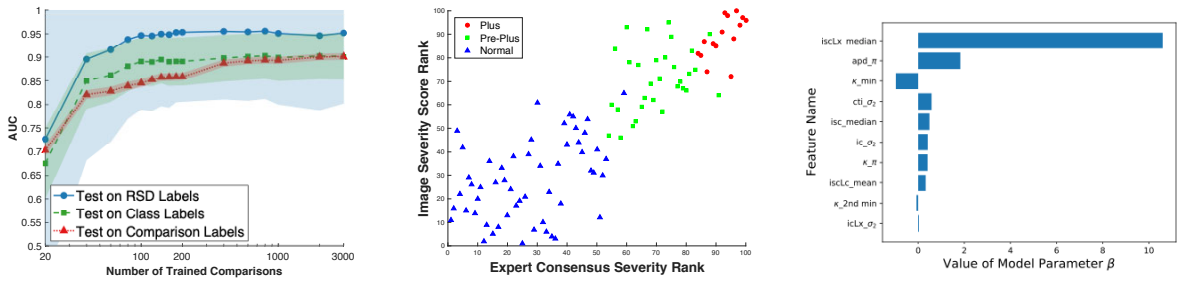
Again, in almost all cases (Table 3), the optimal α is in $(0, 1)$, indicating that training GMEB successfully leverages both diagnostic labels and comparison labels.

As the number of comparison labels is quadratic to the number of class labels, collecting comparison labels is much more challenging than collecting class labels. Thus, we further evaluate the performance of our model with training under a variable number of comparison labels. We randomly select comparison labels, and train

Table 4: Single Expert AUC Results on Manually and Automatically Segmented Features

Model		RSD Labels (α)			
		Manually Extracted Features		Automatically Extracted Features	
		Plus vs Not Plus	Normal vs Not Normal	Plus vs Not Plus	Normal vs Not Normal
Log-Log	Training RSD	0.952 ± 0.076 (0.1)	0.950 ± 0.047 (0.1)	0.845 ± 0.127 (1)	0.850 ± 0.078 (0.9)
	GMEB	**0.966 ± 0.069 (0.9)**	**0.956 ± 0.043 (1)**	**0.895 ± 0.110 (1)**	0.836 ± 0.082 (1)
Log-SVM	Training RSD	0.958 ± 0.073 (0.6)	0.946 ± 0.047 (0.6)	0.816 ± 0.137 (1)	0.843 ± 0.080 (0.9)
	GMEB	0.961 ± 0.071 (1)	0.949 ± 0.047 (1)	0.873 ± 0.118 (1)	0.849 ± 0.078 (1)
SVM-Log	Training RSD	0.952 ± 0.076 (0.1)	0.949 ± 0.047 (0.1)	0.836 ± 0.131 (0)	0.857 ± 0.076 (0.8)
	GMEB	0.960 ± 0.071 (0.9)	0.952 ± 0.045 (0.5)	0.878 ± 0.116 (1)	**0.860 ± 0.074 (1)**
SVM-SVM	Training RSD	0.957 ± 0.073 (0.6)	0.940 ± 0.073 (0.6)	0.806 ± 0.139 (0.7)	0.849 ± 0.078 (0.9)
	GMEB	0.959 ± 0.071 (0.9)	0.946 ± 0.049 (0.7)	0.878 ± 0.116 (1)	0.860 ± 0.076 (1)

our GMEB/GM model at $\alpha = 0$ (no class labels is trained; GM and GMEB are equivalent.) under Logistic penalties. In Fig. 5a , we show the correlation between the number of trained comparison labels under Logistic penalties with $\alpha = 0$ (comparison only) and the AUC on RSD labels (Plus vs Not Plus), class labels (Plus vs Not Plus) and comparison labels. Training with only 40 comparison labels, our model achieves over 0.8 AUCs on RSD labels, class labels and comparison labels. Overall, to reach the same performance as training over 3000 comparison labels, it suffices to use only 180 comparison



(a) (b) (c)

Figure 5: (a) Correlation between the number of trained comparison labels under Log-Log model with $\alpha = 0$ (comparison only) and the AUC of predicting RSD labels (Plus vs Not Plus), class labels (Plus vs Not Plus) and comparison labels. With training only 40 comparison labels, our model achieves over 0.8 AUCs on RSD labels, class labels and comparison labels. Moreover, to reach the almost equivalent performance with training 3000 comparison labels, our model needs only 180 comparison labels to predict RSD labels, 400 comparison labels to predict class labels, and 600 comparison labels to predict comparison labels. (b) Correlation between severity score rank and expert rank. The severity score generated from manually segmented images The legend indicates the RSD classification (Plus, Pre-Plus, Normal). (c) The model parameter β.

Table 5: Image Severity Score Metric of Plus vs Not Plus on Manually Segmented Feature

