Name: Debayan Mitra

Email address: debayanmitra1993@gmail.com

Contact number: +91-9892698218, +91-7981030176

Anydesk address: **651 757 144**

Date: 30th April 2020

Self Case Study -2: APTOS 2019 Blindness Detection

PROJECT ABSTRACT

Here is the **arXiv.org** research paper - https://arxiv.org/pdf/2003.02261.pdf that I want to implement. This paper obtains a sensitivity and specificity score = 0.99, Quadratic weighted Kappa = 0.925

Sharing below the summary pointers from the research paper (Section wise - 7 sections) below.

---

1. INTRODUCTION

- Multi-stage approach to transfer learning is used.
- This is written by 3 people who took part in the APTOS 2019 blindness detection competition, Link to competition - https://www.kaggle.com/c/aptos2019-blindness-detection/
- Sensitivity & Specificity scores = **0.99**, Quadratic weighted Kappa = **0.925**, Leaderboard = **Top 2%**
- 5 class Classification problem, 3600 images in training data given.

---

2. Related Work

- Traditional methods used for Diabetic retinopathy detection in medical literature included feature extraction from Images - fit an SVM classifier ; applying PCA to images - apply DTs, NB classifiers etc. Best results obtained ~ 75%
- Other similar research in diabetic retinopathy detection also involved CNN architectures with transfer learning - ResNet50, InceptionNetV3, DenseNets etc. CNN did a better job in extracting relevant features from images - Best results obtained ~ 81%

3. Problem Statement

- **Datasets used :-**
    - A previous competition dataset was used (Diabetic retinopathy detection, 2015)- https://www.kaggle.com/c/diabetic-retinopathy-detection/overview/timeline contained **35,216** images.
    - Indian Diabetic Retinopathy Image Dataset (IDRiD) (Sahasrabuddhe and Meriaudeau, 2018) = **413** images used.
    - MESSIDOR dataset (Google Brain,2018) dataset
    - The full dataset consists of **18590** fundus photographs, which are divided into **3662** training, **1928** validation, and **13000** testing images by organizers of Kaggle competition
    - All Datasets had a similar distribution of output classes, a fundamental property of this type of data.
- **Evaluation Metric :-**
    - Main Evaluation metric is **Quadratic weighted kappa (Cohen's kappa)**
    - Along with the Kappa score, other metrics used are - **macro F1- score**, **accuracy**, **sensitivity**, **specificity** on holdout dataset of 736 images taken from APTOS2019 training data.

---

4. Method

- The diabetic retinopathy detection problem can be viewed from several angles: as a **classification problem**, as a **regression problem**, and as an **ordinal regression** problem (Ananth and Kleinbaum, 1997). This is possible because stages of the **disease come sequentially**

- **Preprocessing methods :-**
    - Image cropping + resizing was used . Spurious correlations were present between the output class label and several image meta-features, e.g resolution, crop type, zoom level, or overall brightness. **(correlation heatmap image below)**
    - Image Augmentations were used - *(optical distortion, grid distortion, piecewise affine transform, horizontal flip, vertical flip, random rotation, random shift,*

random scale, a shift of RGB values, random brightness and contrast, additive Gaussian noise, blur, sharpening, embossing, random gamma, and cutout)

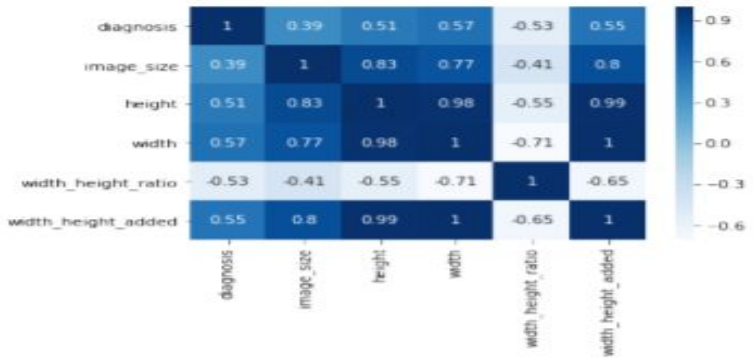

Figure 4: Spurious correlations between meta-features and diagnosis.

