Kaunas University of Technology

Faculty of Mathematics and Natural Sciences

Gaussian Processes for Uncertainty Based Diagnostics of Eye Fundus Pathologies

Masters's Final Degree Project

Vilius Dzidolikas

Project author

Doc. dr. Tomas Iešmantas

Supervisor

Kaunas, 2024

Kaunas University of Technology

Faculty of Mathematics and Natural Sciences

Gaussian Processes for Uncertainty Based Diagnostics of Eye Fundus Pathologies

Masters's Final Degree Project

Applied Mathematics (6211AX006)

Vilius Dzidolikas

Project author

Doc. dr. Tomas Iešmantas

Supervisor

Doc. dr. Mantas Landauskas

Reviewer

Kaunas, 2024

Kaunas University of Technology

Faculty of Mathematics and Natural Sciences

Vilius Dzidolikas

Gaussian Processes for Uncertainty Based Diagnostics of Eye Fundus Pathologies

Declaration of Academic Integrity

I confirm the following:

1. I have prepared the final degree project independently and honestly without any violations of the copyrights or other rights of others, following the provisions of the Law on Copyrights and Related Rights of the Republic of Lithuania, the Regulations on the Management and Transfer of Intellectual Property of Kaunas University of Technology (hereinafter – University) and the ethical requirements stipulated by the Code of Academic Ethics of the University;

2. All the data and research results provided in the final degree project are correct and obtained legally; none of the parts of this project are plagiarised from any printed or electronic sources; all the quotations and references provided in the text of the final degree project are indicated in the list of references;

3. I have not paid anyone any monetary funds for the final degree project or the parts thereof unless required by the law;

4. I understand that in the case of any discovery of the fact of dishonesty or violation of any rights of others, the academic penalties will be imposed on me under the procedure applied at the University; I will be expelled from the University and my final degree project can be submitted to the Office of the Ombudsperson for Academic Ethics and Procedures in the examination of a possible violation of academic ethics.

Vilius Dzidolikas

Confirmed electronically

Dzidolikas, Vilius. Gaussian Processes for Uncertainty Based Diagnostics of Eye Fundus Pathologies. Master's Final Degree Project / supervisor doc. dr. Tomas Iešmantas; Faculty of Mathematics and Natural Sciences, Kaunas University of Technology.

Study field and area (study field group): Mathematics, Applied mathematics.

Keywords: Gaussian process, convolutional neural network, eye pathology diagnostics, multi-label classification, forecast reliability.
Kaunas, 2024. 46 p.

Summary

Deep neural networks are a group of mathematical methods widely applicable in practice. The field of medicine is not an exception, especially medical diagnostics, where medical digital images are used for pathology diagnosis. However, this group of methods has a key flaw, which stops its wider adoptability in practice. That is being incapable of modelling the uncertainty of the forecast, which reduces the reliability of the method in practical settings.

Gaussian processes are non-parametrical models, which are capable of modelling complex relationships and have the capability to evaluate the forecast uncertainties. In this paper we attempt to utilize these properties of Gaussian processes in modelling deep features, which are generated by convolutional neural networks, and modelling probability uncertainties, which arise from the stochastic nature of the data.

The research object of this paper is digital eye fundus images, which are used for diagnostic of various pathologies. We demonstrate that the transition from the convolutional neural network to the Gaussian process does not decrease the accuracy of the diagnosis and allows us to evaluate the reliability of the diagnosis by using the uncertainty measures to identify difficult cases. These cases can then be referred to a specialist, rather than proceeding with automated diagnostics.

Dzidolikas, Vilius. Atsitiktiniai Gauso procesai neapibrėžtumu grįstai akies dugno patologijų diagnostikai. Magistro baigiamasis projektas / vadovas doc. dr. Tomas Iešmantas; Kauno technologijos universitetas, Matematikos ir gamtos mokslų fakultetas.

Studijų kryptis ir sritis (studijų krypčių grupė): Matematikos mokslai, taikomoji matematika.

Reikšminiai žodžiai: Gauso procesas, sąsūkų neuroninis tinklas, akies patologijų diagnostika, kelių etikečių klasifikavimas, prognozių patikimumas.

Kaunas, 2024. 46 p.

Santrauka

Gilieji neuroniniai tinklai yra plačiai praktikoje taikoma matematinių metodų grupė. Ne išimtis ir medicina – ypatingai diagnostikos sritis, kuomet naudojami medicininiai skaitmeniniai vaizdai įvairioms patologijoms diagnozuoti. Tačiau ši metodų grupė turi vieną esminį trūkumą, kuris stabdo platesnį panaudojamumą medicinos praktikoje. Tai negebėjimas tinkamai modeliuoti prognozės neapibrėžtumo, o tai mažina metodo patikimumą praktiniuose taikymuose.

Gauso atsitiktinių procesų grupė yra neparametrinis modelis, įgalinantis modeliuoti sudėtingus sąryšius ir tuo pačiu suteikiantis galimybę apskaičiuoti prognozių neapibrėžtumus. Šiame darbe siekiama išnaudoti šias Gauso atsitiktinių procesų savybes modeliuojant giliuosius požymius, generuojamus sąsūkų neuroninių tinklų, ir tikimybių neapibrėžtumus, kylančius dėl stochastinės duomenų prigimties.

Tyrimo objektas yra akies dugno skaitmeniniai vaizdai, iš kurių siekiama diagnozuoti įvairias patologijas. Tyrimu pademonstruojama, kad perėjimas nuo giliojo neuroninio tinklo į Gauso atsitiktinius procesus ne tik, kad nesumažina diagnozės tikslumo, bet padeda įvertinti diagnozės patikimumą ta prasme, kad gaunami prognozės neapibrėžtumo įverčiai gali būti panaudojami identifikuoti sunkiai diagnozuojamus atvejus ir juos referuojant gydytojui, vietoje to, kad vykdyti automatinę diagnostiką.

Table of contents

List of figures

List of abbreviations

ARD – automatic relevance determination

CNN – convolutional neural network

CRS – Chorioretinitis

CSR – central serous retinopathy

DGP – deep Gaussian process

DN – drusen

DNN – deep neural network

DR – diabetic retinopathy

ELBO – evidence lower bound

FLOPS – floating-point operations per second

GP – Gaussian process

GPDNN – Gaussian process hybrid deep neural networks

IDRiD – Indian Diabetic Retinopathy Image Dataset

KL – Kullback-Leibler divergence

MH – media haze

MIML – multi-instance multi-label

NN – neural network

PCA – principal component analysis

PSF – point spread function

RBF – radial basis function

RFMiD – Retinal Fundus Multi-disease Image Dataset

ROI – region of interest

SVGP – sparse variational Gaussian process

SVM – support vector machines

Introduction

Convolutional neural networks (CNN) for image classification have found their application in various fields. However, they are not ideal learners as their learned representation of knowledge of a selected domain is limited, and errors, even as few and far between, are unavoidable. In certain fields, for example medicine, these errors can carry serious consequences. Therefore, the modelled results require expert-based validation, for the models to have a wider adoption in medicine.

This thesis demonstrates the application of Gaussian processes (GP) for CNN deep feature modelling. The use of GPs allows estimation of forecasted probability uncertainty, which allows to evaluate the confidence of the model's performed forecast. One of the possible utilizations of probability uncertainty is to use it as a secondary tool to identify cases for which a forecast result cannot be trusted, and then to refer the case to a medical professional. This would help to reduce the workload of already strained medical professionals without compromising the accuracy of diagnostics. To achieve this, the following aim and tasks were formulated:

The aim is to create a hybrid method based on deep CNN architecture and GP to enable efficient visual feature learning and uncertainty estimation for eye pathology diagnostics.

Tasks:

1. Review of literature of Gaussian process applications to the medical domain, focusing on eye pathology diagnostics;
2. Selection of a convolutional neural network architecture for learning deep visual features within the context of multi-label eye fundus pathology classification task;
3. Gaussian process kernel design for multi-label classification and uncertainty estimation of eye pathologies using variational formulation of multi-output GPs and minimization of the evidence lower bound (ELBO) function;
4. Investigation of possible strategies for using uncertainty estimates of eye pathology probabilities within the medical decision-making process.

1. Literature analysis

1.1. Deep learning for eye pathology diagnostics

As deep learning adoption has risen over the years, research into medical applications has too. One of the most well-researched fields of application in medicine would be ophthalmology. The application of CNNs to medical eye image data has been researched for numerous eye pathologies, e.g., diabetic retinopathy (DR), glaucoma, age-related macular degeneration, and retinopathy of prematurity [1]:

- [2] has used an Inception-V3 ensemble for referable DR diagnostics, achieving 0.99 AUC with two datasets;
- [3] has used a VGG-19 ensemble to detect referable DR, glaucoma, and age-related macular degeneration with stellar performance over 11 datasets;
- [4] has tested numerous CNN architectures for age-related macular degeneration diagnostic;
- [5] has extended the framework past a CNN with an explainable deep learning solution for DR, providing a heatmap overlay for the input image with global average pooling class activation maps (Fig. 1), providing insights regarding which regions of the image have made the largest effect for the forecast. Furthermore, the classifier is expanded past the CNN to gradient boosted random trees, which take the deep features, extracted from the CNN, along with some extra metadata as input, to elevate the performance of the classifier.
- An automatic DR detection system was proposed in [6], which performs binary detection with an addition of identifying potential signs of it in the image. The system employs a large ResNet architecture, pretrained with the COCO dataset.

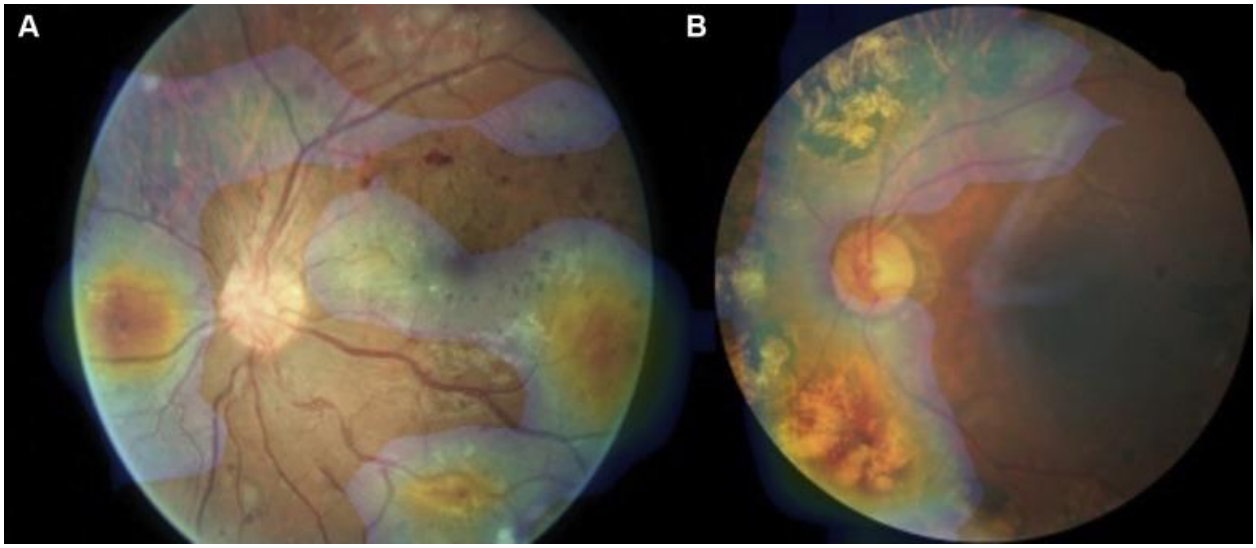


Fig. 1. Class activation maps on eye fundus images, taken from [5]

Even though there are numerous proposed automated diagnostics solutions for eye pathologies, most of them do not have interpretable results. Out of aforementioned papers, [5, 6] propose a framework that allows a medical practitioner to validate the results provided by the system. However, one main drawback of this approach is that, ideally, each case would still require a secondary review from a human, as there is no measure of confidence provided for the obtained forecasts, apart from the forecasted probability itself. For this reason, the field of deep learning has researched a way to estimate the uncertainty of model results.

1.2. Uncertainty estimation for medical image classification

For a framework using a CNN or any sort of variation of it, one of the most common uncertainty estimation methods would be Monte Carlo dropout. The use of a dropout layer allows to introduce variability in the model's predictions by "disabling" neurons with a predefined probability and obtain a predictive distribution. Another alternative for uncertainty estimation would be conformal prediction, which is applicable to a wider range of models, as it is a statistical data-based method. Conformal prediction is a statistical evaluation of the obtained forecasts, as conformal classifiers return the forecast's p-values by ranking based on a decided nonconformity measure by performing case tests against samples of the training set [7, 8]. Apart from these methods, it is also possible to quantify uncertainty via simple model ensembling or test-time data augmentation [8]. The latter utilizes random image augmentations to create variability in a single sample to obtain numerous different probability estimates.

Applications of Bayesian neural networks are also among the methods for obtaining uncertainty estimates, as it replaces the single fixed value of a network weight to a distribution. This theoretically draws the model close to an infinite ensemble of neural networks (NN), which allow obtaining a distribution of predicted probabilities. However, one of the main drawbacks of a Bayesian NN is the extensive changes required to the whole framework for it to be applicable to image data, with the additional increase of computational cost [8].

More concrete examples of uncertainty estimation in eye pathology diagnostics using CNNs would be:

- [9] grading DR severity on retinal images and quantifying prediction uncertainty by calculating Cohen's κ value for model predictions at threshold levels of uncertainty, which are calculated by variance in the grade probability [7];
- [10] diagnosing DR by eye fundus images and quantifying prediction uncertainty by calculating variance of predicted probabilities obtained from test-time data augmentations [7];
- [11] diagnosing DR by eye fundus images and quantifying prediction uncertainty by drawing Monte Carlo samples from the approximate predictive posterior and using its standard deviation to represent uncertainty [7];
- [12] classifying diabetic macular edema from optical cohere tomography images with the addition of recurrent NNs and quantifying prediction uncertainty by mean and standard deviation of probabilistic predictions yielded from ensemble of models [7];
- [13] detecting anomalies in retinal optical coherence tomography images with the addition of a Bayesian U-Net and quantifying prediction uncertainty by passing samples through the model multiple times, dropping weights each run, and calculating variance across the runs [7].

For all the aforementioned tactics to obtain quantification of prediction uncertainty (apart from Bayesian NNs) a single unifying property would be that they are surrogate methods, i.e., the probabilistic behaviour does not rise directly from the model which produces the predictions, but from a secondary modification, that introduces variability. In this regard, a Bayesian NN is the only NN framework that models uncertainty which rises from the model's probabilistic nature. However, as their application to image data is difficult and computational costs are high, alternative probabilistic frameworks are explored in this thesis.

1.3. Gaussian process and its derived frameworks' classification

The neuroimaging community has shown interest in multivariate pattern analysis and machine learning, in this case specifically to learn about the capabilities of GP to perform patient stratification from functional-connectivity brain patterns obtained at a resting state. The paper [14] from 2015 tests the GP logistic regression classifier with linear and non-linear covariance functions. The publication explores this classifier instead of support vector machines (SVM), since "being a probabilistic model, (it) provides a principled estimate of the probability of class membership. Class probability estimates are a measure of the confidence the model has in its predictions, such a confidence score may be extremely useful in the clinical setting." [14]

A rather well-researched field of machine learning application in medicine is DR detection and grading, popularized by the Kaggle competition [15]. Paper [16] explores the DR grading task with a GP twist. DR, a leading cause of blindness in developed countries, is a medical condition in which damage occurs to the retina due to diabetes mellitus. Most detection solutions rely on CNNs due to their edge in performance when working with image data, however the very best solutions do not rely solely on them, e.g., the Kaggle competition winner used random forests to weigh the CNN probabilities along some meta-data to improve the results [17].