Metric	AAED	AC	AAEC	Recall	Precision	Accuracy	CC	AO
AAEA	**0.928 ± 0.010**	0.898 ± 0.002	0.913 ± 0.006	0.933 ± 0.031	0.884 ± 0.026	0.905 ± 0.019	0.806 ± 0.045	0.890 ± 0.012
AC	0.920 ± 0.015	**0.903 ± 0.002**	0.911 ± 0.008	0.907 ± 0.056	0.875 ± 0.035	0.888 ± 0.026	0.792 ± 0.065	0.882 ± 0.025
AAEC	0.926 ± 0.009	0.902 ± 0.002	**0.914 ± 0.005**	0.940 ± 0.021	0.867 ± 0.020	0.898 ± 0.016	0.804 ± 0.047	0.889 ± 0.014
Recall	0.909 ± 0.005	0.843 ± 0.002	0.876 ± 0.004	**0.973 ± 0.034**	0.891 ± 0.016	**0.927 ± 0.015**	0.717 ± 0.018	0.860 ± 0.006
Precision	0.919 ± 0.009	0.858 ± 0.002	0.889 ± 0.005	0.920 ± 0.053	**0.920 ± 0.019**	0.919 ± 0.018	0.773 ± 0.022	0.880 ± 0.014
Accuracy	0.909 ± 0.005	0.843 ± 0.002	0.876 ± 0.004	**0.973 ± 0.034**	0.891 ± 0.016	**0.927 ± 0.015**	0.717 ± 0.018	0.868 ± 0.007
CC	0.927 ± 0.009	0.900 ± 0.002	0.914 ± 0.006	0.940 ± 0.049	0.880 ± 0.020	0.905 ± 0.022	**0.807 ± 0.045**	**0.911 ± 0.012**

labels to predict RSD labels, 400 comparison labels to predict class labels, and 600 comparison labels to predict comparison labels.

4.5 Towards a Single Score

Our analysis so far shows that regressing a GMEB model through Eq. (6) has excellent predictive performance in terms of class, comparison, and RSD label AUC. Regressing with each of these metrics as a target leads to different optimal α, λ, as well as different optimal logistic and SVM combinations.

As mentioned in Section 1, recent papers suggest using a continuous severity score to replace discrete classification [11, 32, 34]. To that end, we would like to produce a single, unique score for each image. One clearly could do so by targeting a single metric in cross-validation, or by combining the above metrics in an ensemble. It is not a priori clear which of the above metrics should be used as a target, or how they should be combined. To address this, we conduct experiments to assess the impact of each metric selection. Note that, to produce a single, expert-independent score in predictions appearing in Table 5 and Fig. 5b below, we train the GMEB model with expert biases b_e, but *do not use them* when testing.

Table 5 summarizes the results of this experiment. Each row corresponds to training a GMEB model with respect to a given metric indicated in the left column: optimal parameters α, λ, and logistic/SVM combinations are selected through cross validation for this target metric. In each column, we show the cross-validated performance of the resulting score w.r.t. *other* metrics.

Metrics used are as follows: (a) Average AUC for expert diagnostic class labels (AAED) computed by predicting each expert's class labels and averaging AUCs across experts, (b) AUC on comparison labels (AC), (c) Average AUC on expert class labels and comparison labels (AAEC), which is between the average of AAEA and AC, (d) Recall, (e) Precision, (f) Accuracy with respect to RSD labels, where the threshold is selected at the point on the ROC curve nearest to the upper left corner (true positive rate equals to 1 and false positive rate equals to 0), (g) Correlation coefficient (CC), which is the coefficient between the severity score rank and expert consensus rank, and (h) Average Off-diagonal (AO) which averages the off diagonal elements of each row. As expected, diagonal elements are highest, as cross-validation parameters are optimized.

Overall, we observe excellent performance across all scores, with almost all being highly correlated. Though CC is an outlier, it is also considerably high when training with respect to other metrics. The higher AO is indeed when training w.r.t. CC, as optimizing for it comes with a low penalty for other metrics.

All in all, these results indicate that picking a single metric to optimize can indeed be done without a significant trade-off towards other metrics.

4.6 Investigating the Optimal Score

Training w.r.t. CC has excellent performance w.r.t other metrics; this is illustrated in the Average Other (AO) column of Table 5. We thus focus in this section on the properties of this specific score. The optimal regularization parameters for this score are $\alpha = 0.8$, $\lambda = 100$, for an objective containing SVMs for both class labels and comparison labels.

Fig. 5b further illustrates the correlation between the severity score rank and the expert consensus severity rank. As we mention in Section 4.1, the expert consensus severity rank is generated through the Elo algorithm [20] based on the comparison labels among all experts. This figure illustrates that our severity score has strong agreement with the induced consensus rank.

In Fig. 5c, we plot the 10 most significant coefficients of the trained vector $\boldsymbol{\beta}$. Under a threshold of 10^{-5}, the support of $\boldsymbol{\beta}$ is 39 features. The most significant feature is *iscLx_median*. This is

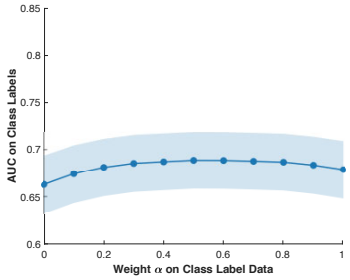


Figure 6: AUC of class labels on FAC as a function of α under SVM penalties. The shaded area indicates 95% confidence interval. Note that EM uses expert specific scores.

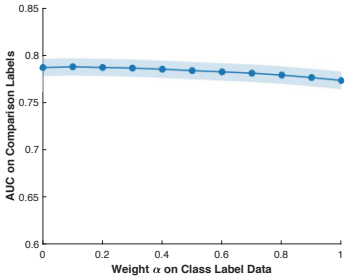


Figure 7: AUC of comparison labels on FAC as a function of α under SVM penalties. The shaded area indicates 95% confidence interval. Note that EM uses expert specific scores.

Table 6: AUC Results on FAC Dataset

Model	Class Labels (α)	Comparison Labels (α)
Log-Log	0.691 ± 0.030 (0.6)	0.782 ± 0.009 (0)
Log-SVM	0.691 ± 0.030 (0.8)	0.788 ± 0.009 (0.1)
SVM-Log	**0.689 ± 0.030 (0.4)**	0.782 ± 0.009 (0)
SVM-SVM	0.689 ± 0.030 (0.5)	**0.788 ± 0.009 (0.1)**

Log-SVM represents the logistic loss for class labels and SVM loss for comparison labels. Similarly, SVM-Log represents the SVM loss for class labels and logistic loss for comparison labels.