- **Training Process (3 stage training) :-** (diagram below)

Figure 5: Three-head CNN structure.

- **Stage 1 - Pre Training**
    - Imagenet weights used, Random weight initialization, 20 epochs, SGD with Cosine annealing LR scheduler on the **2015 competition data**.
    - Every head was minimizing its own loss function - **crossentropy** for classification head, **binary cross-entropy** for ordinal regression head, and **mean absolute error** for regression head.
    - After pretraining, encoder weights were used as initialization for subsequent stages
- **Stage 2 - Main Training**
    - Main training is performed on 2019 data, IDRID, and MESSIDOR combined datasets
    - Loss Functions are changed now, **Focal Loss** (Lin et al., 2017) for classification head, **binary Focal Loss** (Lin et al., 2017) for ordinal regression head and **mean-squared error** for regression head
    - 75 epochs, Rectified Adam optimizer, Cosine Annealing LR sceduler
    - Freeze training of the encoder for five epochs while training heads
    - Live monitoring of TSNE plots - features generated by encoder
- **Stage 3 - Post Training**
    - Fit the linear regression model to outputs of different heads

- **Regularization Used :-**
    - Label smoothing is used for regression and classification heads
    - Random uniform noise is added to Discrete target labels. This reduces the chances of wrong labelling and also helps in generalization of models better
- **Final Ensemble Model :-**
    - 3 Model ensembles were used - EfficientNetB4, EfficientNetB5, SE-ResNeXt50
    - Best performing solution is an ensemble of 20 models (4 architectures x 5 folds) with test-time augmentations generating 200 predictions per image.
    - Predictions were averaged with **0.25 trimmed mean** to reduce model variance and reduce overfitting.
    - The team used the Catalyst framework based on PyTorch with GPU support. Evaluation of the whole ensemble was performed on Nvidia P100 GPU in 9 hours, processing 2.5 seconds per image.

---

5. Results

Model	QWK	Macro F1	Accuracy	Sensitivity	Specificity
EfficientNet-B4	0.965	0.811	0.903	0.812	0.976
EfficientNet-B5	0.963	0.815	0.907	0.807	0.977
SE-ResNeXt50 (512x512)	0.969	0.854	0.924	0.871	0.982
SE-ResNeXt50 (380x380)	0.960	0.788	0.892	0.785	0.974
Ensemble (mean)	0.968	0.840	0.921	0.8448	0.981
Ensemble (trimmed mean)	0.971	0.862	0.929	0.860	0.983
Ensemble (trimmed mean, binary classification)	0.981	0.989	0.986	0.991	0.991

Table 1: Results of experiments and metrics tracked, **without using TTA**.

Model	QWK	Macro F1	Accuracy	Sensitivity	Specificity
EfficientNet-B4	0.966	0.806	0.902	0.809	0.977
EfficientNet-B5	0.963	0.812	0.902	0.807	0.976
SE-ResNeXt50 (512x512)	0.971	0.853	0.928	0.868	0.983
SE-ResNeXt50 (380x380)	0.962	0.799	0.899	0.798	0.976
Ensemble (mean)	0.968	0.827	0.917	0.828	0.980
Ensemble (trimmed mean)	0.969	0.840	0.919	0.840	0.981
Ensemble (trimmed mean, binary classification)	0.986	0.993	0.993	0.993	0.993

Table 2: Results of experiments and metrics tracked, **with using TTA**.

---

6. Model Interpretation

- A method called **SHAP (Shapley Additive exPlanations)**. This method makes possible to visualize features that contribute to the assessment of the output label.
    - https://www.kaggle.com/dimitreoliveira/diabetic-retinopathy-shap-model-explainability/ - This kernel explains how to interpret CNN models
    - https://www.kaggle.com/ratthachat/aptos-augmentation-visualize-diabetic-retinopathy/ - This kernel labels certain features in the image that detect outputs. (for model interpretability)

---

7. Conclusions

- By using Ensemble models and multi-task learning (Classification, Regression) and since data size is small - it is ensured that model does not overfit and generalizes well on unseen data - hence **multi-task learning** is used here.
- There is scope for hyperparameter optimizations on these models for improvements

---