Disease Prediction of Diabetic Retinopathy from Image Data using Deep Learning

Varuni Sarwal, Rosemary He, Hritik Bansal

1 Introduction

Diabetic Retinopathy is a retina disease caused by diabetes mellitus and is the leading cause of blindness globally[1]. It occurs when changes in blood glucose levels lead to swelling of retinal blood vessels(macular oedema) and leaking of fluid into the rear of the eye[1]. If not diagnosed and treated in a timely manner, diabetic retinopathy will cause severe vision damage and even blindness[1]. Therefore, early detection and treatment are necessary in order to delay or avoid such deterioration or vision loss [1]. Our group aim to build and optimize a machine learning model that can be used to predict progression of diabetic retinopathy from retina images. First, we performed MixUp and CutMix to augment the dataset. We then ran 3 popular convolutional neural network (CNN) models on the augmented data to find the optimal model. We find that the VGG16 architecture with cutmix training set outperforms other models we have tried. Pretrained model also outperforms its untrained counterparts in all scenarios. Additionally, we also implement a popular contrastive learning framework to solve this challenge but do not find any significant improvements in the performance due to its unsupervised nature. We think using supervised contrastive learning is a feasible future direction. Our code for the project is available at https://github.com/Addicted-to-coding/Diabetic_Retinopathy_DL.

2 Data Processing and Augmentation

For the dataset, we used the APTOS 2019 Blindness Detection dataset, which contains 3,662 images and is publically available on Kaggle. The dataset was collected by Aravind Eye Hospital in India, in hope to facilitate clinical diagnosis of Diabetic Retinopathy[1]. While the dataset has many samples, one limitation is a large class imbalance, where the severe cases have only 200-300 images compared to 1,800 images for healthy individuals[1]. In addition, the dataset is subject to variances such as camera settings, light exposures and noise[1]. Due to memory issues, we cropped the original images to 160×160. We decided not to do any further imaging pre-processing as we implemented two data augmentation techniques later on to compare the results. In addition to the original dataset, we performed data augmentation using two popular techniques: MixUp and CutMix.

2.1 MixUp

Mixup [2] was proposed to train deep neural networks where additional samples are generated during training by convexly combining random pairs of images and their associated labels. The mathematical formula is as follows:

$$\begin{aligned} \tilde{x} &= \lambda x_i + (1-\lambda)x_j \\ \tilde{y} &= \lambda y_i + (1-\lambda)y_j \end{aligned} \tag{1}$$

where (x_i, y_i) and (x_j, y_j) are two examples drawn at random from the training data and $\lambda \in [0, 1]$. Mixup has shown to be very effective method of data augmentation on a number of image classification tasks. It has also been shown that Mixup improves the calibration and predictive uncertainity for deep neural nets [5].

2.2 Cutmix

CutMix generates a new training sample $(\tilde{x}, \tilde{y})$ by combining two training samples (x_i, y_i) and (x_j, y_j)[3]. The formula is as follows:

$$\begin{aligned} \tilde{x} &= M \odot x_i + (1-M) \odot x_j \\ \tilde{y} &= \lambda y_i + (1-\lambda)y_j \end{aligned} \tag{2}$$

where $M \in \{0, 1\}^{W*H}$ is a binary mask indicating where to drop out and fill in from the two images, **1** is a binary mask filled with ones, and $\odot$ is element-wise multiplication[3].

Below is an example of CutMix, where the first two images are combined to produce the new image on the right.