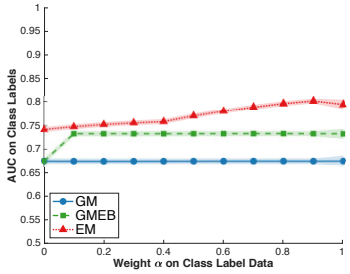


Figure 8: Comparison between expert representation models of class label prediction on Netflix under Logistic penalties. The shaded area indicates 95% confidence interval. Note that EM uses expert specific scores.

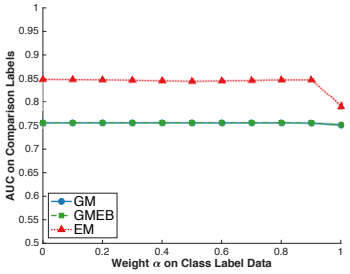


Figure 9: Comparison between expert representation models of comparison label prediction on Netflix under Logistic penalties. The shaded area indicates 95% confidence interval. Note that EM uses expert specific scores.

Table 7: Multi-Expert AUC Results on Netflix Dataset

Model		Class Labels (α)	Comparison Labels (α)
Log-Log	GM	0.676 ± 0.011 (1)	0.756 ± 0.002 (0.3)
	GMEB	0.733 ± 0.005 (0.8)	0.756 ± 0.002 (0.8)
	EM	**0.801 ± 0.005 (0.9)**	**0.848 ± 0.001 (0)**
Log-SVM	GM	0.676 ± 0.006 (0.7)	0.755 ± 0.002 (0.5)
	GMEB	0.733 ± 0.006 (0.8)	0.754 ± 0.002 (0.8)
	EM	0.792 ± 0.005 (0.9)	0.842 ± 0.001 (0.4)
SVM-Log	GM	0.676 ± 0.006 (0.3)	0.756 ± 0.002 (0.2)
	GMEB	0.731 ± 0.006 (0.8)	0.756 ± 0.002 (0.6)
	EM	0.800 ± 0.005 (0.9)	**0.848 ± 0.001 (0)**
SVM-SVM	GM	0.676 ± 0.006 (0.7)	0.755 ± 0.002 (0.4)
	GMEB	0.731 ± 0.006 (0.7)	0.754 ± 0.001(0)
	EM	0.792 ± 0.005 (0.9)	0.847 ± 0.001 (0.2)

Log-SVM represents the logistic loss for class labels and SVM loss for comparison labels. Similarly, SVM-Log represents the SVM loss for class labels and logistic loss for comparison labels.

the median value of *iscLx*, which is the integrated squared curvature value based on vessels. The third most important feature is minimum curvature, whose correlation is negative: this indicates that the effect of curvature is relative, as the minimum value in the image acts as an offset. Intuitively, the features shown in Fig. 5c capture both dilation and tortuosity which is consistent with the most important ROP features defined by experts.

4.7 Evaluations on Additional Datasets

To evaluate the generality of our method, we also report the results on the FAC and Netflix datasets. Fig. 6 and Fig. 7 show the AUC of class labels and comparison labels on FAC as a function of α. In Fig. 6, GM model attains its optimum to predict class labels at $\alpha = 0.5$, which indicates that incorporating comparisons into GM model increases its predictive power on class labels. Similarly, in Fig. 7, the optimal α to predict comparisons labels is 0.1. We observe the same trend in Table 6, which shows the AUC of class label and comparison label predictions under penalty combinations.

In Figures 8 and 9, we compare three expert models on Netflix under Logistic penalties. Note that EM uses expert specific scores to predict class and comparison labels, and GMEB uses expert specific scores to predict class labels. EM outperforms other methods in Fig. 8 and Fig. 9 because it captures the personal preferences on movies. EM attains its optimum to predict class labels at $\alpha = 0.9$. This shows that EM successfully leverages both class and comparison labels. GMEB outperforms GM in Fig. 8 because it has more expert flexibility than GM. GM performs worse because there does not exist an agreement over all experts on rating movies. We find the same trend under different penalty combinations in Table 7.

5 CONCLUSION

We present a multi-expert generative model for ROP that combines both class labels and comparison labels. We find that introducing the expert biases significantly improves the AUC of class, comparison, and RSD label prediction. We show that the single severity score we produced, based on our generative model, has excellent performance and generalizes over an array of important metrics. Although we attain excellent performance on manually segmented features, the performance on automatically segmented features leaves room for improvement.

ACKNOWLEDGMENTS

Our work is supported by NIH (R01EY019474, P30EY10572), NSF (SCH-1622542 at MGH; SCH-1622536 at Northeastern; SCH-1622679 at OHSU), and by unrestricted departmental funding from Research to Prevent Blindness (OHSU).

REFERENCES

[1] JH Ahlberg, EN Nilson, and JL Walsh. 1965. Best Approximation and Convergence Properties of Higher-Order Spline Approximations. *Journal of Mathematics and Mechanics* 14, 2 (1965), 231–243.

[2] M Usman Akram, Shehzad Khalid, and Shoab A Khan. 2013. Identification and classification of microaneurysms for early detection of diabetic retinopathy. *Pattern Recognition* 46, 1 (2013), 107–116.

[3] Abhishek M Appaji, HN Suma, MS Madhurya, S Maria Sonia, and Anand Vinekar. 2017. Comprehensive analysis of retinopathy of prematurity based on tortuosity with development of web connectivity. In *Intelligent Systems and Knowledge Engineering (ISKE), 2017 12th International Conference on*. IEEE, 1–5.