[16] is not an exception to this trend, as here the authors describe the solution as a three-phase process of 1) preprocessing the image, 2) using the CNN as a feature extractor and 3) using the GP as the forecaster with an uncertainty estimation (Fig. 2). The dataset was filtered to exclude "very dark images where the circular (ROI) is not identified" [16], images were cropped to eliminate excess black margins, resized to 299×299 px, and augmented by applying horizontal reflection, brightness, saturation, hue, and contrast changes. Deep feature extraction was performed using an Inception-V3 model with pretrained weights of the ImageNet dataset [18] and fine-tuned using the Kaggle competition DR dataset, using binary cross-entropy loss, RMSprop optimizer and a decaying learning rate policy. The model architecture ends with a global average pooling layer, which results in a 2048 feature vector per image. An interesting choice was made to train the CNN model for a binary classification task of differentiating between non-referable and referable DR, even though the final task for the GP is scoring the severity of DR on a five-grade scale. Perhaps that strengthens the notion, that ultimately the function of the CNN is of a feature extractor, and so its weights should only be fine-tuned for the eye fundus images, rather than training it to be scorer itself, where the responsibility of doing so falls on the GP model. The GP is equipped with a radial basis function (RBF) kernel and, most notably, is set up not as a classification, but as a regression model for the grading from 0 to 5 of the DR scale – the output of this GP is a continuous number.

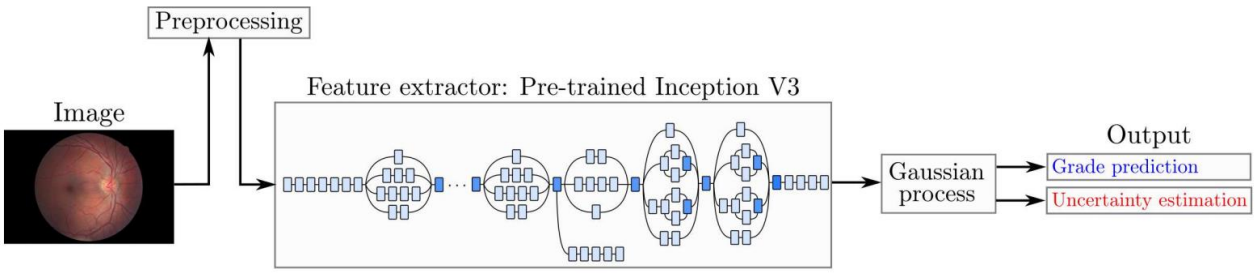


Fig. 2. The experiment model setup, taken from [16]

The paper reports results of considerable performance, one that surpasses of simply using the NN, however they are not comparable to any other contemporaries that use the same dataset because of their unorthodox further "binarization" of the GP results. Furthermore, for a paper that notes the GPs capability of uncertainty quantification, it is rather underutilized by focusing only on the result of the regression and disregarding the possibility to explore predictive uncertainty of the classifier.

The proposed hybrid model of [16] could be reinterpreted as just a GP model that uses extensively preprocessed data, and it so happens that for that instance a CNN was chosen to transform and reduce the dimensionality of the data, as opposed to principal component analysis (PCA) or another method. However, "true" hybrid frameworks have also been proposed, such as [19]. The notion of using a deep neural network (DNN) as a feature extraction tool and then leveraging these features with a GP or any other model is not groundbreaking. However, most of these hybrid solutions are set up as separate stages, whereas in [19] it is trained end-to-end as a single model with a consistent architecture. The structure their GP hybrid DNNs (GPDNN) use the libraries of TensorFlow [20] and GPflow [21], which itself is based on the former, therefore the implementation is rather fluid from a package, dependency, and algorithm cohesion standpoint, as the libraries allow to consistently back-propagate through all the model parts, Cholesky decomposition (for matrix inversion) of GP part included, by using training batches. The NN aspect of the model is covered by a CNN framework and the GP aspect of the model is executed by a sparse variational GP (SVGP) framework.

The GPDNNs of [19] are tested with classification tasks, exposure to adversarial examples and domain transfer to showcase their capabilities and robustness. Image classification results report a consistent performance increase over base CNNs or their modified counterparts, which have extra hidden units before the top layer, to give more hyperparameters to the networks and perform a fairer comparison. What is worth noting is that, again, the difference between performance decreases as more data becomes available to the classifiers. Another unexpected behavior is that when using a different image dataset and a larger CNN base, the DenseNet architecture, the hybrid network failed to start successfully fitting the data from random weight initializations, therefore a workaround of pretraining the base was required.

After exploring how the hybrid framework responds to non-targeted adversarial examples generated by the fast gradient sign method, it was found that their error rate increases much slower and on a lesser scale than CNN counterparts when adversarial perturbations are increased. Furthermore, for large perturbations, which translate to unknown regions in the data/feature space, the model holds reserved confidence when making predictions. For Carlini and Wagner L2 optimization attack method, "the hybrid model appears more robust, with a greater number of attack failures and larger perturbations needed for successful attacks" [19]. After testing the GPDNN behavior in a domain shift setting, it became apparent that they underperform against CNNs in terms of accuracy (although not extremely, with a 0.14 difference in the most difficult dataset change setting), however, as with adversarial examples, hybrid models begin to lean more towards the undecisive likelihoods, showing their domain shift awareness. Overall, this hybrid combination leverages the strong representational ability of CNNs and the domain-awareness and robustness of a GP resulting in a competent model with comparable performance and strong uncertainty estimation, which has great potential for applications in real life, such as in fields of medicine or autonomous driving.

The authors of [22] have developed an analogous approach of an end-to-end trainable solution of a CNN and GP and applied it to DR data with the most practical training and testing scenario for the

medical background of the task. The hybrid approach, developed by [23], uses a GP layer for the network top, which is operating on the CNN's deep feature space. The number of GPs can be chosen, each of which take a subset of the CNN's activations as input. The model's likelihood is computed by passing the outputs of the GPs through a final layer, resembling an additive operation over the GPs, and applying the softmax function. This approach also leverages stochastic variational inference and sparse approximation for GPs with the addition of newly derived efficient and scalable sampling scheme for the approximation of ELBO. As with the previous proposed hybrid approach, here the CNN and GP structure is also jointly optimized via backpropagation and minibatch training. This model, as [23] call it stochastic variational deep kernel learning, is reportedly an improvement over standard NNs and the "separated" CNN-GP hybrid approaches performance-wise. Although it is not a universal upgrade from a separated structure, as being conjoined to and trained by the algorithms of a CNN allows them to be prone to overfitting, whereas an isolated GP is not, but it can be mitigated by applying Monte-Carlo dropout or weight regularization to the NN.

[22] first separately pretrain the CNN, then, after freezing the weights, joined with the GP layer, and trained to fine tune the variational parameters. Afterwards the full hybrid model is trained end-to-end. All models use the Adam optimizer [24] with a plateau scheduler, which halves the learning rate after 10 epochs of no improvement, and an early stopping tactic which interrupts the fitting after 20 plateau epochs. Several deep kernel learning application variations are prepared – ones that only encompass the base description, modifications employing Monte-Carlo dropout, spectral norm for each convolution group and interchanging the original Swish activation function with its Lipschitz alternative, or collections of models as deep ensembles.

For in-distribution data, a union of two popular DR datasets are used – the Kaggle competition/ EyePACS dataset and the Indian Diabetic Retinopathy Image Dataset (IDRiD), both consisting of ophthalmoscopic eye fundus images annotated for 5 classes of DR severity. However, only EyePACS' public data split is used for training, and the private split is kept for testing. The training data oversamples minority classes and augments all images by random horizontal and vertical flipping, rotation, shifts along both axes and scaling. For near-out-of-distribution data, the Retinal Fundus Multi-disease Image Dataset (RFMiD) [25] is used, which, in addition to DR, extends the set to a multitude of other eye diseases. Finally, for out-of-distribution data a medical dataset of dermoscopic images is used, which comprises of somewhat similar features (Fig. 3). The in-distribution data is preprocessed by cropping to a square containing the retinal disc, and out-of-distribution data is padded to a square. All images are downsized to 224×224 px and the local color mean is removed using a Gaussian filter, which is a tactic borrowed from the Kaggle competition winner [17].

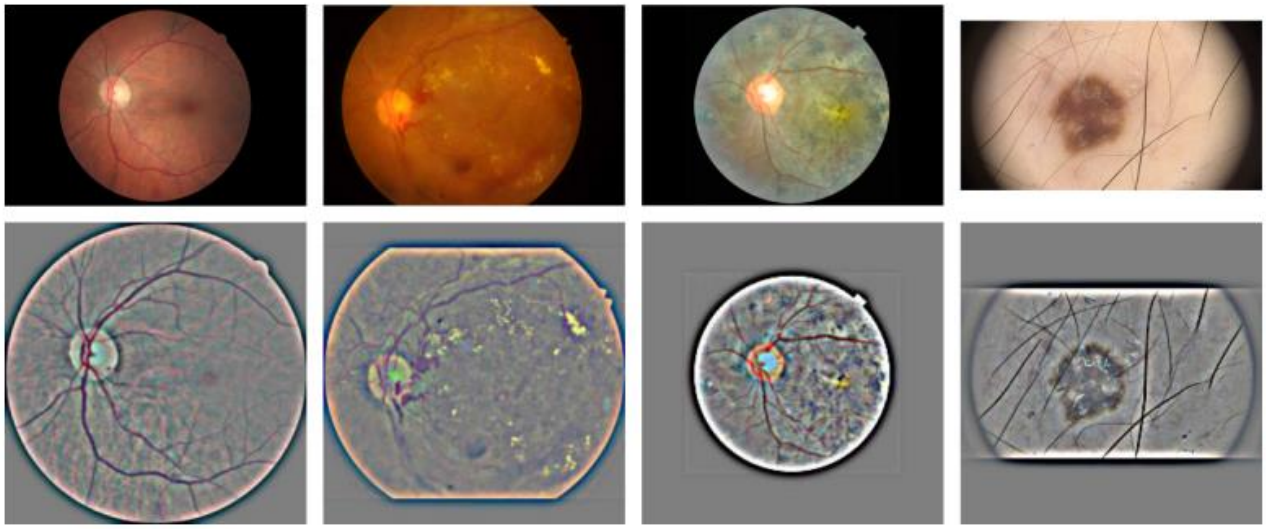


Fig. 3. EyePACS, IDRiD, RFMiD and dermoscopic (left to right) image dataset samples and their preprocessed counterparts, taken from [17]

The models are tested for DR severity grading and referable DR detection tasks. In-distribution performance is evaluated with AUC, accuracy, negative log-likelihood and Cohen's kappa. The uncertainty is evaluated with expected calibration error and area under the accuracy rejection curve. The first uncertainty measure summarizes the model's confidence with a perfectly calibrated one. The second measure imitates the real-life application of the model, where high-uncertainty samples are returned for a second opinion. The model's performance measures, when compared to the baseline's, are tested for significant differences with analysis of variance, t-tests and the Wilcoxon signed-rank test.

This extensive research culminates with results that support the superiority of deep kernel learning. The deep ensembles that use the further-modified models are of greater performance, however their uncertainty evaluations are worse than the base model's for the DR grading task. The base model struggles with near-out-of-distribution samples, it is understandable given the nature of the dataset and the representative nature of the CNN, but it is not all too disappointing, as for fully-out-of-distribution data, the model performs significantly better, although this comparison is far from a practical scenario given the data.

On the other hand, the deep ensemble that utilizes all possible modifications is reported to have an astonishing performance, being capable of modelling the uncertainty at a near-perfect level of performance for both out-of-distribution datasets. As for the comparison against the baseline CNNs, which have analogous modifications as the deep kernel learning counterparts, consistent improvements over performance of in-distribution data are reported for the hybrid models, apart from uncertainty evaluation, where the expected calibration error is better for the CNNs by a slight margin. Although the results are very much in favor of the hybrid method, it would have been interesting to see how it compares to a "simpler" alternative of keeping the CNN and GP separate.

Another alternative GP extension was presented in 2013 by [26] in the form of deep GP (DGP) models. The basic idea is to chain multivariate GP models by making one's output another's input. This idea can be analogously described in terms of DNNs, where the chained GP models are similiar

to network layers and the number of GPs in each model is akin to the number of nodes. It is worth noting that DGP models are a form of deep belief networks, as the nodes are not interconnected between the layers, only layers themselves are.

One of the main benefits of this Bayesian structure of the deep belief network is that the application is possible even when data is not available in large volumes, as opposed to the traditional requirement when using stochastic gradient descent optimization. The DGP defined in [26] also employs automatic relevance determination (ARD) covariance functions for further optimization of the model structure. For the deep multiple-output GPs, which can be applied to multi-label classification tasks, it is only required to expand the model horizontally (i.e., increase the number of nodes) to a desired degree and set them up to perform independently. The authors note that "this special case of our model makes the connection between our model's structure and NN architectures more obvious: the ARD parameters play a role similar to the weights of NNs, while the latent variables play the role of neurons which learn hierarchies of features" [26]. The proposed approach was tested by the authors of [26], with toy data and real-life datasets, e.g., human motion or digit image data, where DGP has proven its applicability with positive results.

1.4. Gaussian process multi-label classification

Up until this part of literature analysis we were only reviewing how the GP can be applied to binary of multi-class classification tasks (with the exception of DGP). However, the literature regarding the application of GP to multi-label classification tasks is rather sparse, possibly due to the difficulty of the task and the computational intensity that comes with trying to apply such an algorithm.

[27] introduced a TensorFlow-based GP model developed for multi-label classification of big data, which addressed computational challenges via the sparse variational framework. The authors employed a semi-parametric latent factor model which "allows to capture the correlation of multiple labels using a small set of shared latent GP functions" [27], extended to a multi-label case. However, the authors of [27] only demonstrated the application for text classification datasets.

In general, it is a multi-output model that uses a set number of latent GPs, in this case called "factors", to generate multiple outputs through a linear mapping. If we denote P as the number of latent GPs, $h_p^{(i)}(p = \overline{1,P})$ as a latent function drawn from a GP of zero-mean and kernel function of choice and evaluated at input of index i, then a utility score, which when transformed by a sigmoidal or Bernoulli likelihood returns the k-th ($k = \overline{1,K}$) binary label, is described as

$$f_k^{(i)} = \sum_{p=1}^{P} \phi_{kp} h_p^{(i)} + b_k,$$

where ϕ_{kp} is the weight for p-th factor for the k-th label and b_k is the bias value of the k-th label for linear mapping. The approach is then extended to the sparse variational framework by employing the inducing variables and stochastic optimization. This allows the GP to be fitted with datasets where there are over a million training samples or over 200 thousand labels. Testing was carried out over 6 text datasets, three small- and three large-scale, 2 kernels, linear and RBF (with extra hyperparameters for extremely high dimensional datasets) or a linear combination of the two, and by allowing for the inducing points to be optimized.

As the results report, this novel method stays on par with its state-of-the-art contemporaries and even manages to overtake them in terms of certain performance measures. However, this comes at the cost of extremely high training times, where other models manage to train in a fraction of the time. Furthermore, as the novelty of the GP is its predictive uncertainty, it is not showcased or described how it could be utilized.

Alternative approaches to multi-label tasks with sparse GPs were also explored by [28] in 2012. The authors have proposed a framework that used multiple named instance feature vectors to represent each sample in image data. The multi-instance multi-label (MIML) framework (Fig. 4) explores possible ambiguity in instance and label spaces [28]. Even though this framework has explainable prediction capabilities, which, as authors state, allow understanding of the causality of predicted labels, it still has two main challenges. The first one is the modelling of connections between instances and labels, as depending on the image, there might be either a single or multiple instances of a subject visible in different regions of the image. The second is how to properly exploit the correlations between different labels, as it is a powerful tool in improving the performance of the model, since it can encode certain data rules, and in sparse data settings it can compensate for the lack of training samples to a degree.