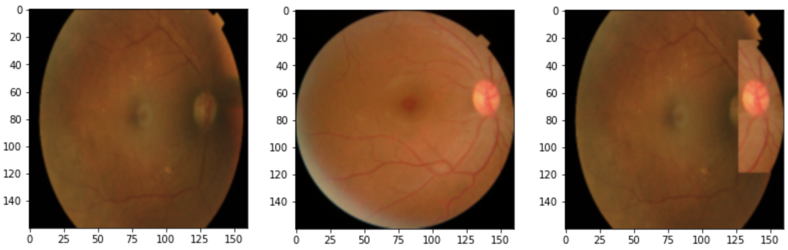

Figure 1: Example of CutMix

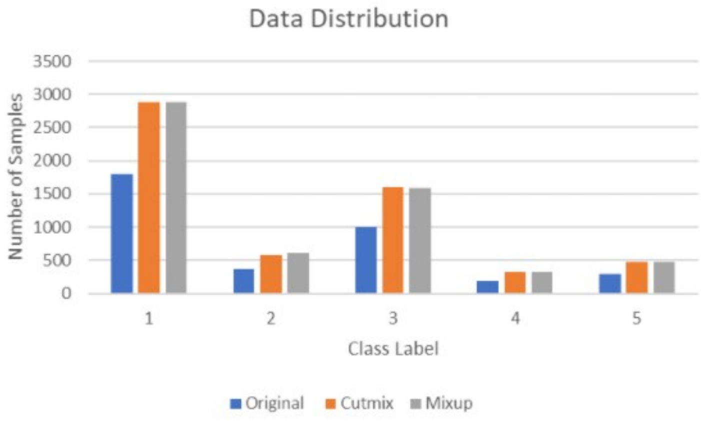

Figure 2: Data Distribution

Scratch									
Models	Original			Cutmix			Mixup		
	Accuracy	Recall	Precision	Accuracy	Recall	Precision	Accuracy	Recall	Precision
ResNet18	74.04	74.38	75.86	75.3	73.84	71.51	74.21	74.93	67.58
VGG16	72.4	73.29	60.2	72.39	71.93	58.83	72.39	72.75	60.49
AlexNet	72.77	74.11	67.95	73.93	73.29	64.96	72.03	70.84	63.37
Finetuning									
Models	Original			Cutmix			Mixup		
	Accuracy	Recall	Precision	Accuracy	Recall	Precision	Accuracy	Recall	Precision
ResNet18	81.29	82.56	82.15	82.29	82.01	81.74	82.74	80.92	80.23
VGG16	81.2	83.65	85.15	**83.56**	**85.55**	**85.8**	82.83	83.1	83.23
AlexNet	81.47	81.47	81.15	79.38	77.38	77.76	79.47	79.56	79.04

Table 1: Summary of the results. Models are trained using Mixup and Cutmix data augmentations.

Simclr Models	Accuracy
Resnet (scratch)	74.93
Resnet (pretrained)	79.84

Table 2: SimCLR Results.

Both mixup and cutmix increased samples sizes in all classes as shown in Figure 2. However, it did not address the issue of class imbalance; in future works, upsampling techniques such as SMOTE can be implemented to address the issue.

3 Models

For modeling, we implemented 3 popular Convolutional Neural Network (CNN) architectures: ResNet18, Alexnet and Vgg16[4].

3.1 ResNet18

ResNet makes use of shortcut connections to solve the vanishing gradients problem. It consists of Convolutional layers with filters of size 3x3 and has around 11 mn trainable parameters.

3.2 Alexnet

Alexnet was the first deep CNN to top the Imagenet challenge 2012. It consists of 5 Convolutional layers and 3 Fully Connected (FC) layers. The activation used is the Rectified Linear Unit (ReLU), and it has a total of 62 mn trainable parameters.

3.3 VGG16

VGG contains conv kernels of size 3x3 and maxpool kernels of size 2x2 with a stride of two. It has a total of 138 million parameters. VGG incorporates 1x1 convolutional layers to make the decision function more non-linear without changing the receptive fields.

