More than a mole? Digging into image classification with pyspark and distributed deployment of neural networks.

DS5110, Big Data Analytics - Final Project Report
Alexander Weech (aaw3ff), Jing Huang (jh4yd)
Note: ipynb list appended at the end of this document and provided separately

Abstract

Skin lesions are a common medical condition that can result in skin cancer. Initial screening of skin lesions is primarily visual and often requires an experienced dermatologist. Machine learning presents an opportunity to improve the visual classification of common skin lesions. Here we explore using spark to develop a pipeline for efficient training of neural networks and deployment of neural networks to featurize images for machine learning. The data used is a collection of 10015 color corrected images of pigmented skin lesions from the 2018 ISIC lesion classification challenge obtained through Kaggle. We find that we are able to achieve sensitivity and accuracy that is comparable with leading modern methods and trained dermatologists. Our goal was to integrate deep learning into our Spark system, and combine with Spark MLlib APIs to examine multi-class image classification on the Spark cluster. We find that logistic regression has superior overall accuracy at ~83%, but that a one vs rest paradigm with Gradient boosted trees provides superior recall for malignant lesions. Given the relatively low cost to a false positive, and the potentially lethal outcome of a false negative in this context, we find gradient boosted trees have more utility for this application,

Introduction:

Pigmented skin lesions are a family of common disorders with rising prevalence in the US, UK, and Australia (Shi, 2018). Most pigmented skin lesions are relatively benign with malignancy present in an estimated 1 in 33000 lesions being malignant (Walter, 2013). Melanoma is the most concerning of the pigmented skin lesions and has the potential to be fatal. However, melanoma has a high remission rate of over 99% when caught and excised early (Walter, 2013).

Accurate classification of pigmented skin lesions based on early screening data has the potential to save lives.

Initial screening of pigmented skin lesions is typically conducted by a dermatologist with a dermatoscope. However, dermatologist accuracy when classifying pigmented skin lesions is highly variable and inexperienced dermatologists with 1-4 years experience average ~60% accuracy in classifying the lesions (Prashad, 2020). Machine learning methods have the potential to aid in the pre screening process for pigmented skin lesions. Increased precision can save on needless medical procedures and increased recall has the potential to prevent serious illness. The goal of this project was to evaluate spark as a scalable platform for image featurization and machine learning in classifying pigmented lesions.

Data Description and EDA

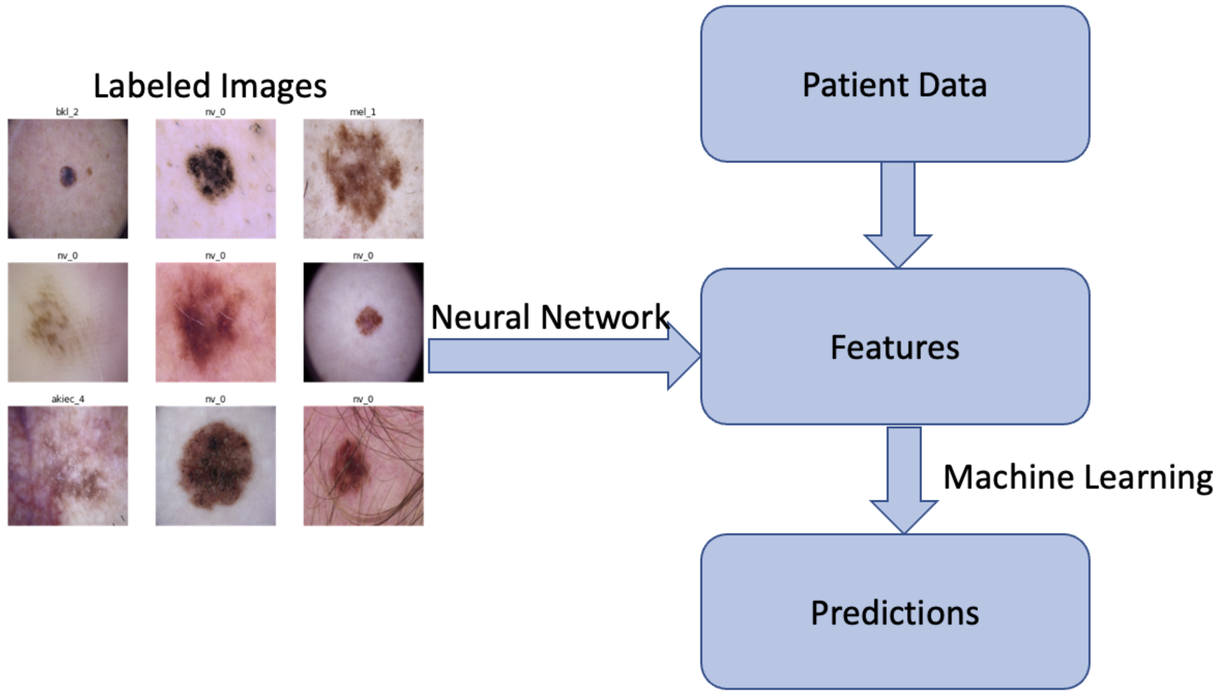
The dataset was obtained via hosting on Kaggle and derives from the Harvard 2018 ISIC challenge dataset. It consists of 10015 dermatoscope images of pigmented skin lesions collected from mixed populations. These lesions are curated to contain representative samples of common pigmented skin lesions and the proportions therein do not reflect population-level occurrence. The data is still highly imbalanced towards benign lesions.

