Ensemble Learning for Detection of Diabetic Retinopathy

David T. Butterworth, Shohin Mukherjee and Mohit Sharma
Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213
dbutterw@andrew.cmu.edu
shohinm@andrew.cmu.edu
mohits1@andrew.cmu.edu

1 Introduction

Diabetes Mellitus is a group of metabolic diseases related to elevated sugar levels in the blood. One of the long-term affects is Diabetic Retinopathy, which is damage to the retina that leads to blindness. Diabetic retinopathy is the leading cause of vision impairment and blindness among working-age adults [1] and affects up to 80 percent of patients who have had diabetes for more than 20 years [2].

The first stage of diabetic retinopathy is called “non-proliferative", where the patient experiences no noticeable symptoms. However microaneurysms (blood-filled bulges) can be seen in the tiny artery walls of the fundus (the back of the eye) by dilating the eye and looking through the pupil using a microscope. Therefore it is critical that diabetic patients have regular eye exams to detect retinopathy before they experience vision loss [3].

Traditionally patients must visit an ophthalmologist who will use expensive, specialized equipment to examine the eye and provide a diagnosis. However, improved technology means that fundus images can be extracted with a fundus microscope accessory for $100. We propose to reduce the workload on medical specialists by automatically classifiying fundus images using machine learning.

Retinal images are classified depending on the prevalence of specific abnormalities (see Fig. 1), in 5 classes ranked from non-proliferate diabetic retinopathy (NPDR) through to proliferate diabetic retinopathy (PDR):

- Healthy
- Mild NPDR (At least 1 microaneurysm)
- Moderate NPDR (Multiple microaneurysms, hemorrhages, and exudates)
- Severe NPDR
- PDR (Neovascularization)

2 Related Work

All the recent work in automatic detection of diabetic retinopathy has used deep, convolutional neural networks (CNN). Pratt et al. [5] use a CNN with data augmentation to classify 5 classes of retinopathy on the Kaggle dataset of 80,000 images. They achieved a sensitivity score of 95% on the dataset with accuracy of 75% on 5,000 validation images. Colas et al. [6] described the work of start-up company DreamUp Vision in classifying the same dataset. They achieved an area under the receiver operating characteristic curve (AUROC) of 0.946 with 96.2% sensitivity.

Earlier work using machine learning to diagnose diabetic retinopathy has used classifiers on top of manually-designed feature detectors to measure the blood vessels and the optic disc, and to count

30th Conference on Neural Information Processing Systems (NIPS 2016), Barcelona, Spain.

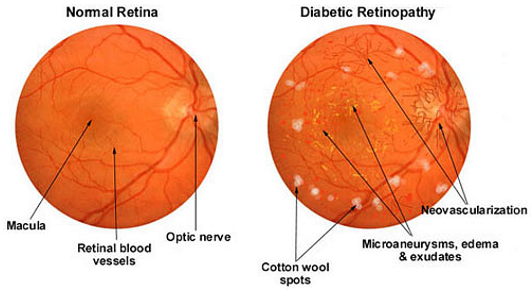

Figure 1: Fundus image of a healthy eye (left) and proliferate retinopathy (right) [4]

the presence of abnormalities such as red lesions, microaneurysms, hard exudates, hemorrhages and cotton wool spots. Roychowdhury et al. [7] developed a 3-stage hierarchical architecture using AdaBoost to select the 30 top features out of 78. The 1st stage enhances the image and detects the optic disc, vasculature and red lesions. Stage 2 classifies the lesions as either cotton wool spots, hard exudates, microaneurysms or hemorrhages. Stage 3 counts the features and assigns one of 5 class labels. They achieved 100% sensitivity, 53.16% specificity, and 0.904 AUC.

In a similar approach, Acharya et al. 2009 [8] used features of blood vessel area, exudes, hemorrhages, microaneurysms and image textures with an SVM to achieve an accuracy of 86%, sensitivity of 82% and specificity of 86%. This is in comparison to the group's earlier work [9] that focused on using the used the area and perimeter of the RGB components of the blood vessels and a neural network to achieve an accuracy of 84%, sensitivity of 92% and specificity of 100%. They also investigated a simpler approach [10] that did not use retinopathy-specific features. Using higher order spectra (HOS) features they achieved an accuracy of 82%, sensitivity of 83% and specificity of 89%. Nayak et al. [11] performed 3-class classification using features of blood vessels, exudates and image texture, to achieve an accuracy of 94%, sensitivity of 90% and specificity of 100%.