[4] Esra Ataer-Cansızoğlu. 2015. *Retinal image analytics: A complete framework from segmentation to diagnosis*. Northeastern University.

[5] Hannah Blencowe, Joy E Lawn, Thomas Vazquez, Alistair Fielder, and Clare Gilbert. 2013. Preterm-associated visual impairment and estimates of retinopathy of prematurity at regional and global levels for 2010. *Pediatric Research* 74, S1 (2013), 35–49.

[6] Stephen Boyd and Lieven Vandenberghe. 2004. *Convex optimization*. Cambridge university press.

[7] Ralph Allan Bradley and Milton E Terry. 1952. Rank Analysis of Incomplete Block Designs: I. The Method of Paired Comparisons. *Biometrika* 39, 3/4 (1952).

[8] James M Brown, J Peter Campbell, Andrew Beers, Ken Chang, Susan Ostmo, RV Paul Chan, Jennifer Dy, Deniz Erdogmus, Stratis Ioannidis, Jayashree Kalpathy-Cramer, et al. 2018. Automated Diagnosis of Plus Disease in Retinopathy of Prematurity Using Deep Convolutional Neural Networks. *JAMA ophthalmology* (2018).

[9] Sarah Marie Brown, Andrea Webb, Rami Mangoubi, and Jennifer G Dy. 2015. A Sparse Combined Regression-Classification Formulation for Learning a Physiological Alternative to Clinical Post-Traumatic Stress Disorder Scores.. In *AAAI*.

[10] Michelle T Cabrera, Sharon F Freedman, Amanda E Kiely, Michael F Chiang, and David K Wallace. 2011. Combining ROPtool Measurements of Vascular Tortuosity and Width to Quantify Plus Disease in Retinopathy of Prematurity. *Journal of American Association for Pediatric Ophthalmology and Strabismus* 15, 1 (2011).

[11] J Peter Campbell, Jayashree Kalpathy-Cramer, Deniz Erdogmus, Peng Tian, Dharanish Kedarisetti, Chace Moleta, James D Reynolds, Kelly Hutcheson, Michael J Shapiro, Michael X Repka, et al. 2016. Plus Disease in Retinopathy of Prematurity: A Continuous Spectrum of Vascular Abnormality as a Basis of Diagnostic Variability. *Ophthalmology* 123, 11 (2016), 2338–2344.

[12] John Canny. 1986. A computational approach to edge detection. *IEEE Transactions on pattern analysis and machine intelligence* 6 (1986), 679–698.

[13] Thitiporn Chanwimaluang and Guoliang Fan. 2003. An efficient algorithm for extraction of anatomical structures in retinal images. In *Image Processing, 2003. ICIP 2003. Proceedings. 2003 International Conference on*, Vol. 1. IEEE, I–1093.

[14] Lin Chen, Peng Zhang, and Baoxin Li. 2015. Fusing Pointwise and Pairwise Labels for Supporting User-adaptive Image Retrieval. In *Proceedings of the 5th ACM on International Conference on Multimedia Retrieval*. ACM, 67–74.

[15] Michael F Chiang, Lei Jiang, Rony Gelman, Yunling E Du, and John T Flynn. 2007. Interexpert Agreement of Plus Disease Diagnosis in Retinopathy of Prematurity. *Archives of Ophthalmology* 125, 7 (2007), 875–880.

[16] Michael F Chiang, Lu Wang, Mihai Busuioc, Yunling E Du, Patrick Chan, Steven A Kane, Thomas C Lee, David J Weissgold, Audina M Berrocal, Osode Coki, et al. 2007. Telemedical retinopathy of prematurity diagnosis: accuracy, reliability, and image quality. *Archives of ophthalmology* 125, 11 (2007), 1531–1538.

[17] Corinna Cortes and Vladimir Vapnik. 1995. Support-vector networks. *Machine learning* 20, 3 (1995), 273–297.

[18] Cryotherapy for Retinopathy of Prematurity Cooperative Group and others. 1988. Multicenter Trial of Cryotherapy for Retinopathy of Prematurity: Preliminary Results. *Archives of Ophthalmology* 106, 4 (1988), 471–479.

[19] George Cybenko. 1989. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems (MCSS)* 2, 4 (1989), 303–314.

[20] Arpad E Elo. 1978. *The rating of chess players, past and present*. Arco Pub.

[21] Rony Gelman, M Elena Martinez-Perez, Deborah K Vanderveen, Anne Moskowitz, and Anne B Fulton. 2005. Diagnosis of plus disease in retinopathy of prematurity using Retinal Image multiScale Analysis. *Investigative Ophthalmology & Visual Science* 46, 12 (2005), 4734–4738.

[22] Clare Gilbert, Alistair Fielder, Luz Gordillo, Graham Quinn, Renato Semiglia, Patricia Visintin, Andrea Zin, et al. 2005. Characteristics of Infants With Severe Retinopathy of Prematurity in Countries With Low, Moderate, and High Levels of Development: Implications for Screening Programs. *Pediatrics* 115, 5 (2005).

[23] Varun Gulshan, Lily Peng, Marc Coram, Martin C Stumpe, Derek Wu, Arunachalam Narayanaswamy, Subhashini Venugopalan, Kasumi Widner, Tom Madams, Jorge Cuadros, et al. 2016. Development and Validation of a Deep Learning Algorithm for Detection of Diabetic Retinopathy in Retinal Fundus Photographs. *JAMA* 316, 22 (2016), 2402–2410.