The dataset is curated to contain representative images of what the Harvard dataverse team considers the seven important classes of pigmented skin lesions. Ground truth for the lesions was determined mostly by histopathology. This is currently the gold standard for diagnosis of dermal lesions. Other methods employed were follow-up, expert consensus among clinicians, and confocal microscopy. While the dataset is relatively small, the labels promise to be of high quality.

The dataset consists of 7 classes of skin lesions that will be abbreviated as: Melanoma (mel), Melanocytic nevi (nv), Basal cell carcinoma (bcc), Actinic keratoses and intraepithelial carcinoma / Bowen's disease (akiec), Benign keratosis-like lesions (solar lentigines / seborrheic keratoses and lichen-planus like keratoses, bkl), Dermatofibroma (df), Vascular lesions (angiomas, angiokeratomas, pyogenic granulomas and hemorrhage, vasc). 3 of them (akiec, bcc, mel) can be malignant, with melanoma standing out as the most dangerous lesion; other 4 types (nv, bkl, df, vasc) are benign.

The dermatoscope images have been color corrected and are generally centered around the lesion with a varying amount of surrounding skin on display. A sample is displayed below. It is worth noting that body hair has the potential to introduce noise at the image level. Consider the melanocytic nevi in the bottom right of the images sampled below. .

Workflow:



Data preprocessing

The images were loaded into a spark image data source from the directory. From there, the raw binaries of the images were read with imageIO and passed to pre-trained neural networks. The top layers of these networks were removed and the penultimate layers were converted to vectors of features for downstream machine learning tasks.

Dummy variable conversion

Sex and lesion location are the categorical predictors of interest from the patient metadata. These variables were indexed with the builtin spark string indexer and one-hot encoded for integration with the feature vector.

Data splitting/ sampling

Because our dataset contains 7 classes, and these classes were imbalanced, most of them were benign. To avoid under-representation of any single class in the test or training datasets, stratified sampling by diagnosis was used to split the data. This preserves the ratio of benign and malignant diagnoses across our training and test sets.

Neural networks for feature engineering were also trained on 80% of the dataset with 20% held out as a validation set. However, the training sample was augmented with random rotations and flips of the images in the 80% training set in order to reduce overfitting. Other transformations, such as random zooms and saturation were not performed. The images are color-corrected and largely centered on the lesion itself. We anticipate that saturation differences should represent true changes in the signal. Random zooms have the potential to inject data that excludes all or part of the lesion itself. Since we do not anticipate background skin to represent a meaningful feature, these are also omitted. Random rotations were considered from -.99*2pi to .99*2pi, effectively a full range of rotation. Orientation of the scope to the image is unknown, we do not expect orientation of the image to be a meaningful feature. Overall, these approaches should improve robustness with respect to noise from varying lesion orientation.

Methods

Dimension Reduction

PCA

The feature vectors generated by the neural networks are sparse and contain many highly correlated features. PCA was performed on normalized neural network features in order to explore the featurespace generated and evaluate whether dimension reduction resulted in appreciable loss of model accuracy. PCA performed particularly well on the Xception and Mobilenet models. The top 10 principal components capture over 90% of the variance in the data.

Feature reduction was also built into the transfer learning models themselves. Mobilenet, Xception, and ResNet50 are all models trained on the imagenet dataset which contains over 1000 classes. The dimensionality of the output space required for a 1000 class problem is substantially lower than the dimensionality of the output space required for our 7 class problem. The dense layers added on top of the pretrained models were reduced in size in order to shrink the dimensionality of the feature space explored downstream.

