Optimal Transfer Learning Model for Binary Classification of Funduscopic Images through Simple Heuristics

Rohit Jammula
College Preparatory School,
Oakland CA, 94566
rojammula@gmail.com

Vishnu Rajan Tejus
Department of Computer Science
Stanford University
Stanford, CA 94305
vrt@stanford.edu

Shreya Shankar
Department of Computer Science
Stanford University
Stanford, CA 94305
shreya@cs.stanford.edu

Abstract

Deep learning models have the capacity to fundamentally revolutionize medical imaging analysis, and they have particularly interesting applications in computer-aided diagnosis. We attempt to diagnose fundus eye exams, visual representations of the eye's interior. Recently, a few deep learning approaches have performed binary classification to infer the presence of a specific ocular disease, such as glaucoma or diabetic retinopathy. In an effort to broaden the applications of computer-aided ocular disease diagnosis, we propose a unifying model for disease classification: low-cost inference of a fundus image to determine whether it is healthy or diseased. We use transfer learning models, comparing their "base" architectures and hyperparameters via. a custom heuristic and evaluation metric ranking system. The Xception base model, Adam optimizer, and mean squared error loss function perform best, achieving 90% accuracy, 94% sensitivity, and 86% specificity.

1 Introduction

1.1 Problem Statement

According to a 2016 study done by Human Resources for Health, the global shortage of health care workers will rise to 15 million by 2030 (Liu et al., 2016). Many patients don't realize they have eye disease until their vision suffers irreversible damage. To most effectively fill the void, deep learning algorithms should be widely implemented on fundus eye exam (funduscopic) images.

1.2 Terminology

In a **funduscopy**, medical professionals use ophthalmoscopes to obtain visual representations of the eye's interior (Schneiderman, 1990). Through **transfer learning**, one can import a "base model", remove top layers, replace them with more suitable ones, and alter the input layer size, while maintaining weights that detect overarching features (Bengio, 2012). **Overfitting** is defined as train minus test accuracy, after model training. **Sensitivity** denotes true positive over all positives and **specificity** denotes true negative over all negatives. Positive is diseased and negative is healthy.

For our purposes, **hyperparameters** are defined as external factors more surface-level than model architecture and data augmentation (i.e. epoch number, batch size, optimizers, loss functions). They are easier to tune than more structural factors (MacKay et al.), such as data preparation and model architecture. Our "hyperparameters" are less significant: suited for granular control in later stages.

1.3 Current Research

Several models predict the existence of specific diseases. For instance, this macular degeneration model predicts 95% and 81% accuracy for local and standard datasets (Langarizadeh et al., 2017). A GoogLeNet based glaucoma model overcomes poor image quality (Cerentinia et al., 2018). Diabetic retinopathy models enjoys constant attention and ever-improving results (Carson Lam et al., 2018).

1.4 Purpose

Current models can't replace doctors yet, so a more general model (healthy vs diseased) is best for initial diagnosis. By definition, overarching categories have more data, and therefore better generalization, than its subcategories. Moreover, a simpler healthy vs diseased model would simplify treatment pipelines.

2 Experiments

2.1 Data Acquisition

We must ensure all datasets have comparable numbers of healthy and diseased fundus images, due to divergent imaging techniques. There can be no extraneous visual artifacts, other than indications of disease. Otherwise, extraneous artifacts must be common to both datasets. Not many substantial datasets are released, due to patient confidentiality policies, and even fewer follow these essential prerequisites. Using this information, we imported images from the EYEPACS (Gulshan et al., 2016) diabetic retinopathy dataset and the ORIGA-650 (Zhang et al., 2010) glaucoma dataset. We randomly sampled 3000 EYEPACS images from the normal category, and 987 images from the diabetic retinopathy category. The ORIGA-650 dataset contains 482 normal fundus images and 168 with glaucoma (glaucomatous).

2.2 Data Preparation

We resized all individual images into arrays of size 128x128x3. All values are integers ranging from [0,255] inclusive. We augmented ORIGA images, by creating b different orientations and zooms, with constant background fill. We also created c different noise configurations, except control, shifting each channel value randomly by an integer from -2 to 2 inclusive, but never outside [0,255]. An image can be replicated by a factor of b(c+1). For glaucoma ORIGA images, b=3 and c=1, and for normal ORIGA images, b=4 and c=3. This yielded 482*3*(1+1) = 2892 normal ORIGA images, and 168*4*(3+1) = 2688 glaucomatous ORIGA images. For the final normal dataset, we combined the 2892 normal ORIGA images with all collected normal EYEPACS images to get 5892 total normal images. For the final diseased dataset, we created 2 copies of each of the 987 EYEPACS images (2961 total EYEPACS diabetic retinopathy images) and combined them with the 2688 glaucomatous ORIGA images to get a total of 5649 diseased images. In total, we had 11541 images to work with. Finally, we allocated 6924 for training, 2308 for validation, and 2309 for testing.