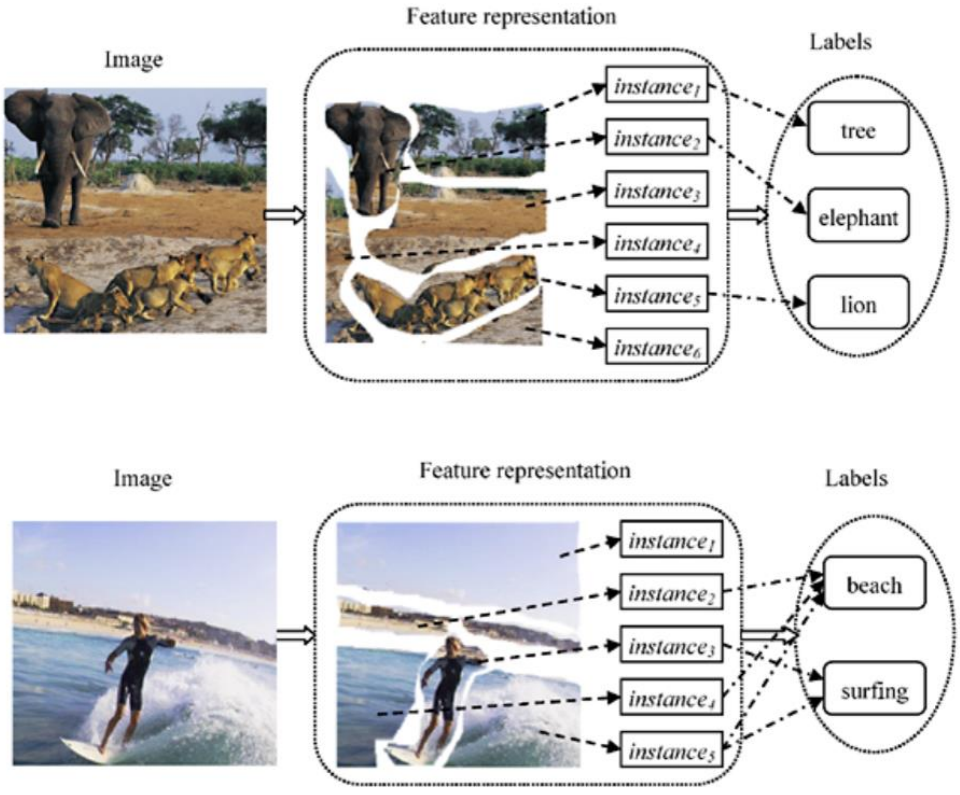


Fig. 4. Visual scheme of the MIML framework, taken from [28]

[28] describes the basic idea of this MIML learning framework as defining a latent GP prior in the instance space for each label. The correlations between labels are captured by the covariance matrix of the GP and the connections between instances and labels can be explored by various likelihood functions, as the kernel matrix is inferred by maximizing marginal likelihood.

The authors have tested their proposed algorithm with multi-label and multi-instance learning problems. The performance of the algorithm is compared to MIML RBF NN, MIML SVM and MIML k-nearest neighbors frameworks to have a relative measure of performance. The first multi-label classification problem is the classification of natural scene images belonging to five classes. Each sample is a bag consisting of several multi-dimensional instances, which are generated by the method described in [28]. Over 22% of images have multiple labels and the average number of labels per image is 1.24. The novel algorithm outperforms its competitors by precision, coverage, hamming loss, one-error and raking loss measures.

The same conclusion of outperforming its competitors is also reached with the second multi-label classification problem. This scenario covers text analysis and is realized with the Reuters-21578 dataset. Here, each document is represented as a bag of instances, obtained by overlapping rolling windows, and the seven largest categories are considered. After a custom dataset preprocessing, the final set contains 2000 documents and around 15% of them have multiple labels.

For the multi-instance setting, first the algorithm is applied to a text categorization problem with data derived from 20 Newsgroups corpus. Here, the framework's performance is compared to MI-Kernel and mi-Graph frameworks, which are consistently overtaken in terms of accuracy. The second setting of application is again an image categorization task, where 1000 or 2000 sample size image sets obtained from COREL are split into 10 or 20 categories of 100 images. The same superiority in performance is also obtained here, as the algorithm is compared to numerous methods, along with ones used for the text categorization task.

1.5. Motivation

As frameworks that allow computiationally- and time-efficient optimization of GPs for larger amounts of data were developed not too long ago, the amount of research of its applications is resurging. However, papers regarding its medical applications, especially for eye fundus pathologies are rather sparse. Furthermore, most of these papers focus on binary or multiclass DR detection, rather than a wider array of diagnostics of eye pathologies. As far as I am aware, this is the first work that looks into a multi-label eye pathology detection using GPs.

On the other hand, medical practitioners remain (rightfully) sceptical of automated diagnostics. Automated diagnosis systems, as well-performing as they may be, have a limited knowledge of the domain they operate on and rarely produce an estimate on how certain these systems are about the predictions they are producing. This can inadvertently lead to errors in diagnostics, which are more severe than human errors.

As for making a diagnosis based on image data, CNNs are incapable of modelling the uncertainty of their predictions on their own. This reduces the reliability of the method in practical medical settings. GPs have the inherent capability to model probability uncertainties. That, in turn, allows us to evaluate the reliability of the diagnosis by using uncertainty measures to identify difficult cases which could be referred to further analysis. As the field of medicine is exceptionally burdened with responsibility, it is physically and mentally demanding work. If we approach the idea of machine learning methods in medicine not as a replacement, but a complement to medical proffesionals, GPs might allow us to reach a compromise, by reducing the workload of an expert.

2. Data and methodology

2.1. Data

The experiments are conducted with the multi-label RFMiD dataset [25]. The total of 3200 images are split into 1920, 640 and 640 images for training, validation and testing respectively. These images contain visual characteristics for 46 categories – 45 eye pathologies and the healthy control group. The supplied categorization is used, which summarizes the images into 28 categories, as only pathologies that have more than 10 samples remain as an independent label. This results in a label stratification of 60±7% samples (depending on the label), 20±7% and 20±5% for training, validation and testing sets respectively [29]. The average label count for a sample with any pathology is ~2.3 and the average label count for the whole dataset is ~1.1.

In this thesis, only a subset of 4 labels is used, as the objective is not to create a quintessential eye pathology detector, but to demonstrate the benefits of expanding classifier framework beyond the CNN to a GP. The selected labels are:

1. Media haze (MH) – this label does not directly correspond to a certain pathology, as it is more of a descriptive label that indicates quality of the obtained image. However, as the data descriptor notes – "The opacity of media can be a hallmark for the presence of (various pathologies)" [29]. Furthermore, it is important to detect, whether an image is affected by any sort of artifact, to improve the decision-making process (Fig. 5a);
2. Drusen (DN) – yellow or white extracellular deposits, naturally occurring in the aged population, which are indicative of various pathologies (Fig. 5b);
3. Central serous retinopathy (CSR) – "a round serous detachement of the neurosensory retina from the underlying (retinal pigment epithelium)" [29], identified by the presence of subretinal fluid. Depending on the severity, vision loss is possible (Fig. 5c);
4. Chorioretinitis (CRS) – caused by infections and causes inflammation of parts of the eye (Fig. 5d).

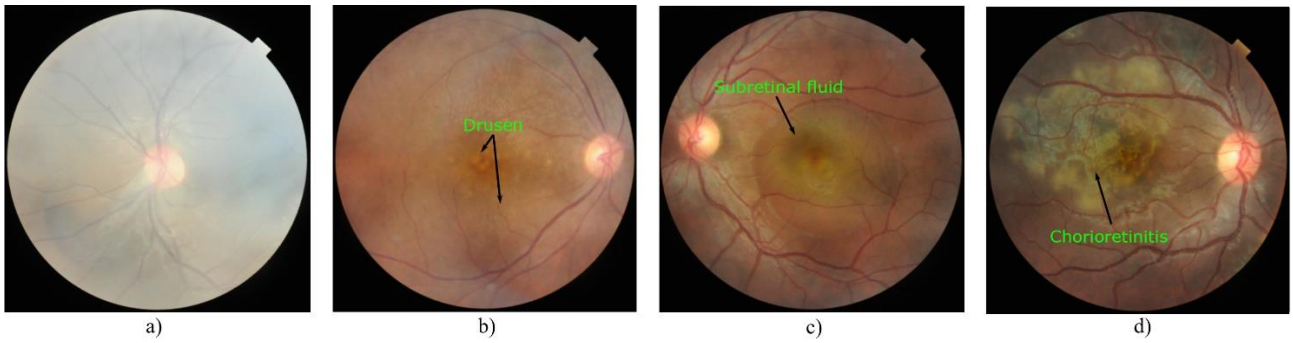


Fig. 5. Image samples of MH (a), DN (b), CSR (c), and CRS (d), along with their visual charateristics, taken from [29]

The selected labels have strongly varying sample sizes. Of all pathology label sample sizes, MH is in 2nd place, DN is in 5th place, whereas CSR and CRS are in 13th and 15th places respectively. In Fig. 6 details of the label distribution in each subset are shown, along with the amount of other pathology and healthy samples, which are encoded as 0 in our multi-hot label subset.

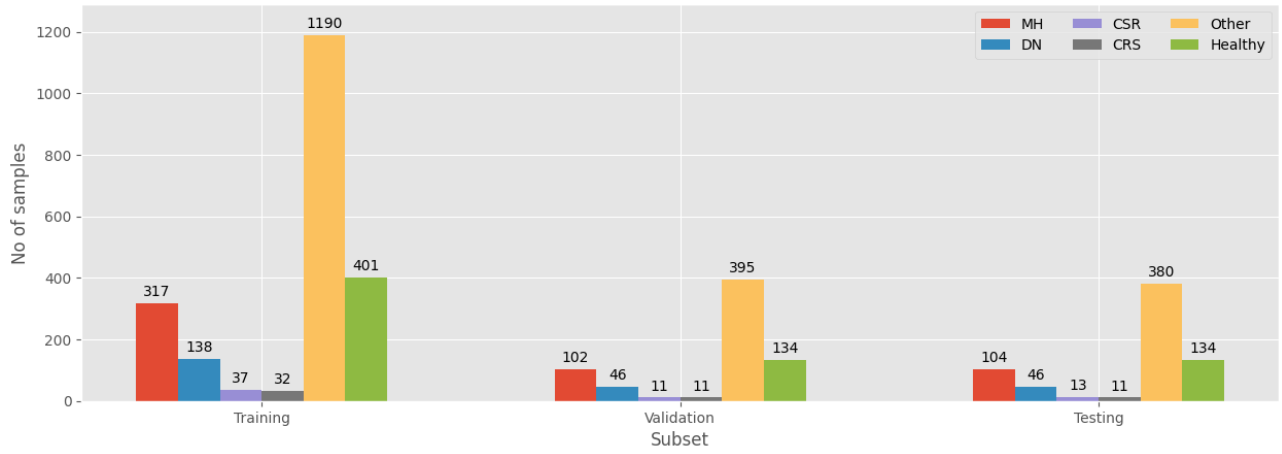
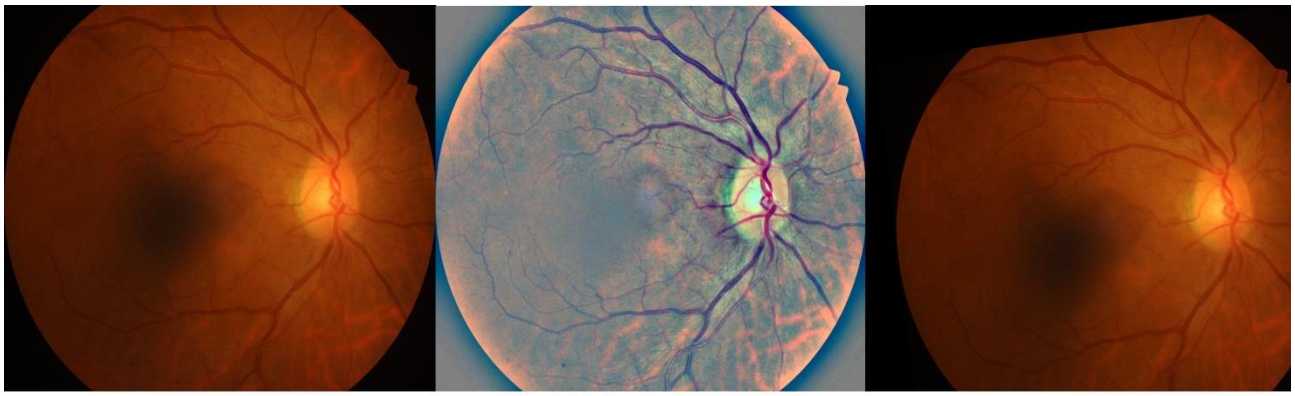


Fig. 6. Label distribution in each subset

The images in the dataset come in various resolutions. They are trimmed by removing the dark padding to bring the dimensions closer to a square format and remove uninformative data (Fig. 7a). For fitting the CNN, apart from the raw images, variations of preprocessed and augmented inputs are considered. All images are resized to 256×256 px.

The preprocessing tactic comes from the Kaggle DR competition [15] winner, used for similar data [17], where the color local average is subtracted from the image (we do not crop the boundaries resulting from this transformation, Fig. 7b). The idea of this approach is to reduce "variation between images due to differing lighting conditions, camera resolution, etc." [17]. Of course, as the feature shift is quite extreme from the original, the preprocessing is applied to all data subsets.

The augmented inputs are raw images for which random horizontal flips with probability $p = 0.5$, random rotations in the range of $[-90°, 90°]$ and shifts with a factor of 0.1, with $p = 1$, and elastic transformations with $p = 0.2$ are applied (Fig. 7c). The augmentations are applied to a 10x expanded testing set to introduce slight variability without transforming the data too drastically. This augmented and expanded training subset is created prior to fitting, to have stable comparability of results between the tested models.



a) b) c)

Fig. 7. Examples of the same image with its raw (a), preprocessed (b) and augmented (c) variants

2.2. Methods

2.2.1. Overall methodology and workflow

For the complete workflow, first a CNN model is trained to optimal performance using a single selected type of image input. After obtaining an optimal CNN model, we use it to extract deep visual features from the same image subsets. PCA is performed for dimensionality reduction and a selected number of first principal components. The GP model is optimized using the principal component inputs. Afterwards, we sample GP realizations from the posterior distribution and pass them through a logistic function to obtain a set of probabilities (Fig. 8).