3 Dataset

3.1 Data

There are multiple databases of fundus images available, including DIARET DB0 [12], DIARET DB1 [13], MESSIDOR [14] , and Kaggle [15].

We used two datasets:

- Messidor dataset: 1,200 images, with 19 extracted features
- Kaggle dataset: 35,000 images, very high-resolution

We used Messidor because we had the extracted features for these images. Also the image files are smaller and easier to manage. However, there was not enough images to train a CNN (Convolutional Neural Network). Therefore, we used Kaggle because it contains a large number of images. But these are very high-resolution which meant that every step - including maintaining files, pre-processing, cropping, filtering, and training the network - took much longer than we expected.

We planned to further validate our classification system by running it on the opposite dataset. To the best of our knowledge, no one has attempted this for diabetic retinopathy. Unfortunately we ran out of time.

3.2 Image Features

The deep learning methods we used either had pre-trained (generic) image features or we learned the deep image features as part of end-to-end training of the CNN. For the classical machine learning methods we tried, the 19 features extracted from the Messidor dataset are shown in Table 1.

Table 1: Image features

q:	0 = bad quality, 1 = sufficient quality
ps:	0 = retina pre-screen ok, 1 = severe retina abnormality
ma.a - ma.f:	No. of micro-aneurysms, with confidence 0.5 - 1.0
ex.a - ex.h:	No. of exudates, with confidence 0.3 - 1.0
dd:	distance from macula to optic disc
dm:	diameter of the optic disc
amfm:	0/1 result of 39 features from green channel

3.3 Feature Analysis

We analyzed the 19 features by visually comparing the distribution of various feature sub-sets (Fig. 2), as well as after performing PCA (Principal Component Analysis), ICA (Independent Component Analysis) and t-SNE (t-distributed Stochastic Neighbor Embedding) [16].

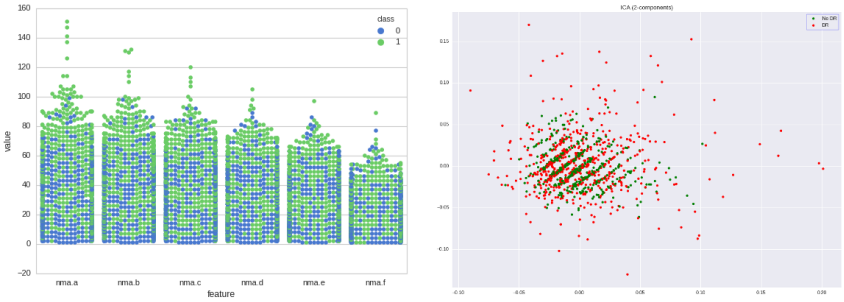

Figure 2: (left) 0/1 distribution of ma.a - ma.f features, (right) ICA decomposition of same features

4 Methods

4.1 "Classic" Ensemble Learning

The idea of ensemble learning [17] is to achieve a higher classification accuracy by combining the results of several classifiers that perform well with certain features or sub-sets of the data.

First we trained various classifiers on the 19 image features to separate the DR (Diabetic Retinopathy) and No-DR (healthy) classes:

- Multinomial Naive Bayes
- Gaussian Naive Bayes
- k-NN
- AdaBoost with Decision Tree Stumps
- Decision Tree
- Random Forest of Trees
- SVM (Linear kernel)
- SVM (Gaussian kernel)

The motivation behind choosing these classifiers was that previous work [11; 18] had used them with excellent results. To select the sub-set of images features (between 2 to 19) and the classifier parameters (k, alpha, gamma) that gave the best classification result, we used the method of forward iterative search [17] over the space of variables. To improve the computation time, the search was conducted at a coarse discretization of possible parameter values, then the results examined, and then specific subsets of the parameter space were manually selected for further investigation.

$$\hat{y} = \arg\max_{i} \sum_{j=1}^{m} w_j p_{ij} \tag{1}$$

An ensemble is created by selecting two or more of the same or different classifiers. We used soft-voting (Eqn. 1) [19] to fuse the results of each classifier. For ensemble selection, exhaustive search was used again, where all possible combinations of classifiers were combined. First we experimented with combinations of the above 8 classifiers. For the better performing classifiers, such as Linear SVM, we tried used multiple SVM classifiers. Lastly, we also tried varying the sub-set of features that were used, based on ensemble theory which says that it's better to combine weak classifiers that each perform well on different sub-sets of the data.