2.3 Baseline Model

We use the Xception base model from Keras with randomly initialized weights, negating transfer learning elements. It takes Inception's crucial depth wise convolution to the extreme (Chollet, 2016). This makes Xception sufficiently robust baseline. Comparison to final model shown in 4.1

2.4 Default Settings

Remove top layers. Replace with flatten command, dropout of 0.5, and Dense layer of size 2 with SoftMax activation. Outputs binary probability vector giving predicted label. Unfreeze batch normalization layers if they exist. Resize input layer to match size of inputs.

2.5 Evaluation Metric Ranking System

Models are ranked against one another for each of 5 evaluation metrics, in order of decreasing importance: overfitting, validation accuracy, validation loss, sensitivity, and specificity. Given N different models in a Stage, highest value given rank of 1, and lowest value given rank of N. Ranks in Stage 1 are unrelated to ranks in Stage 2. If there's a tie for first, least parameters wins, to reward portability. Overall score is defined by this equation:

OVERALL SCORE = 3 * (*overfit rank*) + 2 * (N + 1 – *accuracy rank*) + 1.5 * (*loss rank*) + 1 * (N + 1 – *sensitivity rank*) + 0.25 * (N + 1 – *specificity rank*)

2.6 Hyperparameter Preference Justification

We invoke mathematical plausibility principles to explain subtraction from N+1, for overfit and loss ranks. We only seek to maximize desirable metrics such as accuracy, sensitivity, and specificity. The "+1" standardizes overall scores and mostly does not influence rank. This way, the coefficients can be applied properly to each term. Without the "+1", the overfit term, for instance, could end up slightly de-prioritized when compared to the validation accuracy term. Validation accuracy, sensitivity, and specificity are used in variety of literature (Toghi & Grover, 2018). Sensitivity is more important than sensitivity, because the problem is medical in nature: missing a diseased image is significantly worse than flagging a healthy image.

2.7 Stages

- **Stage 1 – Base Model Selection N=17:** We use 17 ImageNet pretrained base models. The default optimizer is rmsprop, and the default loss function is categorical cross-entropy.
- **Stage 2 – Hyperparameter Optimization N=9:** The Keras system randomly initializes starting point, so model structure itself must be altered. Epoch number and batch size are surface-level and may contribute to randomness. **Optimizers** and **loss functions**, however, are more structural and therefore less prone to randomness. We tune optimizers (rmsprop or RMS, Adam, Adagrad) and loss functions (categorical cross-entropy or CCE, mean squared error or MSE, mean absolute error or MAE).

3 Results

3.1 Data Tables

Table 1: Stage 1

Model	Overfitting	Validation acc.	Loss	Sensitivity	Specificity	Rank
Xception	**0.0952**	**0.9008**	**0.3468**	**0.9407**	**0.8603**	**1**
Resnet50	0.0914	0.8925	1.4468	0.9613	0.8227	2
Resnet50V2	0.1296	0.8219	0.792	0.9355	0.7066	16
Resnet101	0.0968	0.8921	1.1475	0.9355	0.848	6
Resnet101V2	0.1165	0.8618	0.7693	0.9226	0.8	10
Resnet152	0.0892	0.8938	1.5941	0.8942	0.8934	3
Resnet152V2	0.1064	0.8826	0.4262	0.9011	0.8638	7
VGG16	0.0699	0.7552	2.2893	0.7498	0.7607	8
VGG19	0.0751	0.7171	2.5212	0.5701	0.8664	13
InceptionV3	0.0890	0.7773	0.457	0.8426	0.7109	5
InceptionResNetV2	0.1004	0.7409	0.6184	0.6578	0.8253	17
MobileNet	0.1202	0.8228	2.1897	0.9733	0.6699	15
DenseNet121	0.0776	0.8193	0.7664	0.6939	0.9467	4
DenseNet169	0.0438	0.6551	2.2782	0.3336	0.9817	11
DenseNet201	0.0323	0.6937	4.342	0.7455	0.641	9
NASNetLarge	0.0363	0.6308	3.2208	0.6784	0.5825	14
NASNetMobile	0.0131	0.5789	2.9431	0.2614	0.9013	12

Table 2: Stage 2 - Xception Base Architecture

Model	Overfitting	Val accuracy	Loss	Sensitivity	Specificity	Rank
RMS, CCE	0.1395	0.8098	0.6115	0.9587	0.6588	9
RMS, MSE	0.1118	0.8544	0.1195	0.8521	0.8568	3
RMS, MAE	0.0890	0.8011	0.2032	0.7325	0.8707	7
Adam, CCE	0.1097	0.8817	0.3780	0.9251	0.8399	4
Adam, MSE	**0.0846**	**0.9015**	**0.0925**	**0.9413**	**0.8560**	**1**
Adam, MAE	0.0783	0.8219	0.1827	0.8667	0.7222	2
Adagrad, CCE	0.1040	0.8528	0.3540	0.8418	0.8629	5
Adagrad, MSE	0.1158	0.8133	0.1416	0.9036	0.7213	8
Adagrad, MAE	0.0787	0.7595	0.2440	0.7825	0.7362	6