[24] James A Hanley and Barbara J McNeil. 1982. The Meaning and Use of the Area under a Receiver Operating Characteristic (ROC) curve. *Radiology* 143, 1 (1982).

[25] M Elizabeth Hartnett and John S Penn. 2012. Mechanisms and Management of Retinopathy of Prematurity. *New England Journal of Medicine* 367, 26 (2012).

[26] Ann Hellström, Lois EH Smith, and Olaf Dammann. 2013. Retinopathy of Prematurity. *The lancet* 382, 9902 (2013), 1445–1457.

[27] Conor Heneghan, John Flynn, Michael O'Keefe, and Mark Cahill. 2002. Characterization of changes in blood vessel width and tortuosity in retinopathy of prematurity using image analysis. *Medical image analysis* 6, 4 (2002), 407–429.

[28] David R Hunter. 2004. MM algorithms for generalized Bradley-Terry models. *Annals of Statistics* (2004), 384–406.

[29] Peter B Imrey. 1998. Bradley–Terry Model. *Encyclopedia of Biostatistics* (1998).

[30] International Committee for the Classification of Retinopathy of Prematurity. 2005. The international classification of retinopathy of prematurity revisited. *JAMA Ophthalmology* 123, 7 (2005), 991–999.

[31] Jonathan Javitt, Ronald Dei Cas, and Yen-pin Chiang. 1993. Cost–Effectiveness of Screening and Cryotherapy for Threshold Retinopathy of Prematurity. *Pediatrics* 91, 5 (1993), 859–866.

[32] Jayashree Kalpathy-Cramer, J Peter Campbell, Deniz Erdoğmuş, Peng Tian, Dharanish Kedarisetti, Chace Moleta, James D Reynolds, Kelly Hutcheson, Michael J Shapiro, Michael X Repka, et al. 2016. Plus Disease in Retinopathy of Prematurity: Improving Diagnosis by Ranking Disease Severity and Using Quantitative Image Analysis. *Ophthalmology* 123, 11 (2016), 2345–2351.

[33] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer* 8 (2009), 30–37.

[34] Chace Moleta, J Peter Campbell, Jayashree Kalpathy-Cramer, RV Paul Chan, Susan Ostmo, Karyn Jonas, Michael F Chiang, Imaging & Informatics in ROP Research Consortium, et al. 2017. Plus Disease in Retinopathy of Prematurity: Diagnostic Trends in 2016 vs. 2007. *American Journal of Ophthalmology* (2017).

[35] Ricky Parmar and Ramanathan Lakshmanan. 2017. Detecting Diabetic Retinopathy from Retinal Images Using CUDA Deep Neural Network. *International Journal of Intelligent Engineering and Systems* 10, 4 (2017), 284–292.

[36] Robin L Plackett. 1975. The analysis of permutations. *Applied Statistics* (1975).

[37] Elias Khalili Pour, Hamidreza Pourreza, Kambiz Ameli Zamani, Alireza Mahmoudi, Arash Mir Mohammad Sadeghi, Mahla Shadravan, Reza Karkhaneh, Ramak Rouhi Pour, and Mohammad Riazi Esfahani. 2017. Retinopathy of Prematurity-assist: Novel Software for Detecting Plus Disease. *Korean Journal of Ophthalmology* 31 (2017), 0.

[38] Reza Pourreza, Touka Banaee, Hamidreza Pourreza, and Ramin Daneshvar Kakhki. 2008. A Radon transform based approach for extraction of blood vessels in conjunctival images. In *Mexican International Conference on Artificial Intelligence*. Springer, 948–956.

[39] Gwenolé Quellec, Katia Charrière, Yassine Boudi, Béatrice Cochener, and Mathieu Lamard. 2017. Deep image mining for diabetic retinopathy screening. *Medical Image Analysis* 39 (2017), 178–193.

[40] Deepthi Rajashekar, Gowri Srinivasa, and Anand Vinekar. 2016. Comprehensive Retinal Image Analysis for Aggressive Posterior Retinopathy of Prematurity. *PloS one* 11, 10 (2016).

[41] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme. 2009. BPR: Bayesian personalized ranking from implicit feedback. In *Proceedings of the twenty-fifth conference on uncertainty in artificial intelligence*. AUAI Press.

[42] James D Reynolds, Velma Dobson, Graham E Quinn, Alistair R Fielder, Earl A Palmer, Richard A Saunders, Robert J Hardy, Dale L Phelps, John D Baker, Michael T Trese, et al. 2002. Evidence-based screening criteria for retinopathy of prematurity: natural history data from the CRYO-ROP and LIGHT-ROP studies. *Archives of Ophthalmology* 120, 11 (2002), 1470–1476.

[43] David Sculley. 2010. Combined regression and ranking. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM.

[44] Charles Spearman. 1904. The proof and measurement of association between two things. *The American Journal of Psychology* 15, 1 (1904), 72–101.

[45] Wei-Tse Sun, Ting-Hsuan Chao, Yin-Hsi Kuo, and Winston H Hsu. 2017. Photo filter recommendation by category-aware aesthetic learning. *IEEE Transactions on Multimedia* 19, 8 (2017), 1870–1880.

[46] David K Wallace, Graham E Quinn, Sharon F Freedman, and Michael F Chiang. 2008. Agreement among pediatric ophthalmologists in diagnosing plus and pre-plus disease in retinopathy of prematurity. *Journal of American Association for Pediatric Ophthalmology and Strabismus* 12, 4 (2008), 352–356.