Similar Approaches were taken with Xception and mobileNet. Training with differential learning rates, fine tuning, and training with preloaded imagenet weights and a single learning rate did not significantly impact validation accuracy. It is possible that parameters for these models need to be adjusted.

Hyperparameter Tuning:

Hyperparameter tuning was run using 5 fold cross validation on the training dataset. Cross validated models were then evaluated on the 20% holdout dataset in order to evaluate the generalizability of the model.

For logistic regression, number of iterations and the regularization parameters were considered. After an initial coarse search, a more fine-tuned grid was tested around ridge regression with low elastic net parameters and a wider array of regularization parameters. For random forests, the number of trees and the number of variables subsampled were considered. For gradient boosted trees, we considered a one vs rest paradigm. This was implemented using pyspark ML's one vs rest classifier with the pyspark ML GBT package which only supports binary classification. The hyperparameters tuned were the maximum depth and maximum bins the trees could split variables into.

Model evaluation

The models were evaluated by using Spark's function of Multiclass Classification Evaluator. The major metric of the evaluation was weighted recall for overall multiclass classification on the test data. Optimal hyperparameters were determined with 5 fold cross validation and an identical seed in order to ensure that models are compared as fairly as possible. Train and test splits are also identical across models. Repeated cross validation was not feasible due to constraints on computing resources. We estimate that it would have taken at least 2800 core hours. Metrics are reported for the optimal cross-validated

Logistic Regression outperforms our other models in terms of weighted overall metrics. However, Gradient boosted trees has greater recall for melanoma. Gradient boosted trees actually outperforms logistic regression across the board for the detection of minority classes other than melanocytic nevi. Since melanocytic nevi are relatively benign, gradient boosted trees offer superior utility at the cost of overall accuracy. Since GBTs offer superior sensitivity and specificity for the most dangerous lesion on the list, we consider them the most useful model, if not the most accurate overall. This demonstrates the utility of a one vs rest approach can have in mitigating some of the effects of imbalanced data.

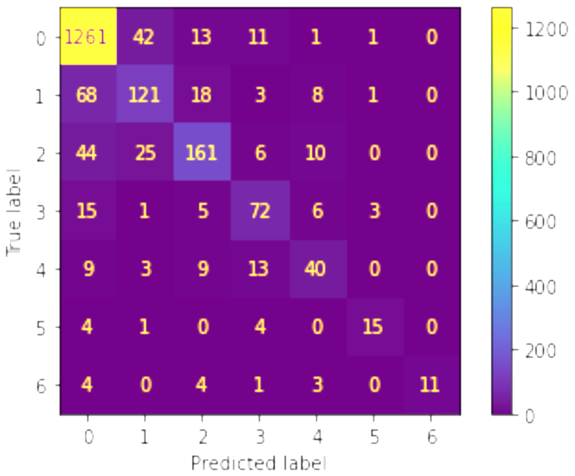
Results

	Keras ResNet50 --> PCA_top50			Ensemble 3 Transfer Learning Models		
Model Type	logistic regression	Random Forest	Gradient-Boosted Trees	logistic regression	Random Forest	Gradient-Boosted Trees
Hyper-parameters	regParam = 0.1, elasticNetParam = 0.0	numTree = 50, maxDepth =15	MaxIter=10, maxDepth=5, maxBin =16	regParam = 0.025, elasticNetParam = 0.0	numTree = 100, maxDepth =15	MaxIter=10, maxDepth=5, maxBin =32
feature size	53	53	53	2563	2563	2563
Test Accuracy	0.7947	0.7423	0.7363	0.8369	0.7803	0.7878
weighted-Precision	0.7776	0.7059	0.6933	0.8316	0.7515	0.7736
weighted-Recall	0.7947	0.7423	0.7363	0.8369	0.7803	0.7878

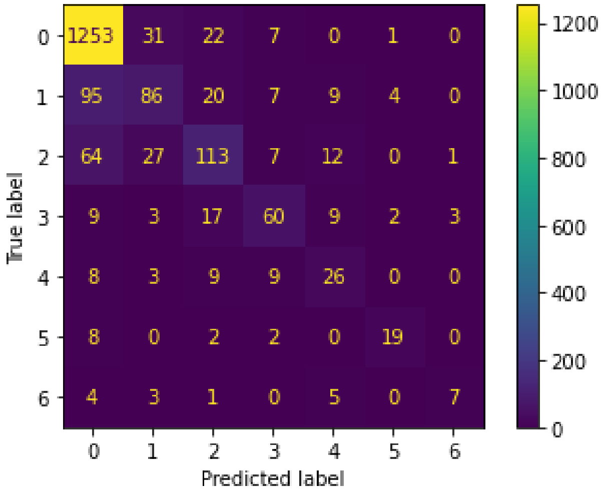
f1	0.7818	0.6939	0.7086	0.8337	0.7543	0.7760
AUROC (micro avg)	0.97	0.96	N/A	0.98	0.97	N/A