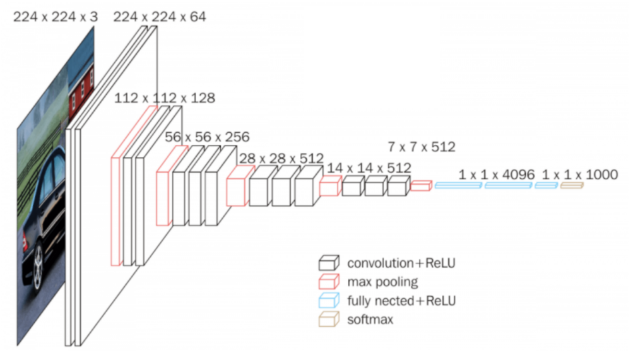

Figure 3: VGG

4 SimCLR

In this work, we also implement a recently popularized unsupervised method of contrastive learning for visual representations [7]. We use Resnet18 [§3.1] as the base encoder network, and have a single layer MLP as the projection head. We train this pipeline solely using a normalized temperature-scaled cross entropy loss. Once the contrastive framework is trained, we throw away the projection head and train a single layer MLP as a classification head for the severity detection. Details of our implementation are available in our repo.

5 Evaluation

In our experiments, we trained each model with the original, cutmix and mixup dataset 3 times, each from scratch and with pretrained set as true. We then took the best results out of the 3 runs and reported it as the final evaluation of each model. As accuracy is a less reliable metric when there is a big class imbalance, we will be evaluating mainly based on recall and precision rate. Out of all models, VGG16 with cutmix augmentation returns the best result with a recall rate of 85.55 and precision of 85.8 [Table 1]. On average, models with pretrained set as true outperforms their counterparts from scratch. Since the landscape of the loss function is non-convex, transfer learning using pretrained models can provide 3 main advantages. The first is a higher start. The initial skill (before refining the model) on the source model is higher than it otherwise would be. Second, a steeper slope. The rate of improvement of skill during training of the source model is steeper than it otherwise would be. Lastly, a higher asymptote. The converged skill of the trained model is better than it otherwise would be.

Results from Table 2.2 suggest that unsupervised contrastive learning does not outperform the Mixup and CutMix strategies. There can be multiple reasons for the observed behaviour. It has been shown that SimCLR can be sensitive to hyperparameter tuning and we do not perform any of that. Unsupervised contrastive learning might be forcing images from the same class to be far away from each other in the representation space.

6 Future work

For data preprocessing, we could try upsampling techniques to alleviate the class imbalance. For modeling, we could try other architectures such as GoogleNet and Inception. In addition, ensemble methods have been proposed for better results. For SimCLR, we could try a recently published work that advocates using supervised contrastive learning [8] when sample labels are available.

References

1. Tsiknakis, N., Theodoropoulos, D., Manikis, G., Ktistakis, E., Boutsora, O., Berto, A., Scarpa, F., Scarpa, A., Fotiadis, D.I. and Marias, K., 2021. Deep learning for diabetic retinopathy detection and classification based on fundus images: A review. Computers in Biology and Medicine, p.104599.

2. Zhang, H., Cisse, M., Dauphin, Y.N. and Lopez-Paz, D., 2017. mixup: Beyond empirical risk minimization. arXiv preprint arXiv:1710.09412.

3. Yun, S., Han, D., Oh, S.J., Chun, S., Choe, J. and Yoo, Y., 2019. Cutmix: Regularization strategy to train strong classifiers with localizable features. In Proceedings of the IEEE/CVF International Conference on Computer Vision (pp. 6023-6032).

4. LeCun, Y. and Bengio, Y., 1995. Convolutional networks for images, speech, and time series. The handbook of brain theory and neural networks, 3361(10), p.1995.

5. Thulasidasan, S., Chennupati, G., Bilmes, J., Bhattacharya, T. and Michalak, S., 2019. On mixup training: Improved calibration and predictive uncertainty for deep neural networks. arXiv preprint arXiv:1905.11001.

6. Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., Liu, T., Wang, X., Wang, G., Cai, J. and Chen, T., 2018. Recent advances in convolutional neural networks. Pattern Recognition, 77, pp.354-377.

7. Chen, T., Kornblith, S., Norouzi, M. and Hinton, G., 2020, November. A simple framework for contrastive learning of visual representations. In International conference on machine learning (pp. 1597-1607). PMLR.

8. Khosla, Prannay, et al. ”Supervised contrastive learning.” arXiv preprint arXiv:2004.11362 (2020).