[47] David K Wallace, Zheen Zhao, and Sharon F Freedman. 2007. A pilot study using "ROPtool" to quantify plus disease in retinopathy of prematurity. *Journal of American Association for Pediatric Ophthalmology and Strabismus* 11, 4 (2007).

[48] Yilin Wang, Suhang Wang, Jiliang Tang, Huan Liu, and Baoxin Li. 2016. PPP: Joint pointwise and pairwise image label prediction. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 6005–6013.

[49] RA Welikala, Jamshid Dehmeshki, Andreas Hoppe, V Tah, S Mann, Thomas H Williamson, and SA Barman. 2014. Automated detection of proliferative diabetic retinopathy using a modified line operator and dual classification. *Computer Methods and Programs in Biomedicine* 114, 3 (2014), 247–261.

[50] Clare M Wilson, Kenneth D Cocker, Merrick J Moseley, Carl Paterson, Simon T Clay, William E Schulenburg, Monte D Mills, Anna L Ells, Kim H Parker, Graham E Quinn, et al. 2008. Computerized Analysis of Retinal Vessel Width and Tortuosity in Premature Infants. *Investigative Ophthalmology & Visual Science* 49, 8 (2008).

A ROP IMAGE FEATURE EXTRACTION

We follow the feature extraction method from Ataer-Cansızoğlu [4]. which consists of three steps: segmentation, tracing, and feature extraction. In our dataset, ROP images have a fixed size of 640×480.

A.1 Segmentation

We preprocess an ROP image as follows: first, we smooth the green channel of the original image with anisotropic diffusion, to get a gray level image that emphasizes the vessels and suppress the noise in the background. Then we apply unsharpening filter and enhance the image based on global mean and variance of the image. Finally, the image is Frangi filtered after an adaptive histogram equalization.

Let $f(\boldsymbol{p}) : \mathbb{R}^2 \to \mathbb{R}$ be a membership function indicating the probability that a pixel $\boldsymbol{p} \in \mathbb{R}^2$ is on a vessel. In our case, we consider our membership function as:

$$f(\boldsymbol{p}) = \sum_{i \in \Omega} w(\boldsymbol{p}_i) G_{\Sigma_i}(\boldsymbol{p} - \boldsymbol{p}_i), \tag{10}$$

where Ω is the set of pixel indices of an image, and Σ_i is the covariance of the Gaussian kernel $G_{\Sigma_i}(\boldsymbol{p}_i) = C_{\Sigma_i} \exp(-\frac{1}{2}) \boldsymbol{p}_i^T \Sigma_i^{-1} \boldsymbol{p}_i$ and C_{Σ_i} is the normalization parameter. The weight of each pixel $\boldsymbol{p}_i$ is given by the preprocessed image. We define $\boldsymbol{H}(\boldsymbol{p})$ as Hessian under function f. Let $\{(\lambda_1(\boldsymbol{p}), \boldsymbol{q}_1(\boldsymbol{p})), (\lambda_2(\boldsymbol{p}), \boldsymbol{q}_2(\boldsymbol{p}))\}$ be the eigenvalue-eigenvector pairs of $\boldsymbol{H}(\boldsymbol{p})$, which are sorted such that $|\lambda_1(\boldsymbol{p})| \geq |\lambda_2(\boldsymbol{p})|$. If a point $\boldsymbol{p}$ is around (away from) a vessel, we have $\lambda_1(\boldsymbol{p}) > 0$ ($\lambda_1(\boldsymbol{p}) < 0$). We define a threshold function $J(\boldsymbol{p}) = 1$ if $\lambda_1(\boldsymbol{p}) < 0$ and $J(\boldsymbol{p}) = 1$, o.w.

To remove the spurious areas, we postprocess the binary image produced by function $J(\boldsymbol{p})$. We discard pixels that is less than $\gamma \times \max_{i \in \Omega} f(\boldsymbol{p}_i)$ and we set $\gamma = 0.0015$. We also remove the isolated small areas less than 100 pixels.

A.2 Tracing

After segmentation, we obtain a binary image that indicates whether a pixel is on the vessel. Tracing contains two steps: finding sample points on vessel center-lines, and exploring the tree structure of vessels.

Sample Points on Vessel Center-lines. We use a principal curve (PC)-based method, to find sample points vessel center-lines. A point is on the principal curve, e.g., vessel center-line, if the local gradient is in the same direction with one of the eigenvectors of the Hessain and all eigenvalues, except the one that local gradient corresponds to, are negative. Let $g(\boldsymbol{p})$ be the gradient of the membership function $f(\boldsymbol{p})$ and use the same definitions of Hessian and eigenvalue-eigenvectors pairs in Appendix A.1. We denote function $\alpha(\boldsymbol{p})$ as follows:

$$\alpha(\boldsymbol{p}) = -\frac{\boldsymbol{g}^T(\boldsymbol{p}) \boldsymbol{q}_2(\boldsymbol{p}) \boldsymbol{q}_2^T(\boldsymbol{p}) \boldsymbol{g}(\boldsymbol{p})}{\lambda_2 \left\| (\boldsymbol{p}) \boldsymbol{H}(\boldsymbol{p}) \boldsymbol{g}(\boldsymbol{p}) \right\| \left\| \boldsymbol{g}(\boldsymbol{p}) \right\|}. \tag{11}$$

To find a point on a vessel centerline, we use an iterative algorithm. Starting from a point $\boldsymbol{p}^{(0)}$, we update the point $\boldsymbol{p}^{(t)}$ at iteration t by:

$$\boldsymbol{p}^{(t+1)} = \boldsymbol{p}^t + \alpha_t \boldsymbol{q}_1(\boldsymbol{p}^{(t)}) \boldsymbol{q}_1^T(\boldsymbol{p}^{(t)}) \boldsymbol{g}(\boldsymbol{p}^{(t)}), \tag{12}$$

where $\alpha_t = \frac{1}{|\lambda_1(\boldsymbol{p}^{(t)})|}$ is the step size. The algorithm stops when $\alpha(\boldsymbol{p}^{(t)}) < 0.1$ and $\boldsymbol{p}^{(t)}$ is the point on a vessel center-line.