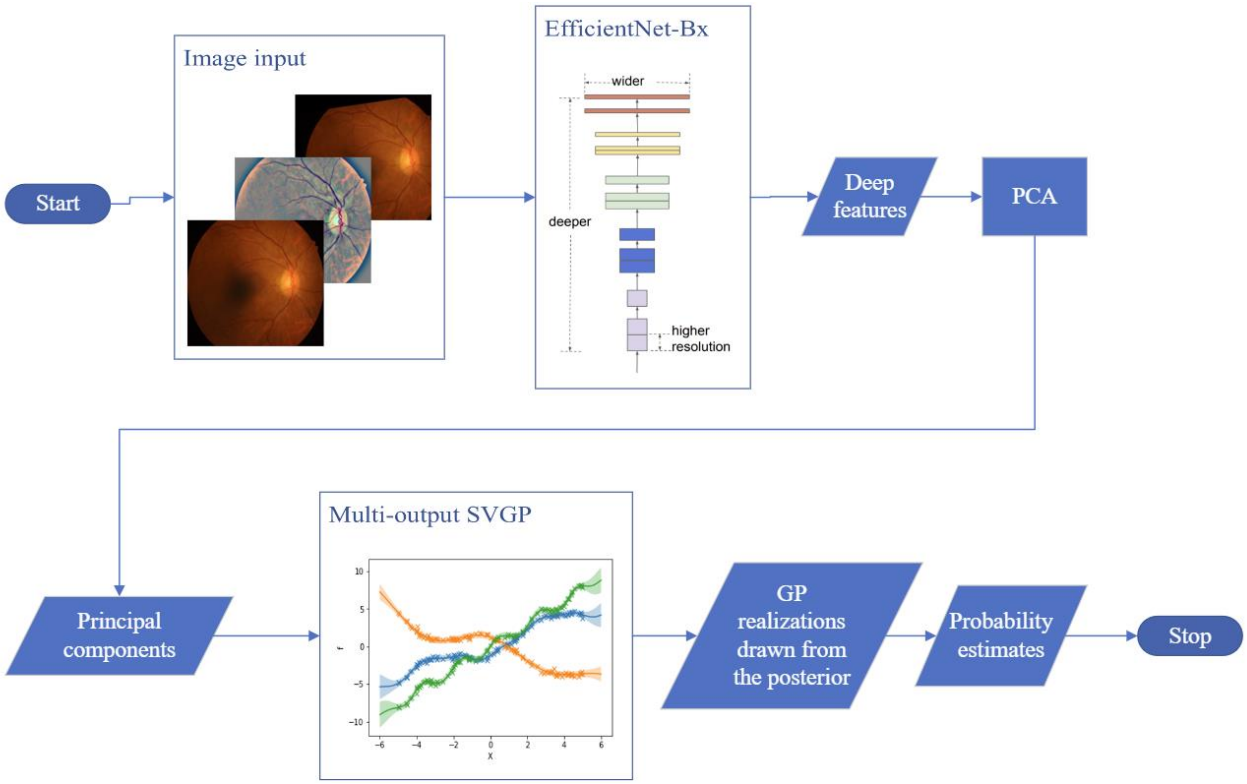


Fig. 8. The workflow of obtaining GP ensemble probability estimates from image data, EfficientNet image taken from [30], Multi-output SVGP image taken from [21]

2.2.2. Convolutional neural network

EfficientNet is a CNN family which was developed by "leveraging a multi-objective neural architecture search that optimizes both accuracy and floating-point operations per second (FLOPS)" [30] and utilizing an originally devised compound scaling method. The method uniformly scales network depth, width and input resolution dimensions using a compound coefficient, justified by intuition, that "if the input image is bigger, then the network needs more layers to increase the receptive field and more channels to capture more fine-grained patterns on the bigger image". The formalized compound scaling dictates that at the cost of approximately 2^{ϕ} times more computational resources, the network depth, width and image resolution should be respectively increased by α^{ϕ}, β^{ϕ}, γ^{ϕ} times. Here α, β, γ are constant coefficients, determined by the authors using a small grid search of the following system of equations with constraints:

$$\begin{cases} \text{depth: } d = \alpha^{\phi} \\ width: w = \beta^{\phi} \\ resolution: r = \gamma^{\phi} \\ s.t.\ \alpha \cdot \beta^2 \cdot \gamma^2 \approx 2 \\ \alpha \geq 1, \beta \geq 1, \gamma \geq 1 \end{cases}$$

The idea of the constraint is that the regular convolution operation is proportional to d, w^2, r^2.

Of course, the method of [19] is also backed up by empirical observations that the different scaling dimensions are not independent and best performance is achieved by scaling the dimensions together (Fig. 9).

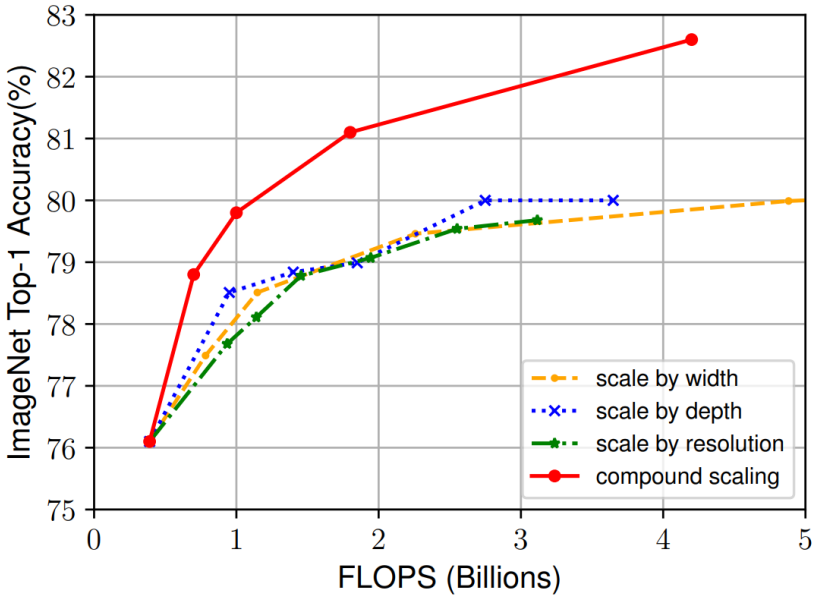


Fig. 9. Baseline EfficientNet architecture scaling method results, taken from [19]

For the root variant of EfficientNet, dubbed EfficientNet-B0, based on the inverted bottleneck residual blocks of MobileNetV2 and complemented with squeeze-and-excitation optimization, the found values for fixed $\phi = 1$ are $\alpha = 1.2$, $\beta = 1.1$, $\gamma = 1.15$. Scaled up variations of EfficientNet-B1 to EfficientNet-B7 are obtained by fixing the found values and increasing ϕ. Such architecture and scaling approach reaped multiple benefits – most notably, the performance to model size ratio is one that was not achieved before, as the EfficientNet models achieve state-of-the-art accuracy on various datasets with an order of magnitude fewer parameters (e.g., matching the ImageNet [18] top-1 accuracy of GPipe of 84.3% while being 8.4 times smaller and 6.1 times faster). Furthermore, in the words of the authors – "EfficientNets also transfer well and achieve state-of-the-art accuracy on 5 out of 8 widely used datasets, while reducing parameters by up to 21x than existing ConvNets". [19] Therefore, models of considerable performance can be developed in a much shorter time frame with the added benefit of speed-up from transfer learning.

2.2.3. Gaussian process

The GP model is constructed as follows. Consider a 1-dimensional case with noise-free observations. We have a training set $\mathcal{D} = \{(x_i, y_i) \mid i = 1, \dots, n\} = (X, Y)$, where x denotes covariates and y – the

dependent variable. A GP $f(x) \sim \mathcal{GP}(m(x), k(x,x'))$ is a collection of random variables, where any finite number of which have a joint Gaussian distribution. It is completely specified by its mean

$$m(x) = \mathbb{E}[f(x)]$$

and covariance (or otherwise called a kernel)

$$k(x,x') = \mathbb{E}[(f(x) - m(x))(f(x') - m(x'))]$$

functions. Vector x' belongs to the testing set, defined analogously as the training set. [31]

The GP definition may seem abstract, but GPs are simple objects – any function $f(x) = w^T \phi(x)$ with w drawn from a Gaussian distribution and $\phi(\cdot)$ being any vector of basis functions is a GP [32]. As the specification of the covariance function implies a distribution over functions, GP regression can be described as a process of drawing functions from the GP prior and conditioning them on observations to obtain a GP posterior. [31]

Assuming noise-free observations, if we denote

$$k(x, x_{1:n}) = [k(x,x_1) \quad \dots \quad k(x,x_n)],$$

$$k(x_{1:n}, x_{1:n}) = \begin{bmatrix} k(x_1,x_1) & \cdots & k(x_1,x_n) \\ \vdots & \ddots & \vdots \\ k(x_n,x_1) & \cdots & k(x_n,x_n) \end{bmatrix},$$

$$f(x_{1:n}) = [f(x_1) \quad \dots \quad f(x_n)],$$

$$\mu = k(x, x_{1:n}) k(x_{1:n}, x_{1:n})^{-1} f(x_{1:n}),$$

$$\sigma^2 = k(x,x) - k(x, x_{1:n}) k(x_{1:n}, x_{1:n})^{-1} k(x, x_{1:n}),$$

then we can compute the conditional distribution of $f(x)$ for any x given $f(x_1), \dots, f(x_n)$ as

$$f(x) \mid f(x_1), \dots, f(x_n) \sim \mathcal{N}(\mu, \sigma^2).$$

The terms of μ and σ^2 are respectively responsible for point prediction and uncertainty. As we are dealing with a Gaussian distribution, the 95% confidence interval of $f(x)$ would be $(\mu - 2\sigma; \mu + 2\sigma)$. Usually, the covariance function also has parameters θ which, for optimal results, must also be estimated. For that, marginal likelihood

$$p(y|x,\theta) = \int p(y|f,x,\theta)\, p(f|x,\theta) df$$

is utilized by maximizing the log marginal likelihood with respect to the parameters

$$\log p(y|x,\theta) = -\frac{1}{2} y\, k(x,x)\, y - \frac{1}{2} \log k(x,x) + c,$$

where c is a constant. [31]

The key property of GPs is that the obtained posterior becomes "distance-aware" of the data points – as the process approaches the data point, variance tends to decrease, and vice versa – as the process furthers from observed data, variance increases (Fig. 10).

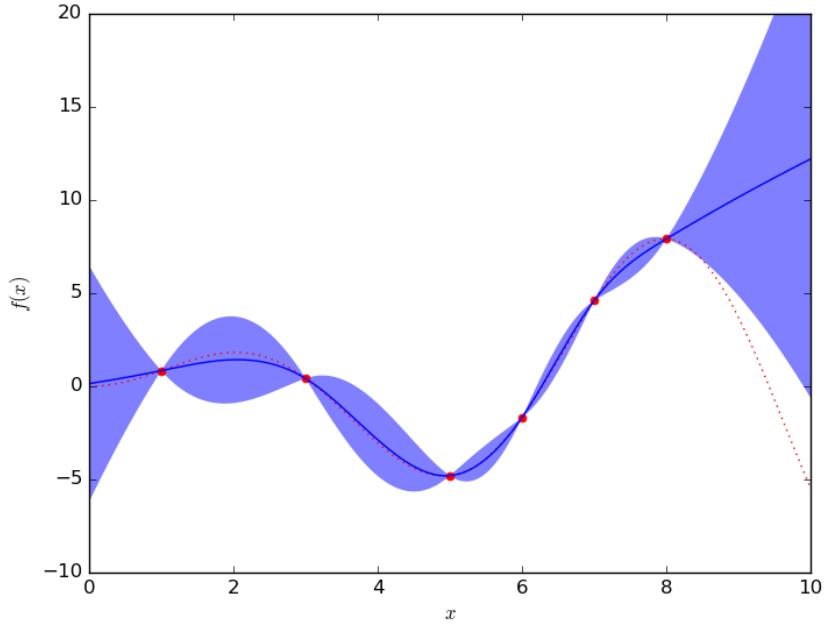


Fig. 10. GP posterior and its variance (blue), adapted from [33]

Another property is that numerous GPs can be drawn from the same posterior, which, in a practical view, is a cheap way to generate GP model ensembles. For binary classification of target values +1 and -1, the basic idea is to place a GP prior over a latent (i.e., hidden) function $f(x)$ and pass it through the logistic function $\lambda(z) = \frac{1}{1+e^{-z}}$ to obtain a prior on $\pi(x) = p(y = +1 \mid x)$. [31]

Considering a machine learning classification or regression task, data comes in vast amounts in a practical setting. This poses a computational problem, as a key element in GP posterior calculation is the computationally expensive matrix inversion operation which has the complexity of up to $O(n^3)$. Therefore, the basic GP algorithm performance is bottlenecked by the number of observed data points used. Simple data down-sampling poses a risk of important information loss, so it is not a viable option. However, assume we select a subset of the dataset, which is much smaller than the full set, but still encompasses the essence of the dataset. By "summarizing" the dataset in a sense, and only passing such data to the GP, we should expect the model performance to be comparable to one that uses all the data, without the expense of long computation times. The SVGP model is built on this idea.

Consider a new set of data points X_s, called inducing points, where $n_s \ll n$. The value of the points can be either known or not, by either choosing them to be a subset of the fixed observed data points, or to be estimated by the algorithm, as to find the optimal summarization based on the number of possible data points. The latter option corresponds to the variational aspect of SVGP. This model aims to generate training data with high probability as to accurately "summarize" it. Therefore, it establishes the relationship between GP function values at inducing points and the target variable at training locations. [34]

Furthermore, variational inference technique minimizes the Kullback-Leibler divergence (KL) between a variational GP and the true posterior GP instead of maximization of the log marginal likelihood objective function for traditional GPs. Alternatively, the minimization is equivalently

expressed as the maximization of the variational lower bound of the true log marginal likelihood (Fig. 11), known as the ELBO formula:

$$ELBO(q) = \mathbb{E}[\log p(z, x)] - \mathbb{E}[\log q(z)].$$

Here $p(z, x)$ is the joint density of observed variables x and inducing variables z, which have the density $q(z)$. [34, 35]

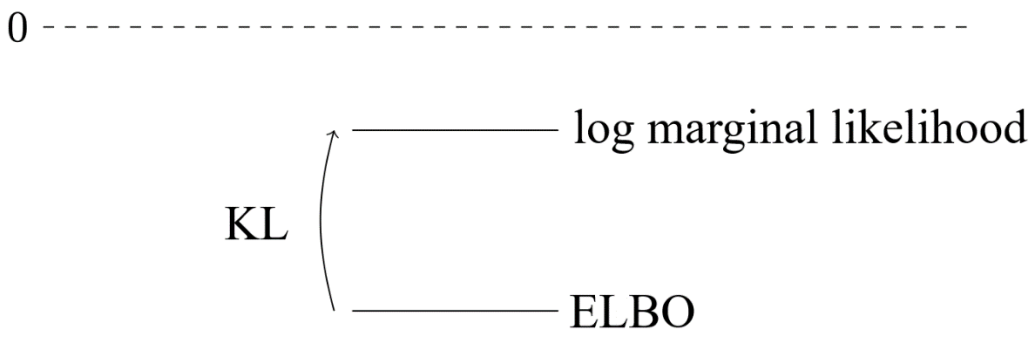


Fig. 11. Visual scheme of relation between marginal likelihood and ELBO

Lastly, a GP extension to multiple outputs can be defined as a GP that approximates T outputs while considering their correlations. The base difference from a single-output GP is that the kernel is now transformed to also model the covariance between its outputs

$$K(x, x') = \begin{bmatrix} k_{11}(x, x') & \cdots & k_{1T}(x, x') \\ \vdots & \ddots & \vdots \\ k_{T1}(x, x') & \cdots & k_{TT}(x, x') \end{bmatrix},$$

where $k_{tt'}(x, x')$ is covariance between outputs $f_t(x)$ and $f_{t'}(x')$. [36]

2.3. Experiment methodology

The CNN models are realized and fitted using the TensorFlow [20] library. The GP models are realized using PyTorch [37] and GPyTorch [38] libraries. All experiments are run on Python [39] using Jupyter Notebook [40], with additional aid of Matplotlib [41], NumPy [42], Scikit-learn [43], pandas [44, 45] and Joblib [46].