4.2 Transfer learning

Transfer learning is the learning of a new task using the knowledge that has been learned for a different task. Transfer Learning has been extensively used throughout Machine Learning literature in different forms. An exhaustive survey of different forms of Transfer Learning can be found in [20]. The use of transfer learning has been extensively researched in the field of Deep Learning [21] [22]. The authors in [21] talk about the two methods extensively used for transfer learning in Deep networks.

1. Feature-extraction from a pre-trained network - In this strategy, the the last two fully connected-layers of the pre-trained network are replaced with our own uninitialized layers. The weights of the pre-trained network are frozen and thus not affected by the back-propagating gradients. The network in this case can be then seen as a feature extractor.
2. Fine-tuning of a pre-trained network - In this strategy, we follow the same steps as above but instead of freezing the weights of the pre-trained network we use a small learning rate to "finetune" the weights for the task in hand. This is considered to be a stronger form of Transfer Learning.

4.2.1 Feature Extraction with AlexNet

Here we used the first strategy. We removed the last two FC layers of a pretrained AlexNet [23]. The activations of rest of the ConvNet was used as features to train another network which consisted of two fully connected layers with ReLU activation for the hidden layer and LogSoftMax activation for the output layer.

4.2.2 Fine-tuning of ResNet

ResNet (Residual Network) [24] won the ILVSRC competition (ImageNet Large Scale Visual Recognition Competition) in 2015. It was observed that with increasing depth, accuracy starts to saturate and then degrade. However, unlike what happens in over-fitting, with degradation, the train error increases as well. ResNet uses the innovative bottleneck architecture and instead of making multiple non-linear layers approximate the hypothesis function $H(x)$, they are made to approximate a residual function $F(x) = H(x) - x$. The idea is that the solver may drive the weights of the layers to zero in order to attain identity mappings. We use 18 layer deep ResNet, trained on the ImageNet database and fine-tune it for our dataset.

4.3 End to End Learning

End to end learning of neural networks here refers to the training of the neural networks from scratch (without pretrained networks) using backpropagation. In this method we tried multiple deep network architectures and trained each of them from scratch. The performance of each network was compared against a held out test dataset. Since the dataset was heavily skewed (10,000 images belong to one class) we have to undersample the largest class (no retinopathy) and oversample the other fewer classes. While oversampling we also performed heavy data augmentation so that the network is able to learn certain discriminative features. The data augmentation techniques tried included random cropping, jittering, rotation, flipping etc.

We started with an AlexNet based architecture [23] in which every convolution layer was followed by a non-linear activation (ReLU non-linearity) and a maxpooling. We had 5 such convolution layers followed by a Fully connected layer which was connected to five node output layer. We initially used the Categorical-Cross Entropy as our loss function to train the network. The results for this network can be seen in **??** (Model 1).

Figure 3: ResNet architecture

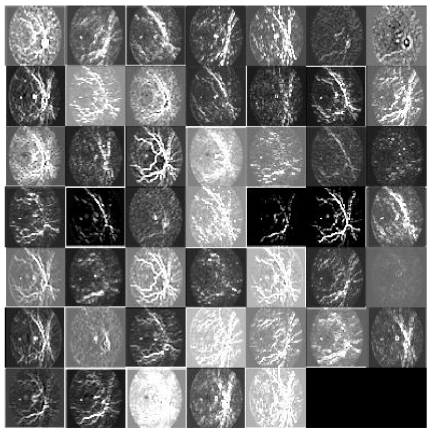

Figure 4: Features extracted from the fist convulational layer of ResNet

VGGNet (Oxford Visual Geometry Group Network) [25] was the network that achieved state-of-the-art performance on ILVSRC (ImageNet Large Scale Visual Recognition Competition) in 2014. The network introduced the concept of using multiple small sized convolutions one after the other instead of using one large convolution since this allows you to learn more fine-grained image features. Our next network was modeled on this idea. We used multiple convolutions followed by non-linear activations and max pooling. We empirically found out this to perform better than our previous architecture. Hence this was used as the standard architecture for all future experiments.

The standard loss function used in multi-class classification problems is Categorical Cross Entropy loss function. This loss function assigns a probability value to each output node based on it's activation value. However our classification problem is more close to an ordinal regression problem since it's much worse to misclassify Severe retinopathy as No retinopathy as compared to Moderate retinopathy. To formulate our problem as an ordinal regression problem we used the Mean Squared Error (MSE) as our loss function. We clip the loss function value between 0 and 4 since our class ranges between these values. Thus our final loss function is an MSE loss. We also experimented with combining the MSE loss with the softmax loss but did not make much progress in this direction given the lack of time.