Confusion Matrices for Ensembles of 3 TL models:

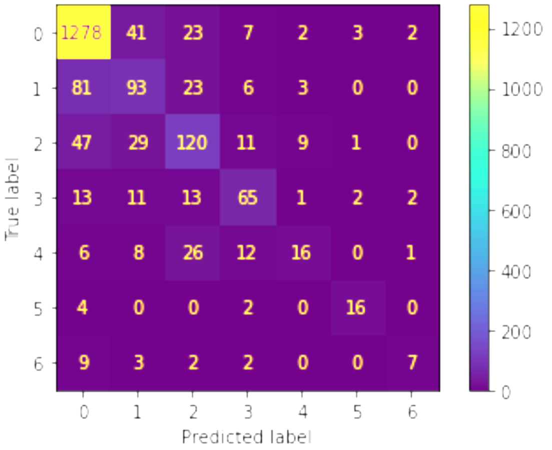
Gradient Boosted Trees



Logistic Regression (PCA)



Logistic Regression (All features)



Label	Dx	Label	Dx
0	**nv**	4	**akeic**
1	**mel**	5	**vasc**
2	**bkl**	6	**df**
3	**bcc**		

Models trained on a full set of 2563 features have substantially worse performance than models trained on features reduced by PCA. For example, a final pass for logistic regression over a narrow range of hyperparameters takes ~4 minutes to run on the reduced PCA features. Neural networks provide many correlated features and present a rich opportunity for feature reduction. This may increase scalability of pipelines involving neural networks for image featurization by improving the performance of downstream models and reducing the space requirements for storage of featurized data.

We demonstrate that spark and spark UDFs offer efficient pipelines for training of neural networks and the applications of complex models to new data. While spark deepImageFeaturizer has been deprecated, UDFs offer endless flexibility in model definition. Spark tensorflow distributor is used here to deploy neural networks to spark. We demonstrate that models are deployable in a scalable fashion. We also provide a cautionary tale on CPU training of neural networks. An ordinary consumer GPU running on a single worker outscales 8 workers using CPU on spark by an order of magnitude with a mean epoch time of ~30s for a resnet50 model vs 5 minutes for cpu training and 6 workers. Variability is noted in CPU training time on Rivanna with the instructional partition coming with an ~100% penalty.

Ensembles of neural networks have a storied history of working well for image classification. This particular set achieves 83% accuracy on holdout data for the HAM 10000 dataset. That would place it in fourth place among the 2018 ISIC competitors and fifth on papers with code's leaderboard. For reference, dermatologists with a decade or more of experience approach accuracy of 80% on this data while those with 3-4 years experience average 65%. These results demonstrate that our framework provides classification accuracy that is competitive with current modern methods. Tuning for both the neural nets and the follow up decision models are likely suboptimal given that this is a project focused on learning new platforms. However, we are able to muscle our way to competitive results using spark so to speak. We posit that utilizing spark to efficiently apply and train neural networks en masse to form ensembles may be an effective path forward to improving skin lesion classification. Our balanced accuracy on holdout data is midrange compared to other contestants in the ISIC 2018 challenge. All comparisons are approximate since submission of predictions on the true holdout data has ended and holdout ground truth has not been published. Since the project involves the application of novel concepts and relative unfamiliarity with ensembles of neural networks, better results could be achieved with more sophisticated neural network frameworks. Spark presents an opportunity to train neural networks in a distributed manner which is explored in a limited capacity here. Spark can also be used for extremely efficient application of trained networks to new data. This helps achieve reasonable results for skin lesion diagnosis and has broader applications to image classification workloads.

Follow-up:

A more rigorous approach to evaluating the benefits of PCA involving repeated cross validation to bootstrap confidence intervals for model metrics would be called for in order to evaluate suitability for large scale deployment. Unfortunately, this is also prohibitively expensive to compute for a class project. As written, these results postulate potential feasibility rather than outright demonstration of

For gradient boosted trees, a one-vs-rest classifier was considered with a single set of hyperparameters for all classifiers within the one vs rest framework at each model step. Tuning across this grid taxed the computational resources available to us. Published literature in image classification indicates that tuning the weights for each model in conjunction with tuning hyperparameters for each one vs rest case produces superior results. Unfortunately, without access to more cores to increase parallelism, this was not practical. It is possible GBTs in a one-vs-rest format would perform more optimally.

