Presentation notes

Every 11th person in the world suffers from diabetes mellitus which is a disorder of sugar metabolism, the prevalence of which is expected to rise to every 10th person by 2040.

Diabetics are 25 times more likely to suffer from sight loss resulting from DR which is a major long-term microvascular complication and the leading cause of blindness in USA.

7.7 million people aged over 40 in the U.S have DR and there is a larger prevalence in developing countries.

WHO & American academy of Ophthalmology recommend eye exams once a year for diabetics. However poor or isolated communities cannot afford such frequent consultation. Around 10% of diabetic people live in countries without eye care professionals.

This makes early detection critical for limiting progression. Reliable automated screening for referral can take the strain off the limited number of health care professionals and help detect DR earlier.

The authors partnered with phelcom technologies with the goal of using their solution inside phelcoms product which is to be a portable machine capable of capturing a retinal image that the authors deep learning solution can then classify as either needing referral or not needing referral.

Current solutions to DR referral focus on lesions detection using handcrafted feature engineering.

Solution is a CNN. Inspired by two competition runners up. o\_O & VGG-16

Architecture resembles VGG-16 in terms of arrangement of pooling and conv layers. Whereas the fully-connected stage is more inspired by the o\_O solution.

The authors modelled the problem as a classification rather than a regression. Referral rather than severity.

Convolutional layer neurons/filters convolve over input. Then after element wise multiplication with the receptive field the result is summed and passed through an activation function (*Leaky ReLU - accelerates the convergence of the gradient in comparison with conventional activation functions*), the result is then outputted to a feature/activation map (of reduced dimensions).

Strides can be applied to further reduce the dimensionality of the input. (done in the 1st & 3rd layers)

Pooling layers merge neighbouring features into one. (Downsampling)

* Very small receptive field (3 X 3)
* Pooling layers separate 2 or 3 convolutional layers
* Convolutional layers start at 32 filters and double after each pooling layer
* Stride in 1st & 3rd convolutional layers
* Leaky RELU (accelerates the convergence of the gradient in comparison with conventional activation functions)

Fully connected layer is a similar traditional neural network using the input from the CNN to classify the image. The 1024 units of the hidden dense layers employ dropout with a probability of 0.5. (Dropout tries to prevent a model overfitting) The last decision layer has 2 units outputting the probibilites of the 2 classes.

Once we have the gradient of the loss function we have a choice of how we adjust the weights. Typically SGD is used, however you can also use, classic momentum the Accelerates GD by accumulating a velocity in the direction of the downward gradient. Accelerates convergence in areas of low curvature. Nesterov momentum computes a partial update to the position allowing for a more responsible stable change in velocity.

Explain terms.

The learning rate decreases over learning epochs.

L2 regularisation (ridge regression): Is to help prevent overfitting. Adds the squared magnitude of the weight to the loss function. Penalising the weights of the nodes more. + 0.0005 \* sum(weights^2)

The loss function used here is cross entropy because the problem is a (binary) classification not a regression (for which mean squared error would be used).  
“seek a set of model weights that minimize the difference between the model’s predicted probability distribution given the dataset and the distribution of probabilities in the training dataset. This is called the cross-entropy.”

Binary CE = - ylog(p) + (1-y)log(1-p)

Operations/perurbations performed before submission to the network. Operations are done by choosing a random variable from a range for each operation. Example: between 0-360 for rotations, translations between 40 and -40. # of perturbed versions of each class depend on the balance weights of that class that is inversely proportional to the number of images(inputs) for each class.

Pertubations

* + Geometric (Zoom, rotations, cropping.. etc)
  + Photometric (Contrast enhancements, histogram equalisations)

Keep classes balanced while inflating training set. Proportions the same because augmentations applied to all images.

Training simplified versions of the CNN that require less training samples and then using learned parameters as a starting point for the next stages

Pseudo-random because same perturbations are always applied to all images.

Feature vectors are extracted by CNN for each image (version) final feature vector for an image is the mean and standard deviation of the 20 feature vectors of each of its versions.

Extract features in last pooling layer of CNN.