5 Results

5.1 "Classic" Ensemble Learning

Using a single Linear SVM classifier with all 19 features from the Messidor dataset we achieved an accuracy score of 0.746, as shown in Table 2. We found that using the same classifier on specific subsets of only 2 or 5 features, we could achieve almost the same accuracy. This is discussed further in Section 6.

Unfortunately we could not find an ensemble classifier that produced a greater accuracy than just a single Linear SVM. Table 3 shows that an ensemble of all 6 basic classifier types resulted in an accuracy of 0.704, and an ensemble of two Linear SVMs on 3 features produced the highest accuracy of 0.740

From the feature analysis it was hard to visibly see any clear boundaries, so we are happy that our classifer achieved a reasonable accuracy. However, previous work reported accuracy scores of 0.94 [11] and 0.90 [18] so we failed to achieve the prior state-of-the-art.

Table 2: Single classifier results

CLASSIFIER	PARAMETERS	ACCURACY	KAPPA	F-SCORE
Multinomial Naive Bayes	smoothing = 0.1	0.576	0.145	0.608
Gaussian Naive Bayes	N/A	0.595	0.22	0.441
k-NN	No. neighbours = 16, Weighted by distance	0.675	0.354	0.670
AdaBoost with Tree Stumps	No. estimators = 100, Learning rate = 1.5	0.694	0.385	0.714
Decision Tree	Max features = 10, Max depth = 9	0.653	0.308	0.641
Random Forest of Trees	No. estimators = 120, Max features = 12, Max depth = none	0.693	0.387	0.691
SVM (Linear)	**Error penalty = 1**	**0.746**	**0.496**	**0.740**
SVM (Gaussian)	Error penalty = 8, Gamma = 0.01	0.658	0.311	0.691
Linear SVM on subset of features	Error penalty = 1, Features = (2,4)	0.731	0.462	0.729
Linear SVM on subset of features	Error penalty = 1, Features = (2,3,4,5,6)	0.738	0.481	0.726

Table 3: Ensemble classifier results

CLASSIFIER	FEATURES	ACCURACY	KAPPA	F-SCORE
GNB, kNN, AdaBoost, Dtree, Forest, LSVM	(1:19)	0.704	0.421	0.667
Forest, LSVM	(1:19)	0.699	0.411	0.653
Forest, LSVM, Dtree	(1:19)	0.704	0.419	0.660
2 x LSVM, C = 1.0, 0.1	**(2,3),(3,4)**	**0.740**	**0.482**	**0.736**

5.2 Transfer Learning

5.2.1 AlexNet

Figure 5: Features extracted from AlexNet

With the transfer learning approach we attained an accuracy of 25 %. We think this is because AlexNet was trained on the ImageNet database which were not medical images. Therefore the features learned by AlexNet were not useful for our dataset. Figure 5, shows the feature activations of AlexNet.

5.2.2 ResNet

Table 4 and Figure 6 show the results of Resnet fine-tuning.

Table 4: Accuracy, Kappa and F-score for ResNet

Accuracy	Kappa	F-Score
0.766	0.651	0.514

Figure 6: Confusion matrix for ResNet

5.3 End to End Learning

The results for training our own network are shown in Table 5 and Figures 7 and 8.

Table 5: Accuracy, Kappa and F-score for End to End Learning

Loss Function	Accuracy	Kappa	F-Score
#1 LogSoftMax	0.621	0.382	0.346
#2 LogSoftMax	0.754	0.423	0.350
#3 MSE	0.736	0.676	0.417

Figure 7: Confusion matrix for end to end learning

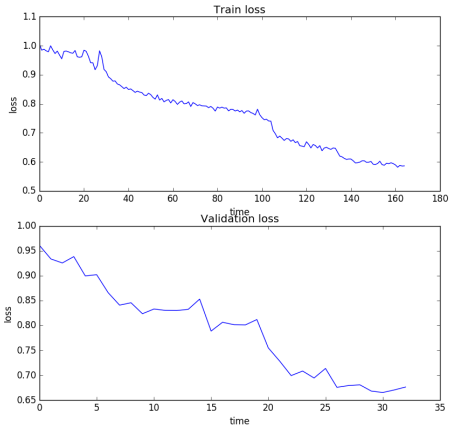

Figure 8: Loss function using MSE loss

6 Discussion

6.1 "Classic" Ensemble Learning

The classic approach to image classification using machine learning relies on having both high-quality image feature detectors and a classifier that can discriminate the classes using these features. Therefore the performance of our classifier can only be as good as the features we detect. Our Linear SVM with only 2-3 features produced an accuracy almost as high as using all 19 features. This suggests that many of the detected features offer very low discriminating power.

