Drashtikon: Extraocular Disease Classification

Chelsea Sidrane (csidrane), Zachary Maurer (zmaurer), Shloka Desai (shloka)

INTRODUCTION

- In this project, we developed a classifier for extra-ocular conditions based on noisy, natural images of people's faces.
- Extra ocular conditions are disorders that affect the outside of the eye.
- A classifier that identifies eye disorders from images of faces would be useful for diagnosis in resource-limited settings, such as in areas that suffer from a lack of expert personnel and/or high end diagnosis equipment.

DATASET

- Data summary
  - A dataset of about 7244 labeled images of patients from *Drashti Netralaya Eye Hospital* in Gujarat, India was used.
  - Each image is labelled with one of five eye conditions.
  - After pre-processing, the dataset of consisted 5059 labelled images.
- The test-train data split used was 80-20

FEATURES

- We pre-processed the images as follows:
  - Normalize the image histogram in order to increase the contrast of the image using the Pillow image processing library.
  - Identify the faces in each image and crop them to contain only the upper portion of the face using OpenFace and Dlib.
  - Resize each image to 150 x 300.

Sample processing for an image:
Normalize -> Crop and resize -> HOG feature extraction

- **HOG features**: The first set of features tried was histogram of oriented gradients, or HOG features on the cropped images. HOG features were chosen because they have proven very successful for object recognition and are resistant to variance in exposure.
- **Cropped HOG**: HOG features from each eye (cropped to 96x192) were also computed and then concatenated together (to avoid image artifacts). This was chosen in attempt to use only relevant portions of the image in classification
- **Principle Component Analysis:** Used to reduce the dimensionality of the HOG and cropped HOG features from 45,000 to roughly 4000.
- This results in four feature sets:
  - HOG features, with PCA
  - HOG features, without PCA
  - Cropped HOG features, with PCA
  - Cropped HOG features, without PCA

MODELS

- Three different models were used, with cost functions as noted:
  - SVM with linear kernel. The cost function, loss and kernel for this are given as:

$$J_\lambda(\alpha) = \frac{1}{m}\sum_{i}^{m} L(K^{(i)T}\alpha, y^{(i)}) + \frac{\lambda}{2}\alpha^T K \alpha$$

$$L(z, y) = \max\{0, 1 - yz\}$$

$$K(x, z) = x^T z$$

  - SVM with Radial Basis Function kernel

$$K(x, z) = \exp\left(-\frac{1}{2\tau^2}||x - z||_2^2\right)$$

  - Logistic regression

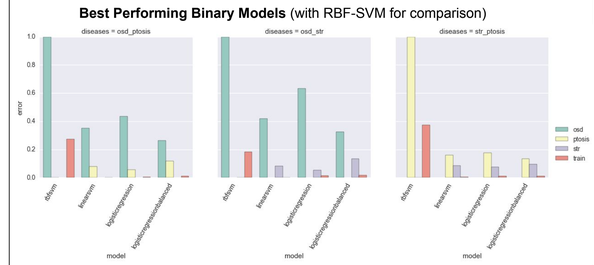
$$J(\theta) = \frac{1}{2}\sum_{i=1}^{m}\left(\frac{1}{1+e^{-\theta^T x}} - y^{(i)}\right)^2$$

- For each of these, a balanced version of the model was also used, which automatically adjust weights inversely proportional to class frequencies in the input data.
- Python's Scikit-learn library [1] was used to run the experiments.

- The first models we tested were binary classification on three 2-disease subsets (Strabismus vs Ptosis, Ptosis vs Ocular Surface Disease (OSD), and OSD vs Strabismus) for each of the feature sets and classification models described above.
- One-vs-all multiclass classification was then performed for five diseases: Strabismus, OSD, Ptosis, Goonderson Flap, and Dermoid Cyst.

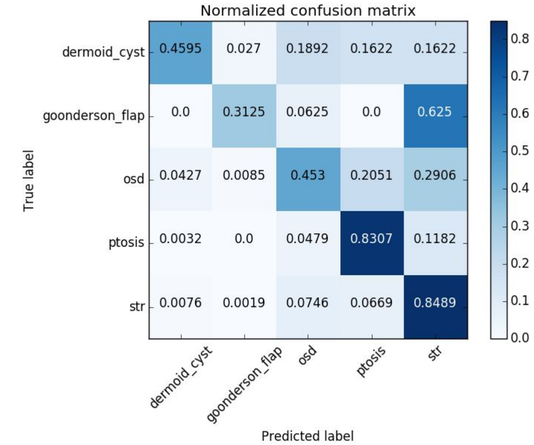
RESULTS

Best Performing Binary Models (with RBF-SVM for comparison)



Logistic Regression (Balanced) with HoG Features

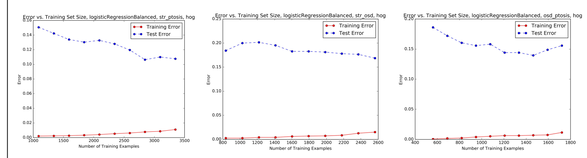
Normalized confusion matrix



Model	Feature	F1 Score (Global)
Logistic Regression (Balanced)	HoG	**0.774**
Logistic Regression (Balanced)	Cropped HoG	**0.775**
Logistic Regression (Balanced)	PCA Cropped HoG	0.744
Logistic Regression (Balanced)	PCA HoG	**0.775**
Logistic Regression	HoG	0.766
Linear SVM (Balanced)	HoG	0.768
Linear SVM	HoG	0.767
RBF SVM	HoG	0.52
RBF SVM (Balanced)	HoG	0.016

DISCUSSION

Error as a function of Training Set Size for the Best Model



Training curves showing the train and test error for binary classification pairs using HOG features

- Logistic regression performs the best, in general we see that linear classifiers (SVM with linear kernel and logistic regression) perform better than nonlinear classifiers (SVM with the RBF kernel and the convolutional neural network (see below)
- Fine-tuning a pretrained VGG16 [2] convolutional neural network was also tried but resulted in high training error (~73%) and test error (50%)
- The learning curve for training and test error vs training set size (see above) using HOG features indicated that our model had high variance. To fix this we reduced the number of HOG features using PCA (from 45,000 to 4000).

FUTURE WORK

- Obtain more data to classify whether a patient has an eye disease or is healthy (no disease), and classify some of the other more common eye diseases that don't necessarily need surgery (like conjunctivitis, stye etc.)
- Build our own neural network architecture, more specific to classifying eye disorders.
- Refine preprocessing, specifically the step where we identify the region corresponding to the eye, to utilize more of the original dataset.

REFERENCES

[1] Scikit-learn: Machine Learning in Python, Pedregosa et al., JMLR 12, pp. 2825-2830, 2011.
[2] Very Deep Convolutional Networks for Large-Scale Image Recognition. K. Simonyan, A. Zisserman. arXiv:1409.1556
[3] https://github.com/cmusatyalab/openface
[4] http://dlib.net/
[5] http://scikit-learn.org/