I THINK WE TRAIN THE CNN NETWORK USING THE FULLY CONNECTED LAYER BUT THEN ADOPT DIFFERENT CLASSFIIERS AFTER FEATURE EXTARCTION TO GET BETTER RUNNING EFFECTIVNESS

Datasets collected at different times, in different hospitals and with different cameras.

Reiterate data augmentation was needed because number of parameters much higher than the number of available training images.

CV is balanced by class (stratified CV).

Messidor-2:

Graded by specialists according to ICDR severity scale.

DR2:

Referral labels provided for 435 by experts

EyePACS:

* + 35,126 for images training & 53,576 for testing
  + Size ranges from 320 x 211 to 5184 x 3456 pixels
  + Includes both left and right eyes graded by severity
  + Converted labels from Severity to Referral Necessity

---

Here we are comparing the results of the previously mentioned progressive improvements. The result is Area achieved under the **receiver operating characteristic curve.**

This is using the train-test validation protocol solely on the Kaggle dataset.

The questions on the side have YES as an answer to all because the addition of each technique in turn provides better effectiveness as shown on the plot.

*Please note o\_O is better but it an ensemble of six classifiers. Ours is one better expert classifier.*

Next slide we see that our one does better for cross-dataset validation, which is more realistic of the real world because different datasets have very different acquisition conditions and it shows our algorithm generalises well to new data.

Here we show the results of training on the Kaggle dataset but then testing on DR2 and Messidor-2. This is the Cross-dataset validation procedure.

Extracting features (via Robust feature-Extraction Augmentation) provides us with flexibility to choose different machine learning algorithms. We choose to use A Neural Network and Random Forest.

Using the best solution from the previous section trained on Kaggle data (exploiting per-patient analysis only when we have access to images of both eyes)

NN:

2 layers 32 units, 3rd layer 2 units. ReLU activation.

Layers intercalated by feature pool layers. Trained the network for 100 epochs using Adam Optimizer (adaptive learning rate **optimization** algorithm)

RF:

Extensive grid search for hyper parameters maximising AUC. Trying 50-300 estimators/trees and gini/entropy criterion.

All experiments used 200 or 300 and entropy

Here we train CNN on Kaggle and use it as a feature extractor for DR and messidor-2. Then run the classifiers on the extracted features???

These results corroborate the hypothesis that it is possible to train a robust data-driven solution to precisely pinpoint diabetic retinopathy referral needs, independently of operators and camera settings of the training set of images.

Note: one of the previous leading solutions run against the Messifor-2 dataset was by Abramoff et el. It had 98% AUC after adopting CNNs however ours had equally remarkable results at 98.2% --

Therefore reinforcing that detecting DR lesions is not essential for a reliable and effective DR screening. NOTE ALSO we don’t use messidor-2 data to train or omptimse our CNN that is done by Kaggle data this shows robustness of the method.

Adapted the o\_O solution for a binary classification problem and evaluated its efficiency and effectiveness.

Recall o\_O is an ensemble of six methods trained with features extracted from 2 CNNs.

Performed the tests using one “GeForce GTX TITAN X”.

Simulated real time diagnostic environment screening 50 patients (100 images)

Time is time to:

* Loading all libraries
* Loading parameters of CNN
* Pseudo-augmenting input images
* Inference of higher probability among eyes

Memory is:

* Disk space for CNN params
* 2 hidden layer neural networks in memory

Our method has huge improvements in space and time efficiency. And also slight improvements in classification effectiveness.

Idea of transfer learning comes from the idea that many deep neural networks trained on natural images learn similar features: textures, corners, edges, colour blobs… etc.

2 types=>

Feature Extraction: Freeze the CNN and use it to extract features

Fine Tuning: Freeze the former high level layers and fine tune the later layers

Source problem: Severity assessment, Target Problem: Referral Assessment

Here we train a CNN of the same architecture to work with to assess severity of DR (five outputs in the decision layer instead of 2) we also use our previously mentioned improvements (data aug, multi-res ..etc)

Fine tuning is better. With RF

On the left provide decision for each eye

On the right provide decision based on both eyes. And out performs as you would imagine.

Same here: Fine tuning is better. With RF

Although these results confirm that patient based is better than per image.