For an ROP image, we find sample points on the vessel center-lines as follows: first, we interpolate the segmented image by Eq. (10). Note that the $w(\boldsymbol{p}_i)$ in the segmented image is binary. We use all the pixels which are not zeros in the segmented image, as starting points. Then we iterate these points to sample points on vessel center-lines via Eq. (12). We add the optic disc center (OD) to the sample points.

Tree Structure of Vessels. We denote $\mathcal{V}$ as a set of the sample points on vessel center-lines, including OD. Let $\mathcal{V}$ be the nodes of an undirected weighted graph $\mathcal{G}(\mathcal{V}, \mathcal{W})$, and $\mathcal{W} = \{w_{(\boldsymbol{i},\boldsymbol{j})}\}_{\boldsymbol{i},\boldsymbol{j} \in \mathcal{V}}$, where $w_{(\boldsymbol{i},\boldsymbol{j})}$ is the Euclidean distance between two sample points $\boldsymbol{i}$ and $\boldsymbol{j}$ on vessel center-lines. Then we use minimum spanning tree algorithm to obtain a graph $\mathcal{G}'(\mathcal{V}, \mathcal{W}')$, which keeps all the nodes in $\mathcal{G}$ and attains the minimum total distances in $\mathcal{W}'$. Finally, we obtain a tree structure by setting OD as the root node.

A.3 Feature Extraction

A tree structure of each ROP image is obtained after tracing. This structure consists of several vessel segments, which are curves between two junction points, or between a junction point and an end point. Moreover, after tracing, each vessel segment is a set of ordered points and any two neighbor points in the set produce the vessel centerline. However, these points are not necessarily sampled in equal distance over the curve. Thus, we fit cubic splines and sample equally distance points from the splines.

Let $\boldsymbol{c}(t) : [a, b] \in \mathbb{R} \to \mathbb{R}^2$ be the parametrization of a vessel segment x where $\boldsymbol{c}(a)$ and $\boldsymbol{c}(b)$ are two end points. Curve length $L_c(x)$ is computed as:

$$L_c(x) = \int_a^b \left\| \frac{\partial \boldsymbol{c}(t)}{\partial t} \right\| \partial t, \tag{13}$$

and chord length $L_x(x)$ is computed as:

$$L_x(x) = \left\| \boldsymbol{c}(a) - \boldsymbol{c}(b) \right\|, \tag{14}$$

which is the distance bewteen two end points of the segment x. We define curvature as the rate of changing in velocity vector with respect to curve length and is computed as:

$$\kappa(s) = \left\| \frac{\partial \boldsymbol{v}(t)}{\partial t} \frac{\partial t}{\partial s} \right\|, \tag{15}$$

where $\boldsymbol{v}(t)$ is the unit tangential vector of $\boldsymbol{c}(t)$ and s is the curve length parameter.

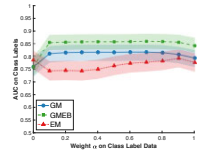
We extract two types of features: segment-based features and point-based features.

Segment-based Features. We extract several features based on segment tortuosity. Cumulative tortuosity index (cti) is the ratio between curve length and chord length. To measure tortuosity,we also consider two variants of curvature: integrated curvature (ic), and integrated squared curvature (isc). By dividing ic and isc with chord length or curve length, we obtain 4 more variants of curvatures: ic normalized by chord length (icLx), ic normalized by curve length (icLc), isc normalized by chord length (iscLx), and isc normalized by curve length (iscLc). To measure dilation of each segment, we compute average segment diameter (asd) as the total number of pixels from the segment divided by the curve length. Distance to disc center (ddc) is the distance between the end point of each segment and the disc center $\rho \in \mathbb{R}^2$.

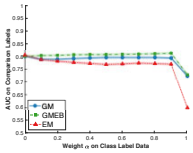
Table 8: Extracted Features [4]

Feature	Formula
Cumulative tortuosity index	$cti(x) = L_c(x)/L_x(x)$
Integrated curvature	$\text{ic}(x) = \int_a^b \lvert\kappa(s)\rvert ds$
Integrated squared curvature	$\text{isc}(x) = \int_a^b \kappa(s)^2 ds$
IC normalized by Chord Length	$\text{icLx}(x) = \text{ic}(x)/L_x(x)$
IC normalized by Curve Length	$\text{icLc}(x) = \text{ic}(x)/L_c(x)$
ISC normalized by Chord Length	$\text{iscLx}(x) = \text{isc}(x)/L_x(x)$
ISC normalized by Curve Length	$\text{iscLc}(x) = \text{isc}(x)/L_c(x)$
Average Segment Diameter	$\text{asd}(x) = \#\text{pixels}/L_c(x)$
Distance to Disc Center	$\text{ddc}(x) = \lVert c(b) - \rho\rVert$
Curvature	$\kappa(s)$
Norm of Acceleration vector	$\text{acc}(t) = \left\lVert \frac{\partial^2 c(t)}{\partial t^2} \right\rVert$
Point Diameter	$\text{pd}(x) =$ width of the vessel

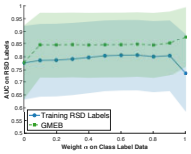
In each formula, vessel segment x is parameterized by $c(t)$ in which a and b are the starting and ending points respectively. $\kappa(s)$ is the curvature value at point s and ρ denotes the disk center. $L_x(x)$ denotes the chord length and $L_c(x)$ represents the curve length.