2.3.1. Convolutional neural network application

We started with an extensive search for an optimal CNN setup between the three input data variations and the EfficientNet architecture family, from B0 to B6 variants (B7 excluded due to computational and time limitations), resulting in 21 models. The model architectures are standard implementations, except for our added global average pooling layer before the dense output layer for later deep feature extraction purposes (Fig. 12).

```
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_2 (InputLayer)        [(None, 256, 256, 3)]     0

 efficientnetb0 (Functional)  (None, 8, 8, 1280)       4049571

 global_average_pooling2d (G  (None, 1280)             0
 lobalAveragePooling2D)

 dense (Dense)               (None, 4)                 5124

 tf.math.sigmoid (TFOpLambda  (None, 4)                0
 )

=================================================================
Total params: 4,054,695
Trainable params: 4,012,672
Non-trainable params: 42,023
_________________________________________________________________
```

Fig. 12. CNN architecture example for EfficientNet-B0

Models are set up with the Adam [24] optimizer using learning rate of 10^{-5} for raw or preprocessed data and learning rate of 10^{-6} for augmented data. As the models are set up for a multi-label classification task, binary cross entropy is used

$$-\frac{1}{N}\sum_{i=1}^{N}[y_i \log p(x_i) + (1-y_i)\log(1-p(x_i))]$$

as the loss function, where N is the total number of samples, y_i – the ground truth and $p(x_i)$ – the classifier prediction based on the associated features x_i of the ground truth. The sigmoid function

$$\frac{1}{1+e^{-x}}$$

is used for output layer activations. Fitting is started using the weights obtained from the ImageNet dataset [18] for faster model convergence. Performance is evaluated using the AUC score separately for each label

$$\frac{\sum_{i=1}^{m}\sum_{j=1}^{n} 1_{x_i > y_i}}{mn}$$

where m and n are numbers of positive and negative samples respectively, x_i $(i = \overline{1,m})$ and y_j $(j = \overline{1,n})$ are outputs of a classifier on positive and negative samples, 1_X is the indicator function of a set X [47]. The TensorFlow implementation of AUC calculation is used during fitting to collect training and validation results, however we use the Scikit-learn implementation for testing results for a consistent comparison with GP results.

EfficientNet variants from B0 to B2 are fit with a batch size of 32, B3-B5 with a batch size of 16 and B6 with batches of 8 images. Models are fit past the point that would be considered overfitting (when training loss is lower than validation loss), as our validation AUC values improve past that point. We also decrease the learning rate by a factor of 0.1 to see if any improvement in performance can be

gained after the current fit metrics saturate. Finally, an optimal model is chosen as the performance baseline and deep feature extractor.

2.3.2. Gaussian process application

The CNN global average pooling layer's outputs are collected as our deep features for each subset of data to use as input for our GP models. As this results in a large number of features, PCA is performed on them to reduce the strain of calculations of the GP.

The GP is set up as an independent multitask SVGP using 800 inducing points, the initial values are set to the first 800 observations of the training set, and the inducing locations are made part of the learned parameters. The mean field variational distribution, independent multitask variational strategy, and the multivariate Bernoulli distribution likelihood (set up from sigmoid values of the GP outputs) is used, and the batch shape for the GP components is set to be equal to the number of labels in order to set it up as a multi-label classifier.

For the mean function, the primary experiments are run using $\mu(x) = 0$, but after finding the optimal kernel setup we also test if a learned $\mu(x) = C$ improves the results. For the kernel, experiments are focused on the Matérn-5/2 covariance function

$$k_{\text{Matérn}}(x, x') = \sigma^2(1 + \sqrt{5}\frac{\|x - x'\|}{\rho} + \frac{5}{3}\left(\frac{\|x - x'\|}{\rho}\right)^2)e^{-\sqrt{5}\frac{\|x-x'\|}{\rho}}$$

where σ and ρ are learnable outputscale and lengthscale parameters. The ARD variant of the kernel is also considered, where each input dimension m of d has its own lengthscale parameter, therefore the term

$$\frac{\|x - x'\|}{\rho}$$

is replaced with

$$\sum_{m=1}^{d} \frac{\|x_m - x'_m\|}{\rho_m}.$$

Furthermore, it is investigated if additional parameters in the covariance function improve the results, so an addition to the Matérn kernel is also considered in the form of a linear kernel:

$$k_{Composite}(x, x') = k_{\text{Matérn}}(x, x') + \nu x^T x'$$

where ν is the variance parameter.

The GP models are optimized with minibatches of 16 samples with a learning rate of 10^{-3} using the Adam optimizer, by calculating the variational ELBO of the selected likelihood. The learning rate is not decreased further as it does not improve performance. Lastly, for the selected options of the covariance function, the effect of Gaussian noise applied on input is tested with mean $\mu = 0$ and standard deviation values of $\sigma = 0.1$ and $\sigma = 0.5$, resulting a total of 12 primary test GP models.

We collect the GP optimization metrics every $N/100$ steps, where N is the total number of minibatch iterations. The loss values of GP are collected for both training and validation subsets, however label AUC values are calculated only for the validation subset, to reduce the optimization loop time and since there is no risk for the GP to overfit. Furthermore, metrics are collected only for GP ensembles, of sizes 10, 100 and 1000 to investigate if there are differences in performance.

3. Results and discussion

3.1. Convolutional neural network results

After extensive training of the CNN models, it was found that the best data handling tactic for this dataset was applying augmentations, and the worst was applying preprocessing. This is a natural conclusion, as augmentations are a tried-and-true approach to fitting regularization, and since the original training set is rather small, the extra introduced variability of samples is beneficial. The lowest performance of preprocessed inputs can be explained by the possible loss of information due to local color mean subtraction, although the average performance for CRS overtakes its raw counterpart, meaning for some labels it helps to accent the relevant features (Fig. 13).

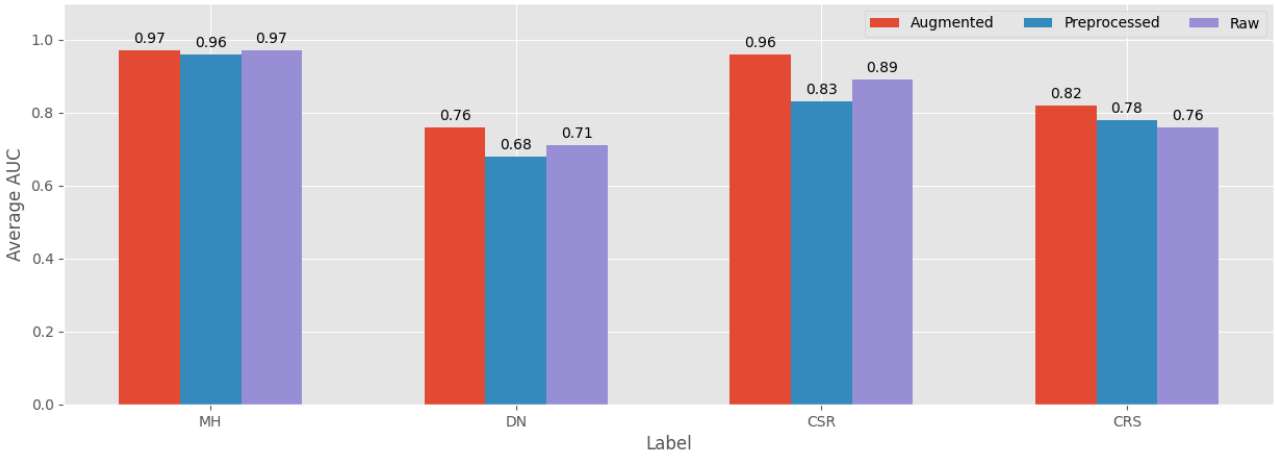


Fig. 13. Testing AUC of each label by input handling tactic, averaged over EfficientNetB0-B6

While using the augmented inputs, the performance of architecture variants does not differ greatly, but it seems that EfficientNet-B1 is best suited for 256×256 px inputs, as the slight decrease in performance as the models get bigger would indicate that the architectures are becoming too large for the used input dimension (Fig. 14).

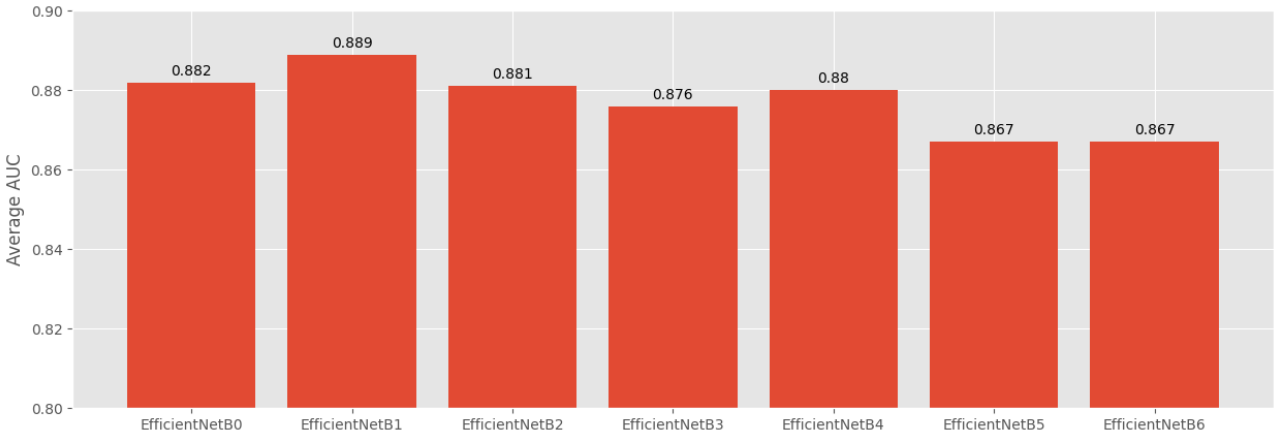


Fig. 14. Testing AUC of each architecture variant using augmented inputs, averaged over labels

The augmented input EfficientNet-B1 variant obtains high performance for MH, CSR and CRS labels. That is expected, as these labels have the most prominent features, as opposed to the subtle

dotting of DN, for which the AUC performance leaves more to be desired (Fig. 15). EfficientNet-B1 is selected for extraction of deep visual features in the next section.

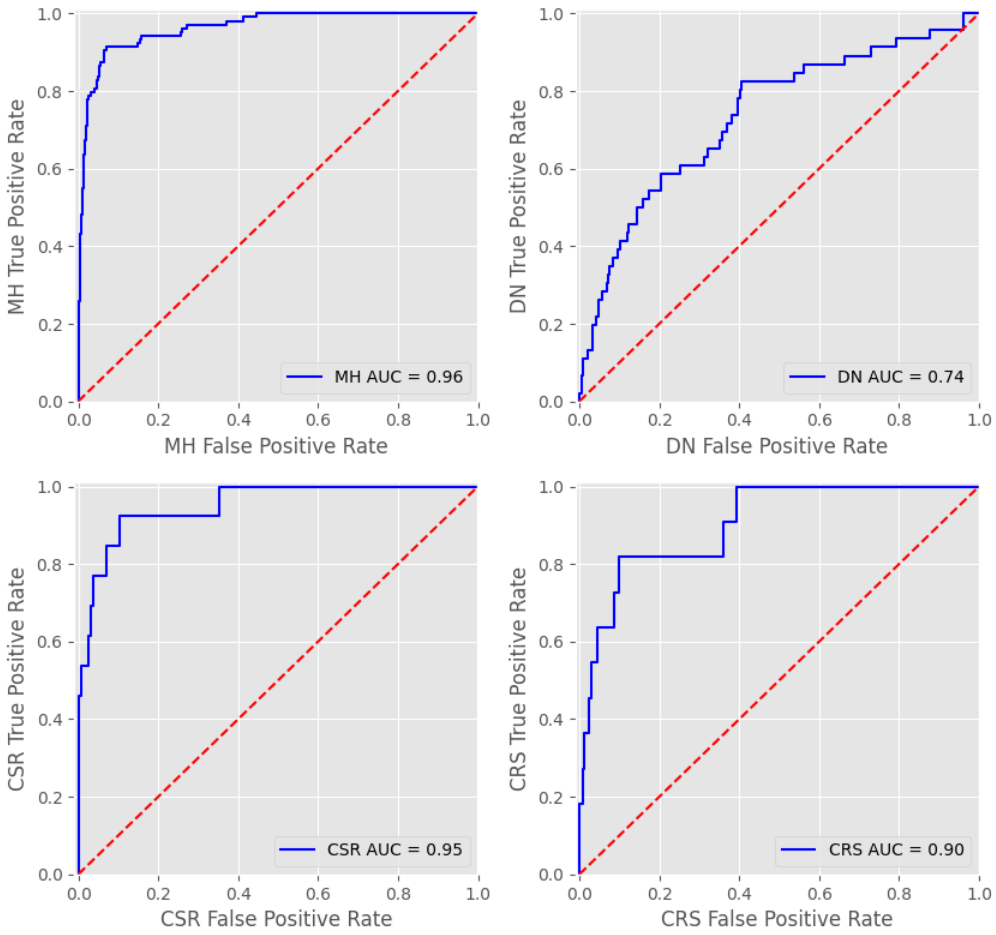


Fig. 15. Test set receiver operating characteristic curves of the augmented input EfficientNet-B1 for each label

3.2. Gaussian process results

Training Gaussian processes requires larger amounts of computer memory. Therefore, to avoid computational issues a PCA dimensionality reduction was used. After performing PCA and reducing the dimensions of input data from 1280 deep features, we keep the first 550 components, which explain ~96% of total variance (Fig. 16).

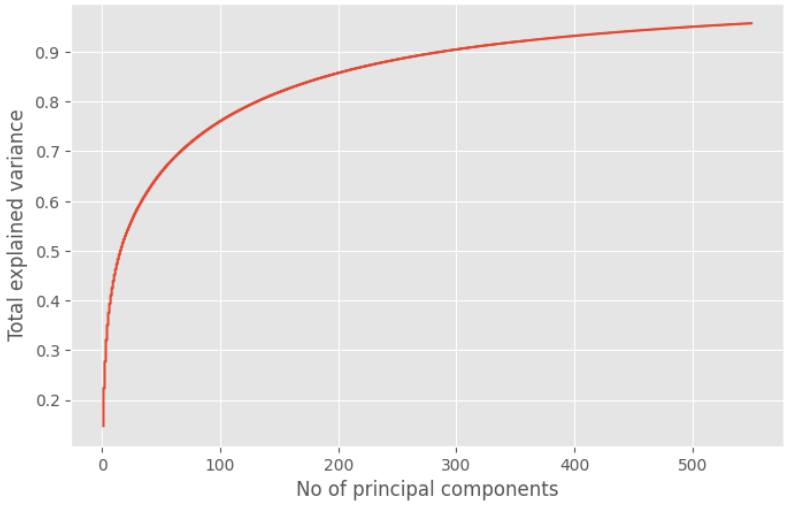


Fig. 16. Total deep feature explained variance by number of principal components

As was detailed in the methods section, a number of different kernel design options were considered. In total 12 alternative models were trained and analyzed. The following are condensed insights of what was learned from this process:

- The basic Matérn kernel cannot handle noise with standard deviation as strong as $\sigma = 0.5$, whereas $\sigma = 0.1$ barely makes an effect;
- The ARD Matérn kernel cannot handle noise of even $\sigma = 0.1$;
- The basic composite kernel is indifferent to the use of noise;
- The composite variant using the ARD Matérn kernel benefits from noisy inputs, the difference in results is an inprovement of 0.3 AUC at most;
- It seems that the CNN learned a linear decision boundary within the deep feature space, as the composite kernel variants outperform the basic ones, as the addition of the linear kernel introduces linearity to GP's decision boundary;
- Increasing the size of the ensemble (i.e. number of samples from the trained Gaussian process) does not make a significant difference to our setups, however since the ensemble generation is very cheap, we can continue to use our maximum selected number of GPs for increased probability distribution stability.

In the reviewed literature about the method, there was a notion of GPs being able to discern that their knowledge of the domain is limited. In hindsight, an assumption can be made that this behaviour can be applied to the evaluation of probability uncertainty. Perhaps difficult cases, e.g., ones that have multiple pathologies, can confuse the classifier and the GP could model its parameters in a way that would result in an "indecisive" label probability of $p = 0.5$. However, this type of behaviour is not possible in our setting, where a multi-label classification is performed on deep features, obtained from a multi-label CNN classifier; I believe, this expectation should be attributed to multi-class problem cases.