For training neural networks, a single GPU vastly outperforms the maximum resources we were able to requisition on the Rivanna cluster. An ensemble of a greater number of neural nets should be considered. Training on a local desktop with an RTX 3080 GPU resulted in epoch times of 20-40s, a roughly 10X speedup vs epochs of ~5 minutes accomplished with distributed CPU training. Additionally, custom-built neural networks may fit this data better. Initial forays were made in this regard, but our models all underperformed ResNet, Xception, and MobileNet. This is almost certainly due to relative inexperience in neural network architecture rather than unsuitability of this method.

Works Cited:

Chaturvedi, Saket S., et al. “Skin Lesion Analyser: An Efficient Seven-Way Multi-Class Skin Cancer Classification USING MOBILENET.” *Advances in Intelligent Systems and Computing*, 2020, pp. 165–176., doi:10.1007/978-981-15-3383-9_15.

“Deep-Learning-Transfer-Learning-Keras.” *Databricks*, docs.databricks.com/_static/notebooks/deep-learning/deep-learning-transfer-learning-keras.html.

“Receiver Operating Characteristic (Roc).” *Scikit*, SciKit Learn Developers, scikit-learn.org/stable/auto_examples/model_selection/plot_roc.html.

Shi, Katherine, et al. “A Retrospective Cohort Study of the Diagnostic Value of Different Subtypes of Atypical Pigment Network on Dermoscopy.” *Journal of the American Academy of Dermatology*, vol. 83, no. 4, 2020, pp. 1028–1034., doi:10.1016/j.jaad.2020.05.080.

“Transfer Learning and Fine-Tuning : Tensorflow Core.” *TensorFlow*, www.tensorflow.org/tutorials/images/transfer_learning.

Walter, Fiona M, et al. “Using the 7-Point Checklist as a Diagnostic Aid for Pigmented Skin Lesions in General Practice: A Diagnostic Validation Study.” *British Journal of General Practice*, vol. 63, no. 610, 2013, doi:10.3399/bjgp13x667213.

Doc List

FileName	Function Description
EDA.ipynb	load up meta file and images, combine them into one dataframe EDA, remove null and unknown, save the clean dataframe to a parquet file
PCA.ipynb	reduce the feature size from 100,352 to 50, save as a parquet file
three_model_pca.ipynb	PCA on the features generated by all three neural networks.
LogisticRegression_PCA.ipynb	read pre-saved quarquet file, split data into training/ validation set by a ratio 0.8:0.2 1. fit logisticRegress with 100,352 features, 5X CrossValidation 2. fit LR mode with PCA top50, 5X CrossValidation
ResNet50.ipynb	train a new transfer learning model Base model is ResNet50, train and validate the new model, the feature size of final model is 2048
LR_resnet_only.ipynb	split data into training/ validation set by a ratio 0.8:0.2 re-train LogisticRegression model with the new transfer learning model and 5X CrossValidation
RandomForest_PCA.ipynb	split data into training/ validation set by a ratio 0.8:0.2 1. fit RandomForest model with PCA top50, 5X CrossValidation 2. fit RF model with the new TL modeland 5X CrossValidation
GBT_PCA.ipynb	split data into training/ validation set by a ratio 0.8:0.2 1. fit Gradient Boosted Tress model with PCA top50, 5X CrossValidation 2. fit RF model with the new TL modeland 5X CrossValidation

Data_Organizer.ipynb	A small script for organizing files in the image directory into relevant subdirectories. Mostly a quality of life thing, open if you want to make sure we know how to use shutils.
Image_featurizer.ipynb	Loads images into a dataframe of spark image data sources, applies models to featurize image files, and combines the files with patient metadata for analysis.
LogisticRegression_all_features	Explore 5 fold CV logistic regression on the features yielded by all three transfer learning models. As well as undersampling.
RandomForest_all_features	Explore 5 fold CV Random Forest on the features yielded by all three transfer learning models. As well as undersampling.
GBT_all_features	Explore 5 fold CV one vs rest gradient boosted trees on the features yielded by all three transfer learning models. As well as undersampling.
Benchmark_notebook_local	Train a resnet 50 transfer learning model in local mode and time it (big freaking oof). Our Rivanna work is… suboptimal.