Our best ensemble classifier produced a slightly lower accuracy than the single Linear SVM. This is not as expected and suggests that the system is over-fitting. In some cases we noticed a sharp decrease

in accuracy which suggests there is too much noise in the feature data and subsequent classifiers are not learning anything new.

Unfortunately, to improve our method we would have to improve the image feature detectors we used, which is a separate research area.

6.2 Deep Learning

Using pretrained networks for feature extraction is a standard technique in Computer Vision. The low level image features e.g. Haar descriptors, blobs and interest point detectors are shared all across the Vision Community. It is also thought that deep neural networks do learn these low level representations and are able to finetune them according to the task in hand. For our initial method since we are using the pretrained AlexNet for feature extraction we believe that the network is not able to generalize to retinal images since the ImageNet training dataset it has been pretrained on doesn't have any medical images in it.

For our second method in Deep Learning we tried finetuning all the layers in pretrained ResNet [24] model. We chose ResNet-18 and ResNet-34 for our experiments since the larger ResNet models wouldn't fit in our GPU's. Finetuning all the layers is considered as a strong form of Transfer Learning and has shown to give excellent results [21] in the past. Our experiments show that indeed the image features being learnt by the Deep Network are general in nature and can be used in a very different classification task by finetuning the weights for the new dataset.

In our experiments with end to end learning of our own deep network we empirically found the use of VGG net type of architecture of multiple convolutions being bundled together to perform better. We also discovered the benefits of pre-processing images before feeding into the network. Also, since the dataset was highly skewed undersampling and oversampling of the data proved immensely useful. Our experiments proved that without proper data augmentation a high capacity network can easily overfit the training data and thus generalize worse on the test dataset. Another, interesting result we found was that the network can overfit on oversampled classes, even with data-augmentation and thus using regularization techniques such as Dropout, BatchNormalization, PReLU etc. assume large amount of significance. In our best trained model every couple of convolution layers were followed by batch normalization before feeding it into PReLU.

We trained most of our networks with the standard Categorical Cross Entropy loss function and an MSE loss function. Since the problem is similar to an ordinal regression problem our experiments showed that using an MSE loss function better models the problem. In Table 2 above we can see that the networks "#1" and "#2" have a high accuracy but a low Kappa score. These networks have been trained with a Cross Entropy loss function and we get a high accuracy since the model tries to classify most images into the more dominant category. However, in row 3 our "#3" model gets a high accuracy as well as a high Kappa score. This models uses an MSE loss function. We believe that using both the Cross Entropy loss and the MSE loss together will give the best results. We experimented with such a loss function but had to give up in the interest of time.

Most image classification tasks today use an Ensemble of ConvNets for their final classification to get state of the art results. We believe that since propping up the accuracy or the Kappa score is just a number game using a more deeper network and ensemble of such networks will give the best results. Another future research direction could be to use Discriminative Unsupervised Feature Learning to learn better feature representations which are discriminative enough for the classification task.

References

[1] M. Engelgau, L. Geiss, J. Saaddine, J. Boyle, S. Benjamin, E. Gregg, E. Tierney, N. Rios-Burrows, A. Mokdad, E. Ford, G. Imperatore, and K. Narayan, "The Evolving Diabetes Burden in the United States," *Annals of Internal Medicine*, vol. 140, no. 11, pp. 945–950, 2004.

[2] P. J. Kertes and T. M. Johnson, *Evidence-based Eye Care*. Lippincott Williams & Wilkins, 2007.

[3] H. Xu, T. Curtis, and A. Stitt, "Pathophysiology and Pathogenesis of Diabetic Retinopathy," *Diapedia*, 2013, [Accessed 22-October-2016]. [Online]. Available: http://dx.doi.org/10.14496/dia.7104343513.14

[4] Arleo Eye Associates, [Accessed 28-October-2016]. [Online]. Available: http://www.arleoeye.com

[5] H. Pratt, F. Coenen, D. M. Broadbent, S. P. Harding, and Y. Zheng, "Convolutional Neural Networks for Diabetic Retinopathy," *Procedia Computer Science*, vol. 90, pp. 200–205, 2016.