A multi-label CNN classifier models independent binary classifications for each label. This results in discriminative deep features, which are mainly modelled to allow an efficient detection of the required subject, without any dissection of features for supplementary information, as they are simply generalized into groups of positive and negative samples. Of course, this is the intended behaviour

which allows the linear boundary of accurate separation of sample groups. However, as the deep features are already so generalized on a per-label basis, any supplementary information, in our case regarding any extra unknown pathologies, is lost. This was tested with methods described in [48] for various subsets of data (both for CNN and GP), but as experiments were not fruitful, any results regarding them are omitted for brevity. Furthermore, the approach of $p \to 0.5$ when feature variance increases is only possible for GP kernels like RBF or Matérn, which model non-linear decision boundaries, but since the deep features are well-refined for a linear boundary, in our case we only suffer a loss of performance, as the pilot model results reveal.

After selecting and comparing the top performing setups of each kernel variant, we deem that the best performing setup was one that used an ARD Matérn and linear kernel composition with $\sigma = 0.5$ Gaussian noise used on input. The difference between the composite kernel and its ARD variant is minuscule, however the performance of latter is ever so slightly better with the MH label (Fig. 17), which can be attributed to the increased parametrization of the model.

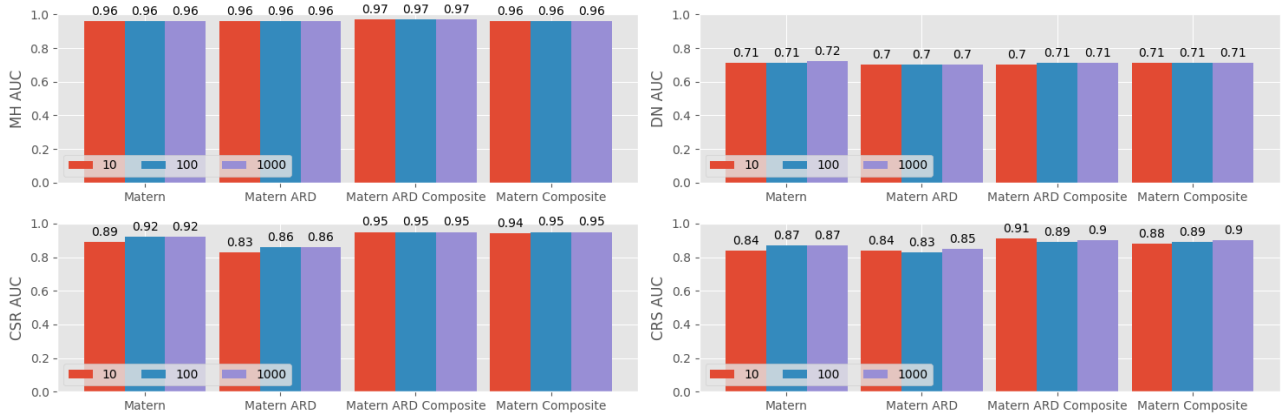


Fig. 17. Top performing setups of each kernel variant

The top performing setup is then reoptimized using a constant mean, to see for improvements. Lastly, the setup is reoptimized using deep features obtained from the raw images, to see whether using the original feature distribution improves the results. As this time the full training set only consists of 1920 original observations, 550 components manage to explain 98% of total variance. This helps to improve upon the CNN results for MH, CSR, CRS labels by 0.1 AUC and bring DN label performance up to par (Fig. 18).

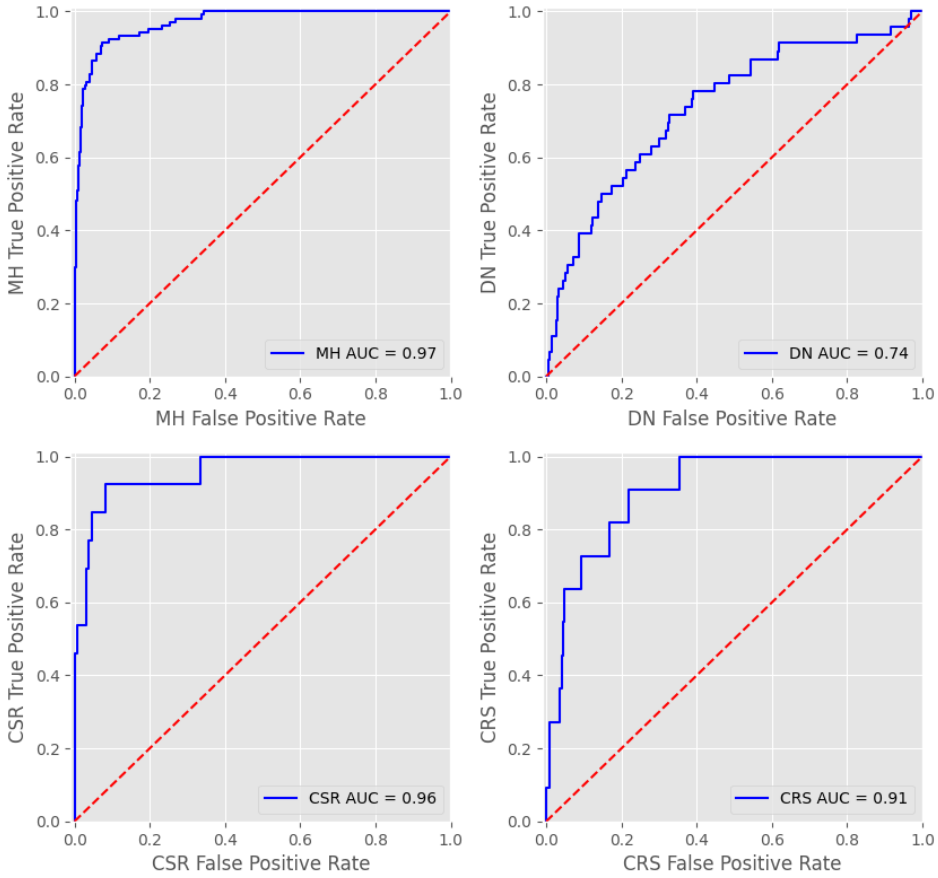


Fig. 18. Test set receiver operating characteristic curves for 1000 GP ensemble using constant mean, composite ARD kernel and raw image deep features

A variation of max pooled deep features and their concatenation to global average pooling features were also tried, but did not improve the performance, possibly due to the CNN weights being trained with an architecture setup using the global average pooling layer. DGP models were also tested, with a single or double hidden layer setup employing multiple GPs, set up in a similar manner to the best GP model, however they were prone to overfitting and underperformed in comparison to the CNN and GP. Lastly, two versions of hybrid ensembles were tested – one where the CNN predictions share the same weight as the GP realizations, and one where the GP realization results are first averaged and then share the same weight as the CNN predictions. For the former, the use of 1000 GP samples far outweighs the CNN predictions and make no impact on the results when comparing to pure GP performance. For the latter, it resulted in performance similar to CNN's, due to CNN probabilities having more extreme values.

3.3. Gaussian process probability uncertainty utilization

Since GP ensembles can be generated by sampling functions from the learned distribution, probability uncertainty estimation can be obtained. 1000 realizations from the trained Gaussian process are sampled and then passed through a logistic function to obtain a sample of probabilities for each testing case. This sample represents a posterior distribution of an event where a given test image belongs to a particular class. Difficult test set images, i.e. the ones for which diagnostics is more challenging, are expected to have wider confidence intervals and therefore a larger uncertainty estimates (of course this is only valid under a condition that comparisons are made for the same sample size, as increasing

the sample size will generally reduce uncertainty of the prediction). Uncertainty levels could be used to determine whether a particular test image requires attention of a professional and therefore whether it should be diagnosed manually rather than automatically. The probability range can and should be fine tuned for each label separately, as labels with a sample distribution of smaller variance in the deep feature hyperspace will have tighter decision boundaries, especially ones with smaller sample sizes. For further discussion it is assumed that the cases referred to a medical specialist will receive the correct diagnosis.

Probability threshold of $p = 0.151$ is selected to identify CNN errors for MH label. The selected probability threshold corresponds to 91.35% sensitivity and 91.6% specificity. By choosing the probability variation range wider than 0.6, 69 cases are flagged, 36 of which would have been mislabelled by the CNN. By referring the flagged cases this system would raise the AUC score from 0.969 to 0.984. In other words, around 1 out of 10 cases would require a manual review to improve AUC by 0.015. For the extreme improvement to 1 AUC, around half of all cases would need to be manually reviewed by selecting the probability range of 0.05 or larger (Fig. 19).

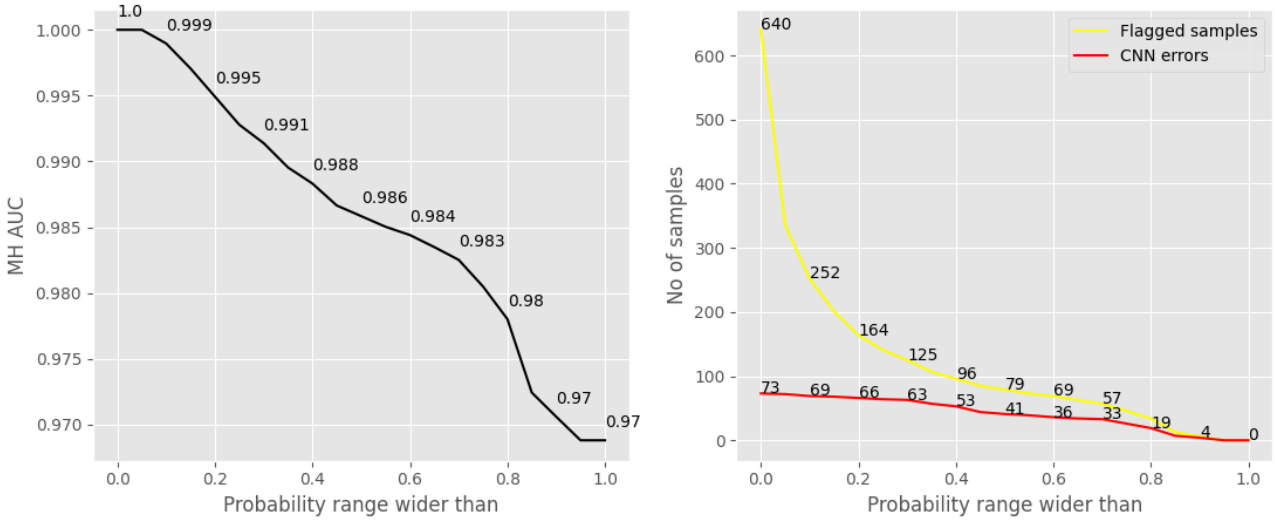


Fig. 19. MH label AUC, flagged sample number and CNN error number dependency on probability range width

Fig. 20 depicts an MH case, that would be caught using a probability range wider than 0.6. The MH in this sample is rather obvious, with visual artefacts around the edge and center of image. Both CNN and GP predictions are lower than the set threshold, therefore a false negative labelling would occur. However, the GP prediction is ever so slightly closer to the threshold.

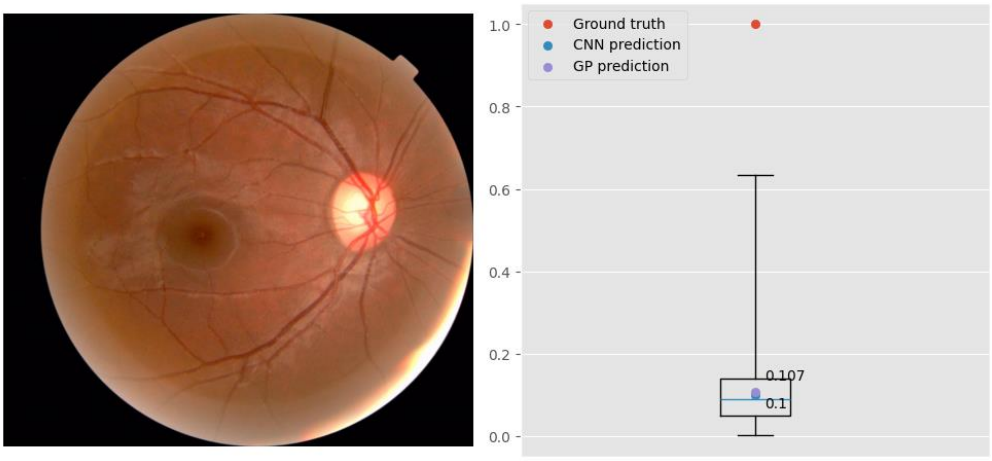


Fig. 20. Image with MH, its CNN/GP predictions, and GP prediction distribution. Whiskers mark min/max values

Fig. 21 depicts a case, that would be wrongfully labelled as MH by the CNN, but was correctly labelled by the GP. Nonetheless, it was also caught by selecting cases with MH probability variation range larger than 0.6. It is possible that the sparse dark red dotting, which is indicative of another pathology, has confused the CNN model into labelling it as MH. Here we observe again, that the GP is closer to the correct result than the CNN. The probability distribution is rather symmetrical and wide, which is indicative of a difficult case, which throughout the ensemble rests of varying places of the decision boundary.

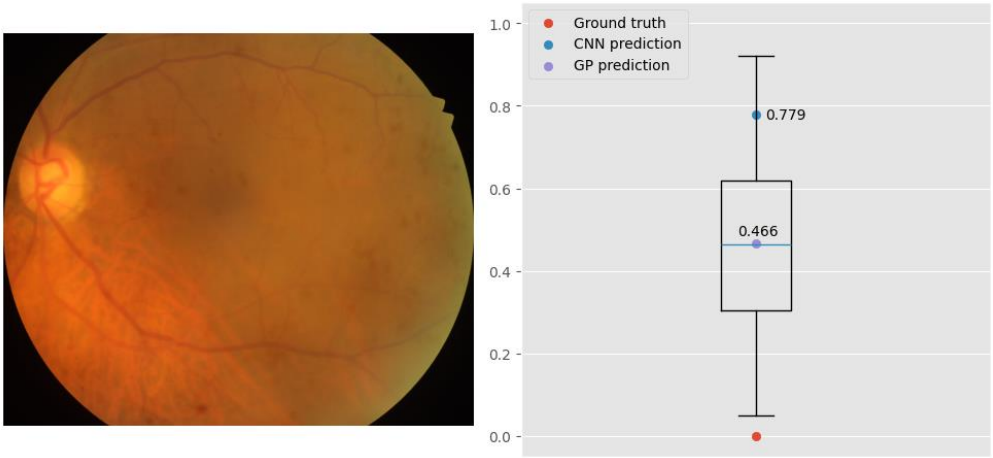


Fig. 21. Sample, wrongfully labelled with MH by CNN, its CNN/GP predictions, and GP prediction distribution. Whiskers mark min/max values

Probability threshold of $p = 0.033$ is selected to identify CNN errors for DN label. The selected probability threshold corresponds to 69.57% sensitivity and 67.68% specificity. By choosing the probability variation range wider than 0.2, 161 cases are flagged, 133 of which would have been mislabelled by the CNN. By referring the flagged cases this system would raise the AUC score from 0.737 to 0.88. In other words, around 1 out of 4 cases would require a manual review to improve AUC by 0.143. In this label's case, probability range of 0.1 would result in roughly half of all cases being manually reviewed for AUC of 0.947 (Fig. 22).