3.2 Results Verification

The Xception base structure, Adam optimizer, and MSE loss function produce the best results. To verify validation results, we note the absolute value of the differences between the validation and testing versions of accuracy, sensitivity, and specificity. They represent their own versions of "overfitting" separate from the one defined in 1.2. The absolute values of the differences are extremely small, indicating that the model as a whole generalizes to the test set as well as it does with the validation set.

|Accuracy Difference|	|Sensitivity Difference|	|Specificity Difference|
0.0267	0.0461	0.0001

4 Conclusion

4.1 Comparison with Baseline

Our final model performs substantially better than the baseline with randomly initialized weights. We cannot confirm this by comparing overall score results, because the formula depends on ranks which would be infeasible for only two models. So we must compare both models through each evaluation metric individually.

Model	Overfitting	Validation Accuracy	Loss	Sensitivity	Specificity
Final Model	0.0846	0.9015	0.0925	0.9413	0.8560
Baseline Model	0.0223	0.5841	0.6948	0.4996	0.6699

4.2 Analysis of Results and Other Notable Trends

Xception succeeds due to constant repetition of depth wise convolutions. All V2 variations of ResNet perform worse than predecessors. Some models with overall poor performance have low overfitting.

4.3 Implications for Deep Learning Research and Healthcare

Hopefully, researchers consider pre-trained ImageNet base models. Data collection method illuminates model bias question. Two-stage selection process and novel heuristic equation can generalize to other transfer learning applications. Healthy vs diseased model streamlines diagnoses for general public.

4.4 Future Work

We could use a more robust Capsule Net baseline to evaluate spatio-temporal relationships among model's features set (Sabour et al., 2017). The CodaLab framework can be used for improved reproducibility. We can incorporate image segmentation to create tally for binary classification. We can gather more diverse datasets, and construct knowledge base for "Medical ImageNet". Finally, we could perform ablation studies to detect classification discrepancies.

References

Bengio, Y. Deep learning of representations for unsupervised and transfer learning. In *Proceedings of ICML workshop on unsupervised and transfer learning*, pp. 17–36, 2012.

Carson Lam, D. Y., Guo, M., and Lindsey, T. Automated detection of diabetic retinopathy using deep learning. *AMIA Summits on Translational Science Proceedings*, 2018:147, 2018.

Cerentinia, A., Welfera, D., d'Ornellasa, M. C., Haygertb, C. J. P., and Dottob, G. N. Automatic identification of glaucoma sing deep learning methods u. In *MEDINFO 2017: Precision Healthcare Through Informatics: Proceedings of the 16th World Congress on Medical and Health Informatics*, volume 245, pp. 318. IOS Press, 2018.

Chollet, F. Xception: Deep learning with depthwise separable convolutions. *CoRR*, abs/1610.02357, 2016.

Gulshan, V., Peng, L., Coram, M., Stumpe, M. C., Wu, D., Narayanaswamy, A., Venugopalan, S., Widner, K., Madams, T., Cuadros, J., et al. Development and validation of a deep learning algorithm for detection of diabetic retinopathy in retinal fundus photographs. *Jama*, 316(22): 2402–2410, 2016.

Langarizadeh, M., Maghsoudi, B., and Nilforushan, N. Decision support system for age-related macular degeneration using convolutional neural networks. *Iranian Journal of Medical Physics*, 14(3):141–148, 2017.

Liu, J. X., Goryakin, Y., Maeda, A., Bruckner, T., and Scheffler, R. *Global health workforce labor market projections for 2030*. The World Bank, 2016.

MacKay, M., Vicol, P., Lorraine, J., Duvenaud, D., and Grosse, R.

Sabour, S., Frosst, N., and Hinton, G. E. Dynamic routing between capsules. *CoRR*, abs/1710.09829, 2017.

Schneiderman, H. The funduscopic examination. 1990.

Toghi, B. and Grover, D. MNIST dataset classification utilizing k-nn classifier with modified sliding window metric. *CoRR*, abs/1809.06846, 2018. URL `http://arxiv.org/abs/1809.06846`.

Zhang, Z., Yin, F. S., Liu, J., Wong, W. K., Tan, N. M., Lee, B. H., Cheng, J., and Wong, T. Y. Origa-light: An online retinal fundus image database for glaucoma analysis and research. In *2010 Annual International Conference of the IEEE Engineering in Medicine and Biology*, pp. 3065–3068. IEEE, 2010.