(a) Plus vs Not Plus of Automatically Segmented Result on Diagnostic Labels



(b) Plus vs Not Plus of Automatically Segmented Result on Comparison Labels



(c) Plus vs Not Plus of Automatically Segmented Result on RSD Labels

Figure 10: Comparison between expert representation models on automatically segmented features under SVM penalties. The shaded area indicates 95% confidence interval. Note that GMEB *does not use* expert specific biases in predicting comparison and RSD labels. EM uses expert specific scores. We observed several important trends. First, GMEB outperforms GM and EM. Second, GMEB attains an optimal AUC at $\alpha = 0.9$. Then, EM performs worse than GMEB and GM. Finally, GMEB model trained with diagnostic label outperforms the model trained with RSD labels even in predicting RSD labels.

Point-based Features. We compute pointer diameter (pd) and norm of acceleration vector (acc). Point diameter of a center-line point is computed by drawing an orthogonal line w.r.t. the center-line and finding its intersection with the vessel boundary. The norm of acceleration vector is computed by $\text{acc}(t) = \left\lVert \frac{\partial^2 c(t)}{\partial t^2} \right\rVert$.

We summarize all twelve features extracted in Table 8. Each feature in Table 8 forms a pool of numbers, measured on either segments or points. We compute two kinds of statistics of the pool: traditional statistics and Gaussian mixture statistics. The traditional statistics includes two smallest values, two largest values, mean, median, second central moment and third central moment of the pool. We can also consider the pool as samples from a two component Gaussian Mixture Model (GMM), based on the observation that an ROP image might contain both diseased and healthy vessels. We first split the pool into two groups by kmeans and then fit two Gaussians on the two groups. For a pool of numbers generated by a feature in Table 8, the probability of a extracted number $f \in \mathbb{R}$ being from this pool is: $p(f) = \pi\mathcal{N}(f; \mu_1, \sigma_1) + (1-\pi)\mathcal{N}(f; \mu_2, \sigma_2)$, where π is the weight of the first group. For $i \in \{1, 2\}$, μ_i and σ_i are mean and variance of i-th group, respectively, and we assume that $\mu_1 < \mu_2$. Thus each feature in Table 8 of an ROP image is represented by the five parameters of GMM: μ_1, μ_2, σ_1, σ_2, and π.

Each feature has 13 statistics including 8 traditional statistics and 5 GMM parameters. By concatenating all the statistics of 12 features, each ROP image is represented as a 156 dimensional feature vector.

B ELO ALGORITHM

We use Elo algorithm [20] to generate a ranking of the original 100-images dataset. The Elo algorithm proceeds as follows: first, the initial ranking scores for all 100 images are set at 2200. In each comparison event between images i and j, we denote the current ranking score of images i and j as s_i and s_j, respectively. Let $Q_i = 10^{s_i/400}$ and $Q_j = 10^{s_j/400}$. Then the expected scores of images i and j are defined as $E_i = Q_i/(Q_i + Q_j)$ and $E_j = Q_j/(Q_i + Q_j)$, respectively. If image $i(j)$ is the winner, e.g., the worse one in ROP, let $r_i = 1$ and $r_j = 0$ ($r_j = 1$ and $r_i = 0$). For $l \in \{i, j\}$, the updated ranking score is defined as:

$$s'_l = s_l + K(r_l - E_l), \tag{16}$$

where $K = 16$. After calculating the scores for all comparison events, all 100 images obtain their ranking scores and we sort the ranking scores to get the rank of 100 images.

C ADDITIONAL DATASET DESCRIPTION

The Filter Aesthetic Comparison (FAC) dataset [45] has 1280 unfiltered images in 8 categories. Twenty two different image filters are applied to each image and filtered images are labelled by Amazon Mechanical Turk users. For two filtered images i and j, the comparison label $y_{(i,j)} = +1$ if image i has better quality than image j, and $y_{(i,j)} = -1$ o.w. Each filtered image appears in three comparison pairs. The class labels are generated as follows: for each filtered image pair (i, j), the images i and j receive scores $+1, -1$, respectively, if $y_{(i,j)} = +1$. Then each image has a score in $[-3, +3]$. A filtered image i that receives a score $+3(-3)$ is labelled as $y_i = +1$ ($y_i = -1$); images that do not receive $+3$ or -3 are discarded. Hence, the class label $y_i = +1$ indicates that image i has high quality and $y_i = -1$ that it does not. As the comparison labels only exist in the same category, we use one of the categories with $N = 3520$ filtered images. As a result, there are 3520 binary class labels and 4964 comparison labels. As the class labels are generated from multiple users and to let comparison labels be consistent with class labels, we discard the multiple user information and train FAC dataset via GM model in Eq. (7).

The Netflix dataset contains 17770 movies rated by multiple users. We select 53 users who rate more than 1000 movies and select 201 movies which receive at least 350 rate scores. Each movie has a 30-dimensional feature vector obtained by matrix factorization [33] over the entire dataset. We generate class and comparison labels as follows: if the rate score is higher (lower) than the user's average rate score, we generate the class label as $+1(-1)$. For each user e, if the scores between two movies i and j are different, we generate the comparison label as $y^e_{(i,j)} = +1$ if score i is higher than score j, and we assign $y^e_{(i,j)} = -1$ o.w. All in all, this dataset contains $8,411$ class labels and $442,208$ comparison labels.