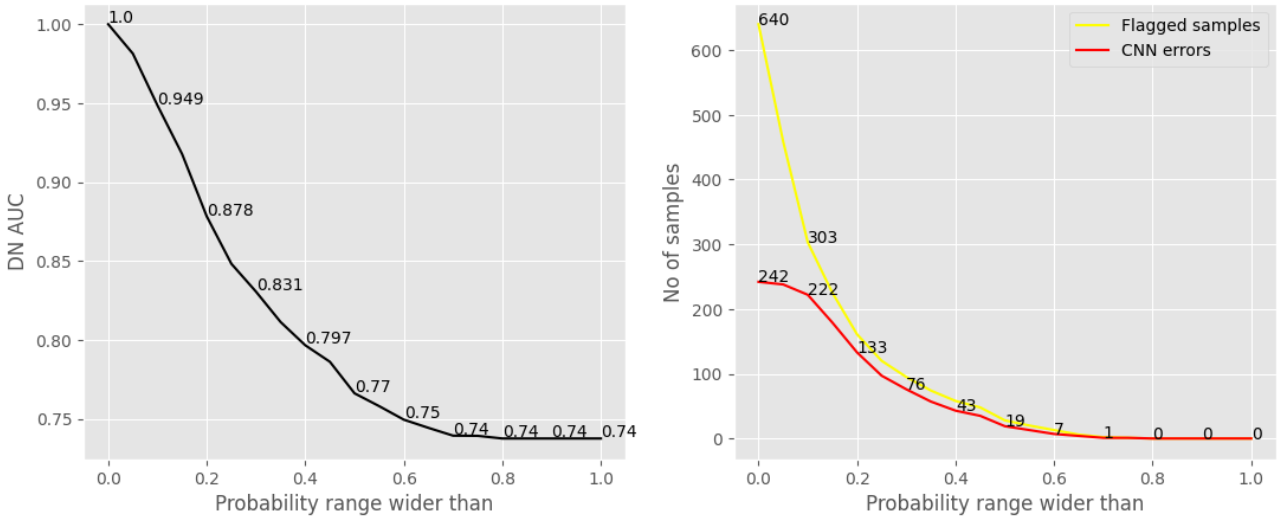


Fig. 22. DN label AUC, flagged sample number and CNN error number dependency on probability range width

Fig. 23 depicts a DN case, that would be caught using a probability range wider than 0.1. The DN in this sample is singular, most prominently visible on the right side. In this case the GP probability barely manages to overcome the threshold, whereas the CNN result falls short.

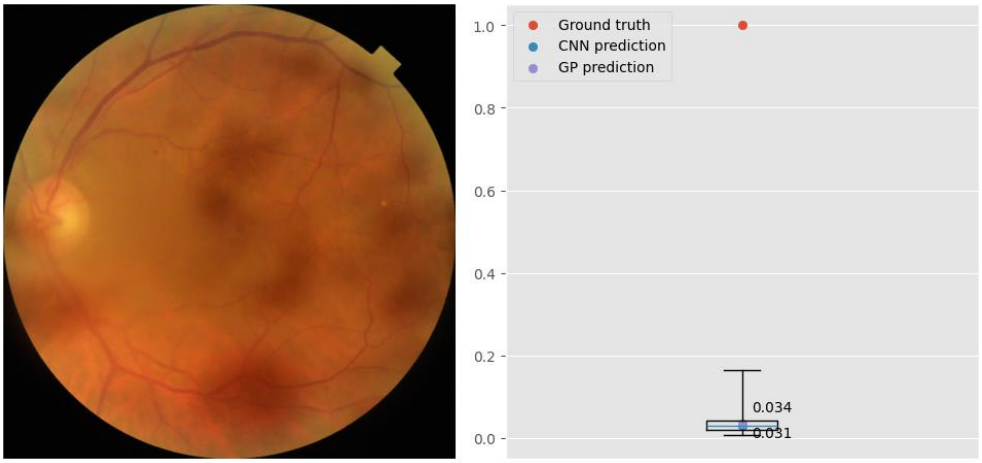


Fig. 23. Sample with DN, its CNN/GP predictions, and GP prediction distribution. Whiskers mark min/max values

Fig. 24 depicts a case, that would be wrongfully labelled as DN by both CNN and GP. It was caught by selecting cases with DN probability variation range larger than 0.2. This is a visually confusing case, as there is a certain light dotting, especially in the top right corner of the image, which has fooled the models into labelling it as DN. The GP ensemble probability distribution is rather symmetrical, and the average is lower than the GP forecast, but that is not enough for a correct result. As the probability range for this case is rather wide, it would be caught easily.

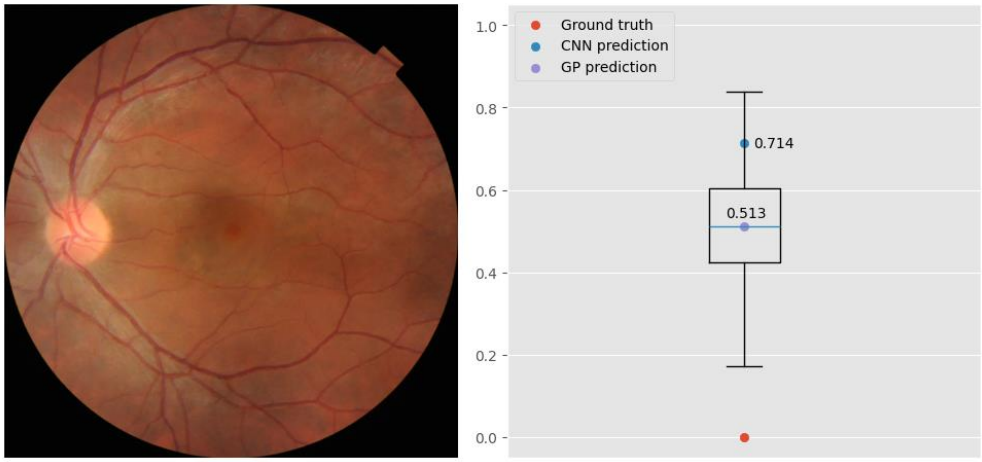


Fig. 24. Sample, wrongfully labelled with DN by both models, its CNN/GP predictions, and GP prediction distribution. Whiskers mark min/max values

A probability threshold of $p = 0.018$ is selected to identify CNN errors for CSR label. The selected probability threshold corresponds to 92.31% sensitivity and 91.55% specificity. By choosing the probability variation range wider than 0.1, 81 cases are flagged by choosing the probability variation range wider than 0.1, 58 of which would be mislabelled by the CNN. By referring the flagged cases this system would raise the AUC score from 0.954 to 0.982. In other words, around 12% of cases would require a manual review to improve AUC by 0.028. For a near-perfect AUC of 0.995, roughly a third of cases would require referring to a medical specialist, with a probability range wider than 0.05 (Fig. 25).

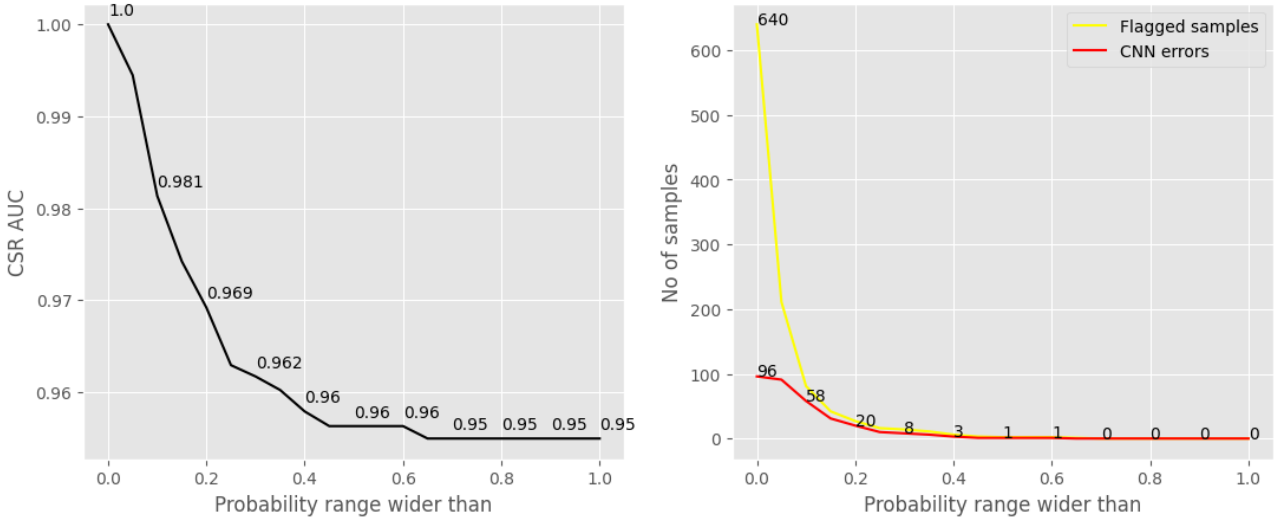


Fig. 25. CSR label AUC, flagged sample number and CNN error number dependency on probability range width

Fig. 26 depicts a false positive CSR case, that would be caught using a probability range wider than 0.1. The visual features of depicted fundus are more visually more similar to one affected by CRS. Yet again the GP prediction is lower than the CNN's and is twice as closer to the threshold. However, that is not enough for a correct result but nonetheless it is still referable.

Fig. 26. Sample with false positive CSR, its CNN/GP predictions, and GP prediction distribution. Whiskers mark min/max values

Fig. 27 depicts another false positive CSR case, that would be caught using a probability range wider than 0.1. The yellow dotting is indicative of another pathology, but there are no discernable visual features indicative of subretinal fluid or CSR. Once more, the average of GP probabilities is closer to the decision threshold.

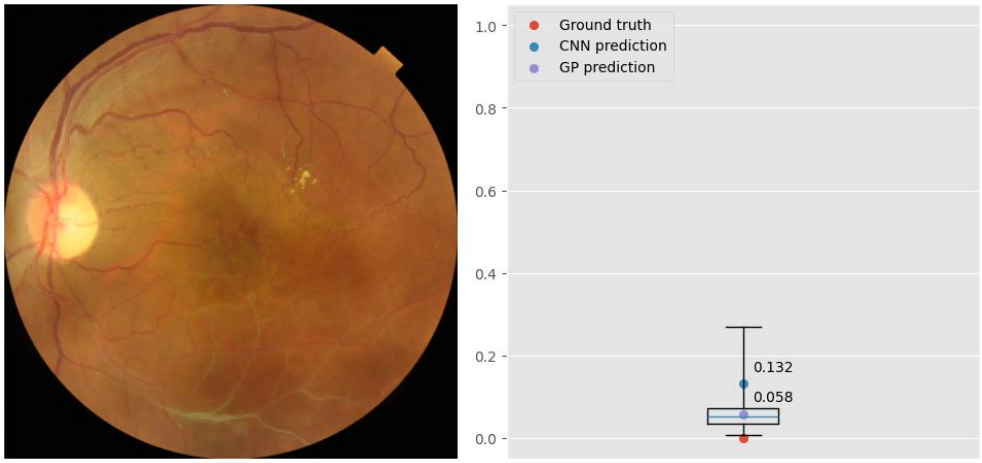


Fig. 27. Sample with false positive CSR, its CNN/GP predictions, and GP prediction distribution. Whiskers mark min/max values

A probability threshold of $p = 0.011$ is selected to identify CNN errors for CRS label. The selected probability threshold corresponds to 81.82% sensitivity and 81.72% specificity. By choosing the probability variation range wider than 0.1, 101 cases are flagged, 91 of which would be mislabelled by the CNN. By referring the flagged cases this system would raise the AUC score from 0.903 to 0.97. In other words, around 15% of cases would require a manual review to improve AUC by 0.067. The perfect AUC of 1 can be achieved by also referring cases of probability range wider than 0.05, which would result in a 60% decrease in manual labour (Fig. 28).

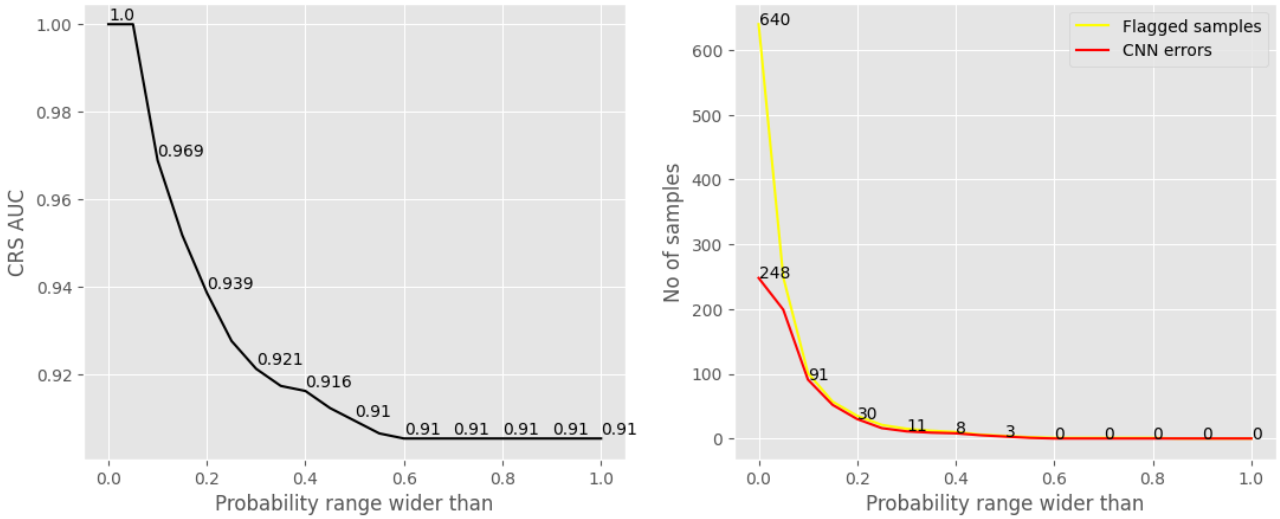


Fig. 28. CRS label AUC, flagged sample number and CNN error number dependency on probability range width

Fig. 29 depicts a false positive CRS case, that would be caught using a probability range wider than 0.1. The eye is definitely affected by a pathology; however it is not CRS. Yet again, even though the GP prediction is closer to the threshold than CNN's, it fails to be low enough, but the possibility to evaluate the variability of the probability redeems the performance.

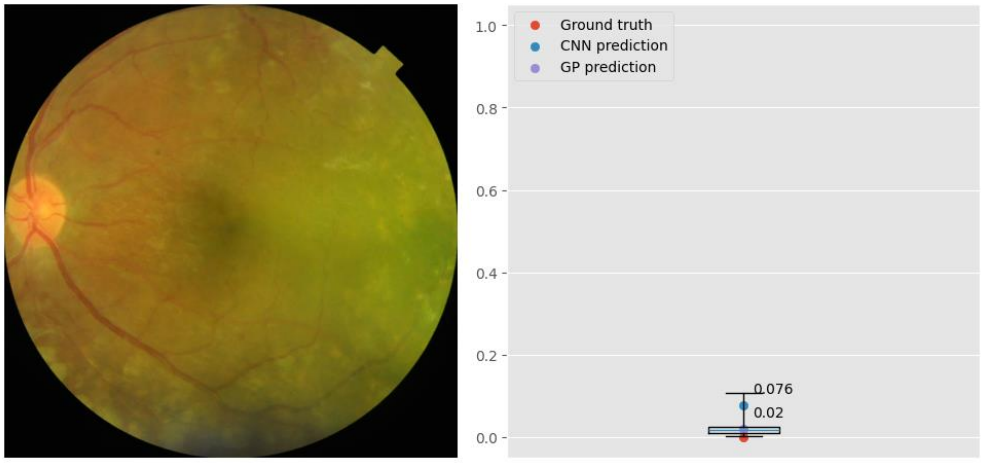


Fig. 29. Sample with false positive CRS, its CNN/GP predictions, and GP prediction distribution. Whiskers mark min/max values

Lastly, Fig. 30 is another false positive CRS case, that would be caught using a probability range wider than 0.1. From the distribution of probabilities, we see that this case is highly irregular. Both mentioned cases have unusual discoloration, which, seemingly, confuses the models as fundi affected by CRS do have irregual colouring, however it is much more distinctive and less gradient. Yet again the GP probability is lower than CNN's, but not to a satisfactory degree.