[6] E. Colas, A. Besse, A. Orgogozo, B. Schmauch, N. Meric, and E. Besse, "Deep Learning Approach for Diabetic Retinopathy Screening," *Acta Ophthalmologica*, vol. 94, no. S256, 2016.

[7] S. Roychowdhury, D. D. Koozekanani, and K. K. Parhi, "DREAM: Diabetic Retinopathy Analysis Using Machine Learning," *IEEE Journal of Biomedical and Health Informatics*, vol. 18, no. 5, pp. 1717–1728, 2014.

[8] U. R. Acharya, C. M. Lim, E. Y. K. Ng, C. Chee, and T. Tamura, "Computer-based Detection of Diabetes Retinopathy Stages using Digital Fundus Images," *Proceedings of the Institution of Mechanical Engineers, Part H: Journal of Engineering in Medicine*, vol. 223, no. 5, pp. 545–553, 2009.

[9] W. L. Yun, U. R. Acharya, Y. Venkatesh, C. Chee, L. C. Min, and E. Ng, "Identification of Different Stages of Diabetic Retinopathy using Retinal Optical Images," *Information Sciences*, vol. 178, no. 1, pp. 106 – 121, 2008.

[10] U. R. Acharya, C. K. Chua, E. Y. K. Ng, W. Yu, and C. Chee, "Application of Higher Order Spectra for the Identification of Diabetes Retinopathy Stages," *Journal of Medical Systems*, vol. 32, no. 6, pp. 481–488, 2008.

[11] J. Nayak, P. S. Bhat, R. Acharya U, C. M. Lim, and M. Kagathi, "Automated Identification of Diabetic Retinopathy Stages Using Digital Fundus Images," *Journal of Medical Systems*, vol. 32, no. 2, pp. 107–115, 2008.

[12] T. Kauppi, V. Kalesnykiene, J.-K. Kamarainen, L. Lensu, I. Sorri, H. Uusitalo, H. Kälviäinen, and J. Pietilä, "DIARETDB0: Evaluation Database and Methodology for Diabetic Retinopathy Algorithms," *Machine Vision and Pattern Recognition Research Group, Lappeenranta University of Technology, Finland*, 2006.

[13] T. Kauppi, V. Kalesnykiene, J.-K. Kamarainen, L. Lensu, I. Sorri, A. Raninen, R. Voutilainen, H. Uusitalo, H. Kalviainen, and J. Pietila, "The DIARETDB1 Diabetic Retinopathy Database and Evaluation Protocol," in *Proceedings of the British Machine Vision Conference*. BMVA Press, 2007, pp. 15.1–15.10.

[14] E. Decencière, X. Zhang, G. Cazuguel, B. Lay, B. Cochener, C. Trone, P. Gain, R. Ordonez, P. Massin, A. Erginay, B. Charton, and J.-C. Klein, "Feedback on a Publicly Distributed Database: The MESSIDOR Database," *Image Analysis & Stereology*, vol. 33, no. 3, pp. 231–234, Aug. 2014. [Online]. Available: http://www.ias-iss.org/ojs/IAS/article/view/1155

[15] "Kaggle datasets: Diabetic Retinopathy Detection," [Accessed 5-October-2016]. [Online]. Available: https://www.kaggle.com/c/diabetic-retinopathy-detection/kernels

[16] L. van der Maaten and G. E. Hinton, "Visualizing high-dimensional data using t-sne," *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008.

[17] L. I. Kuncheva, *Combining Pattern Classifiers: Methods and Algorithms*. Wiley-Interscience, 2004.

[18] B. Antal and A. Hajdu, "An ensemble-based system for automatic screening of diabetic retinopathy," *CoRR*, vol. abs/1410.8576, 2014.

[19] D. Ruta and B. Gabrys, "Classifier selection for majority voting." *Information Fusion*, vol. 6, no. 1, pp. 63–81, 2005.

[20] S. J. Pan and Q. Yang, "A survey on transfer learning," *IEEE Transactions on Knowledge and Data Engineering*, vol. 22, no. 10, pp. 1345–1359, Oct 2010.

[21] J. B. Y. L. H. Yosinski, Jason; Clune, "How transferable are features in deep neural networks?" in *Advances in Neural Information Processing Systems 27*, 2014, pp. 3320–3328.

[22] Y. Bengio, "Deep learning of representations for unsupervised and transfer learning," *JMLR: Workshop and Conference Proceedings 27:17*, vol. 27, no. 17, 2012.

[23] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[24] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *arXiv prepring arXiv:1506.01497*, 2015.

[25] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *CoRR*, vol. abs/1409.1556, 2014.