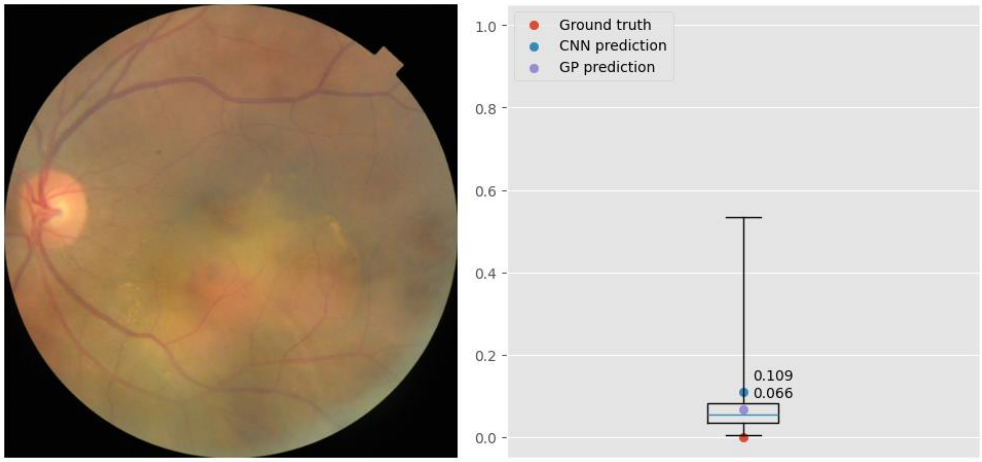


Fig. 30. Sample with false positive CRS, its CNN/GP predictions, and GP prediction distribution. Whiskers mark min/max values

Overall, we can see that the averaging over the GP ensemble probabilities consistently results in a more regularized, "softened" estimate, when compared to the CNN counterparts. Understandably, the improvements in performance for already well-performing labels are not extreme and the tradeoff of additional effort of manual labor might not sound too attractive. However, for more difficult cases, e.g., DN detection, this might be a very attractive alternative to a classifier that does not have stellar performance and cannot supply any sort of measure of confidence for its probability forecasts, or a great alternative to pure manual labour. Furthermore, MH results show that it is possible to achieve perfect performance with only 50% of manual labor and CRS results show that the same is possible with only 40% of manual labor.

Conclusions

1. From EfficientNet-B0 to EfficientNet-B6, EfficientNet-B1 is best suited for eye fundus image inputs: observed drop in performance as the model size increase indicates deterioration of learned features and overfitting.
2. The subtle discolored dotting of drusen are more difficult features to capture for the EfficientNet convolutional neural network family, with average AUC ranging from 0.68 to 0.76.
3. Matérn-5/2 kernel is sensitive to stronger noise in data – a Gaussian process using deep feature inputs polluted with Gaussian noise with mean $\mu = 0$ and standard deviation $\sigma = 0.5$ will fail to fit data.
4. The additive composition of the Matérn-5/2 and linear kernels achieved better results compared to just the Matérn-5/2 kernel. Possible explanation would be that the CNN learned a linear decision boundary within the deep feature space, therefore a linear kernel in GP context was optimal to learn these boundaries, while the Matérn kernel captured local nonlinearities which were not resolved by the linear kernel.
5. Among the tested Gaussian process model setups, the best results are obtained by using a constant mean, an additive composition of Matérn-5/2 kernel (with automatic relevance determination) and a linear kernel in combination with added Gaussian noise ($\mu = 0, \sigma = 0.5$) on the inputs (first 550 principal components). The improvement was by 0.1 AUC for media haze, central serous retinopathy and chorioretinitis labels.
6. Cases that are more prone to be mislabeled by the CNN generally have larger uncertainty. Therefore, by using uncertainty estimates, provided by GPs, cases that require further manual review can be identified:

   – Media haze detection – AUC of 1.0 with manual review of ~50 % of all cases, as opposed to automatically labelling all samples with AUC of 0.97;
   – Drusen detection – AUC of 0.947 can be achieved by manually reviewing ~50 % of all cases, as opposed to automatically labelling all samples with AUC of 0.74.;
   – Central serous retinopathy detection – AUC of 0.995 can be achieved by manually reviewing ~33 % of all cases, as opposed to automatically labelling all samples with AUC of 0.96;
   – Chorioretinitis detection – perfect performance of 1 AUC can be achieved by manually reviewing ~40% of all cases, as opposed to automatically labelling all samples with AUC of 0.91.

List of references

1. TING, Daniel Shu Wei, et al. Artificial intelligence and deep learning in ophthalmology. In: *British Journal of Ophthalmology*. 2019, vol. 103, no. 2, pp. 167-175. Available from: DOI: https://doi.org/10.1136/bjophthalmol-2018-313173
2. GULSHAN, Varun, et al. Development and Validation of a Deep Learning Algorithm for Detection of Diabetic Retinopathy in Retinal Fundus Photographs. In: *JAMA.* 2016, vol. 316, no. 22, pp. 2402-2410. Available from: https://doi.org/doi:10.1001/jama.2016.17216
3. TING, Daniel Shu Wei, et al. Development and validation of a deep learning system for diabetic retinopathy and related eye diseases using retinal images from multiethnic populations with diabetes. In: *JAMA.* 2017, vol. 318, no. 22, pp. 2211-2223. Available from: https://doi.org/doi:10.1001/jama.2017.18152
4. GRASSMANN, Felix, et al. A deep learning algorithm for prediction of age-related eye disease study severity scale for age-related macular degeneration from color fundus photography. In: *Ophthalmology*. 2018, vol. 125, no. 9, pp. 1410-1420. Available from: DOI: https://doi.org/10.1016/j.ophtha.2018.02.037
5. GARGEYA, Rishab, Theodore LENG. Automated Identification of Diabetic Retinopathy Using Deep Learning. In: *Ophthalmology*. 2017, vol. 124, no. 7, pp. 962-969. Available from: DOI: https://dx.doi.org/10.1016/j.ophtha.2017.02.008
6. KIND, Adrian, George AZZOPARDI. An Explainable AI-Based Computer Aided Detection System for Diabetic Retinopathy Using Retinal Fundus Images. In: *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Springer International Publishing, 2019, pp. 457-468. ISBN 9783030298876.
7. LOFTUS, Tyler J., et al. Uncertainty-aware deep learning in healthcare: a scoping review. In: *PLOS Digital Health*. 2022. Available from: DOI: https://doi.org/10.1371/journal.pdig.0000085
8. LAMBERT, Benjamin, et al. Trustworthy clinical AI solutions: a unified review of uncertainty quantification in deep learning models for medical image analysis. In: *Artificial Intelligence in Medicine.* 2024. Available from: https://doi.org/10.1016/j.artmed.2024.102830
9. ARAÚJO, Teresa, et al. DR| GRADUATE: Uncertainty-aware deep learning-based diabetic retinopathy grading in eye fundus images. In: *Medical Image Analysis*. 2020, vol. 63. Available from: DOI: https://doi.org/10.1016/j.media.2020.101715
10. AYHAN, Murat Seçkin, et al. Expert-validated estimation of diagnostic uncertainty for deep neural networks in diabetic retinopathy detection. In: *Medical Image Analysis*. 2020, vol. 64. Available from: DOI: https://doi.org/10.1016/j.media.2020.101724
11. LEIBIG, Christian, et al. Leveraging uncertainty information from deep neural networks for disease detection. In: *Scientific reports*. 2017, vol. 7. Available from: DOI: https://doi.org/10.1038/s41598-017-17876-z
12. WANG, Xi, et al. UD-MIL: uncertainty-driven deep multiple instance learning for OCT image classification. In: *IEEE Journal of Biomedical and Health Informatics*. 2020, vol. 24, no. 12, pp. 3431-3442. Available from: DOI: https://doi.org/10.1109/JBHI.2020.2983730
13. SEEBÖCK, Philipp, et al. Exploiting Epistemic Uncertainty of Anatomy Segmentation for Anomaly Detection in Retinal OCT. In: *IEEE Transactions on Medical Imaging*. 2020, vol. 39, no. 1, pp. 87-98. Available from: DOI: https://doi.org/10.1109/TMI.2019.2919951

14. CHALLIS, Edward, et al. Gaussian process classification of Alzheimer's disease and mild cognitive impairment from resting-state fMRI. In: *NeuroImage*. 2015, vol. 112: pp. 232-243. ISSN 1053-8119. Available from: DOI: https://doi.org/10.1016/j.neuroimage.2015.02.037
15. *Diabetic Retinopathy Detection* [interactive]. 2015 [accessed 2024-04-25]. Available from: https://www.kaggle.com/c/diabetic-retinopathy-detection/overview
16. TOLEDO-CORTÉS, Santiago, et al. Hybrid deep learning Gaussian process for diabetic retinopathy diagnosis and uncertainty quantification. In: *Ophthalmic Medical Image Analysis: 7th International Workshop, OMIA 2020, Held in Conjunction with MICCAI 2020, Lima, Peru, October 8, 2020, Proceedings 7*. Springer International Publishing, 2020. ISBN 978-3-030-63419-3. Available from: DOI: https://doi.org/10.1007/978-3-030-63419-3_21
17. GRAHAM, Ben. Kaggle diabetic retinopathy detection competition report. In: *University of Warwick*. 2015, vol. 22. Available from: https://kaggle-forum-message-attachments.storage.googleapis.com/88655/2795/competitionreport.pdf
18. DENG, Jia, et al. ImageNet: A large-scale hierarchical image database. In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. Miami, FL, 2009. pp. 248-255. Available from: DOI: https://doi.org/10.1109/CVPR.2009.5206848
19. BRADSHAW, John, et al. Adversarial examples, uncertainty, and transfer testing robustness in Gaussian process hybrid deep networks. 2017. Available from: DOI: https://doi.org/10.48550/arXiv.1707.02476
20. ABADI, Martín, et al. TensorFlow: Large-scale machine learning on heterogeneous systems. 2015. Available from: https://www.tensorflow.org/
21. MATTHEWS, Alexander G. de G., et al. GPflow: A Gaussian process library using TensorFlow. In: *Journal of Machine Learning Research 18*. 2017, no.40. pp. 1-6
22. SIEBERT, Marlin, et. al. Uncertainty Analysis of Deep Kernel Learning Methods on Diabetic Retinopathy Grading. In: *IEEE Access*. 2023, vol. 11, pp. 146173-146184. Available from: DOI: https://doi.org/10.1109/ACCESS.2023.3343642
23. WILSON, Andrew G., et al. Stochastic variational deep kernel learning. In: *Advances in neural information processing systems*. 2016, vol. 29, pp. 2586-2594. ISBN: 9781510838819
24. Kingma, Diederik P., and Jimmy BA. Adam: A method for stochastic optimization. 2014. Available from: https://arxiv.org/abs/1412.6980
25. PACHADE, Samiksha, et al. Retinal Fundus Multi-disease Image Dataset (RFMiD). IEEE Dataport: 2020. Available from: DOI: https://doi.org/10.21227/s3g7-st65
26. DAMIANOU, Andreas, and Neil D. LAWRENCE. Deep gaussian processes. In: *Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics*. PMLR, 2013, vol. 31, pp. 207-215. Available from: https://proceedings.mlr.press/v31/damianou13a.pdf
27. PANOS, Aristeidis, et al. Large scale multi-label learning using Gaussian processes. In: *Machine Learning*. 2021, vol. 110(5), pp. 965-987. Available from: DOI: https://doi.org/10.1007/s10994-021-05952-5
28. HE, Jianjun, et al. Bayesian multi-instance multi-label learning using Gaussian process prior. In: *Machine learning.* 2012, vol. 88 (1), pp. 273-295. Available from: DOI: https://doi.org/10.1007/s10994-012-5283-x
29. PACHADE, Samiksha, et al. Retinal fundus multi-disease image dataset (RFMiD): A dataset for multi-disease detection research. *Data*. 2021, vol. 6(2). Available from: DOI: https://doi.org/10.3390/data6020014

30. TAN, Mingxing, and Quoc V. LE. EfficientNet: Rethinking model scaling for convolutional neural networks. In: *International conference on machine learning*. 2019. pp. 6105-6114. Available from: DOI: https://doi.org/10.48550/arXiv.1905.11946
31. RASMUSSEN, Carl Edward and Christopher K.I. WILLIAMS. *Gaussian process for machine learning*. London, England : The MIT Press, 2006. ISBN 026218253X. Available from: https://gaussianprocess.org/gpml/chapters/RW.pdf
32. ZHANG, Aston, et al. *Dive into deep learning*. Cambridge, England : Cambridge University Press, 2024. ISBN 1009389432. Available from: https://d2l.ai/index.html
33. SAMMUT, Claude, et al. Gaussian Process. In: *Encyclopedia of Machine Learning*. New York, New York: Springer, 2011. pp. 428-439. Available from: DOI: https://doi.org/10.1007/978-0-387-30164-8
34. TITSIAS, Michalis. Variational learning of inducing variables in sparse Gaussian processes. In: *Artificial intelligence and statistics*. PMLR, 2009. pp. 567-574. Available from: https://proceedings.mlr.press/v5/titsias09a/titsias09a.pdf
35. BLEI, David M., Alp KUCUKELBIR and Jon D. MCAULIFFE. Variational inference: A review for statisticians. In: *Journal of the American statistical Association*. 2017. pp. 859-877. Available from: https://arxiv.org/pdf/1601.00670.pdf
36. LIU, Haitao, et al. Remarks on multi-output Gaussian process regression. In: *Knowledge-Based Systems*. 2018, vol. 144, pp. 102-121. Available from: DOI: https://doi.org/10.1016/j.knosys.2017.12.034
37. ANSEL, Jason, et al. PyTorch 2: Faster Machine Learning Through Dynamic Python Bytecode Transformation and Graph Compilation. In: *29th ACM International Conference on Architectural Support for Programming Languages and Operating Systems*. ACM, 2024, vol. 2. Available from: https://doi.org/10.1145/3620665.3640366
38. GARDNER, Jacob, et al. GPyTorch: Blackbox Matrix-Matrix Gaussian Process Inference with GPU Acceleration. In: *Advances in Neural Information Processing Systems.* 2018, vol. 31. Available from: DOI: https://doi.org/10.48550/arXiv.1809.11165
39. VAN ROSSUM, Guido and Fred L. DRAKE. *Python 3 Reference Manual*. Scotts Valley, CA. 2009.
40. KLUYVER, Thomas, et al. Jupyter Notebooks – a publishing format for reproducible computational workflows. In: *Positioning and Power in Academic Publishing: Players, Agents and Agendas*. 2016. pp. 87–90.
41. HUNTER John D. Matplotlib: A 2D Graphics Environment. In: *Computing in Science & Engineering*, 2007, vol. 9, no. 3. pp. 90-95
42. HARRIS, Charles R., et al. Array programming with NumPy. *Nature 585*. 2020. pp. 357–362. Available from: DOI: https://doi.org/10.1038/s41586-020-2649-2
43. PEDREGOSA, Fabian, et al. Scikit-learn: Machine Learning in Python. In: *Journal of Machine Learning Research 12.* 2011. pp. 2825–2830.
44. The pandas development team. *pandas-dev/pandas: Pandas*. Zenodo, 2024. Version 2.2.1. Available from: DOI: https://doi.org/10.5281/zenodo.10697587
45. MCKINNEY, Wes. Data Structures for Statistical Computing in Python. In: *Proceedings of the 9th Python in Science Conference*. 2010, vol. 445, pp. 56-61. Available from: DOI: http://doi.org/10.25080/Majora-92bf1922-00a

46. *Joblib* [interactive]. 2008-2018 [accessed 2024-04-24]. Available from: https://joblib.readthedocs.io/en/latest/
47. CORTES, Corinna, and Mehryar MOHRI. Confidence intervals for the area under the ROC curve. In: *Advances in neural information processing systems 17*. 2004. Available from: https://proceedings.neurips.cc/paper/2004/file/a7789ef88d599b8df86bbee632b2994d-Paper.pdf
48. WANG, Haoran, et al. Can multi-label classification networks know what they don't know?. In: *Advances in Neural Information Processing Systems*. 2021, vol. 34, p. 29074-29087. Available from: DOI: https://doi.org/10.48550/arXiv.2109.14162