Classifying Bacterial Species using Computer Vision and Machine Learning

Venkatesh Vijaykumar
Independent Researcher
D1-402, Lenyadri CHS, Sector 19A, Nerul
Navi Mumbai- 400706, Maharashtra, India

ABSTRACT
The study embodied in this paper, aims at making use of machine learning and computer vision algorithms in order to reliably identify the species of bacteria, from their microscopic images. The study has taken into consideration three of the most commonly occurring species of bacteria that are clinically important. The work shown further in this study can be extended to a larger number of bacteria species. The study makes use of the Speeded Up Robust Features or SURF algorithm for detecting image keypoints. The artificial intelligence classifier makes use of these keypoint vectors as its input variable. It is noteworthy that the images used are taken at the gram staining stage of bacterial identification, in order to minimize any biases in the dataset owing to a variance in staining technique, although the feature detector invariably grayscales the image prior to keypoint computation. The paper follows the study from the conception of the idea, to the formulating of the algorithm, the training and testing of the same.

General Terms
Computer Vision, Machine Learning, Bioengineering

Keywords
Bacteria, Machine Learning, Classification, Computer Vision, Features, Neural Networks.

1. INTRODUCTION
Bacteria are single celled prokaryotic organisms. The term bacteria, is used for a very large sphere of organisms. Most bacteria are harmless, and are often beneficial to humans. Certain species of bacteria however, are capable of causing disease in humans and/or animals, thus becoming clinically important. These species may collectively be called pathogenic bacteria.

It is important to correctly identify bacterial species present in the bodily fluids or other bodily media in order to ensure swift and correct diagnosis and treatment. Typically, bacterial identification involves the preparation of slides with smears of the fluid or other media samples containing the bacteria, for microscopic study and also the preparation of bacterial cultures on specific growth materials (gels/agars). Both techniques involve certain biochemical tests performed in order to identify the bacterial species.

The most common biochemical test, and in most cases, the first step in identifying the bacterial species is the Gram staining procedure. The test, named after its inventor Hans Gram, classifies bacteria based on the presence of peptidoglycan in the bacterial cell wall. Those species with a presence of peptidoglycan layer in their cell wall retain the dye crystal violet. The remaining species get stained by a counterstain, usually safranin or fuchsine. Although this test is perhaps the first step taken in classifying bacteria, not all bacterial species are sensitive to this test, thereby creating a group of species, which are known as gram-indeterminate or gram-invariable bacteria.

To a great extent, bacteria are classified on the basis of their morphology [1] into sub groups such as, cocci, bacilli, vibrio et cetra. Microscopic study helps determine the morphological sub group of the bacteria under consideration. This further aids in classifying bacteria as per their species. Earlier attempts at classifying bacterial morphology have been made based on probabilistic classifiers [2]. However, attempts at these techniques in this study resulted in poor accuracy [3, 4].

The study makes use of computer vision and machine learning algorithms in order to attempt to classify bacterial species from their microscopic images. The procedure may briefly be outlined as such:

- Image Pre-Processing: Once the bacterial microscopic image is obtained, it must undergo some basic processing, in order that the algorithm runs efficiently. This involves cropping the image at a region of interest, into a 64 by 64 pixels image. This cropped image is used for all further purposes.
- Feature Extraction: The image is then subjected to a SURF feature detection algorithm, which detects the key points in the image, and generates a descriptor for each of these. These key points are used as the input vector for the machine learning classifier.
- Machine Learning Classifier and Prediction: The classifier is trained on a vast set of keypoint vectors. The newly generated key point vector is applied as an input to the network, and the class to which it belongs is predicted based on the characteristics of the vector.

2. METHODS
2.1 Cropping
The raw microscopic images of the bacteria were of a diverse range of resolutions and scales. While the scale of the image was not standardized, owing to the scale invariant nature of the feature detector, the resolution of the images had to be standardized, in order to obtain data points in an unbiased manner for all the classes of bacteria considered. In order to ensure this, it was decided that each image be broken up into parts of 64X64 pixels each, containing relevant visual information as regards the bacteria being observed. The cropping of images was also instrumental in avoiding redundant data points, or false features arising from structures other than bacterial cells or their colonies. The cropped images were then used to extract feature points, which will be used as the input vectors for the classifier network.

2.2 The Feature Extractors

The SIFT or Scale Invariant Feature Transform is an algorithm used to detect features as keypoints in a given image. These keypoints are scale as well as rotation invariant. These keypoints provide robust matches even upon viewpoint changes, or addition of noise [5]. The SIFT algorithm generates these features or key points using these principle steps:

- Scale-space Extrema Detection: This process makes use of the Difference of Gaussian (DoG) levels obtained by progressive Gaussian blurring at varying image sizes, in order to approximate a Laplacian of Gaussian to find edges and corners.
- Keypoint Localization: This process is performed in two stages: Locating the maxima or minima in the DoG images, and finding the sub-pixel maxima or minima. The sub-pixel values are generated by performing a Taylor expansion around the maxima or minima pixel location to locate sub-pixel keypoints.
- Orientation Assignment: Orientations are assigned to each of the located key points, based on the local image gradients. Future operations on the image data is on transformed values relative to scale, orientation and location for each keypoint, rendering it invariant to transformations of these
- Key point descriptor: The local image gradients are measured at the selected scale around each key point. These are transformed into a representation that allows for significant variance in illumination and local shape distortion.

The SURF or Speeded Up Robust Features is a feature detection algorithm, inspired by the SIFT algorithm. Unlike SIFT, the Laplacian of Gaussian is approximated with box filters. This helps convolve the box filter with integral images in parallel for different scales, speeding up the process. SURF also relies on the determinant of the Hessian matrix for both scale and location of the feature point [6].

For assigning orientations for the keypoints, SURF uses wavelet responses in horizontal and vertical direction for a neighborhood of size '6s' where 's' is the size. Adequate Gaussian weights may also be applied to this response. The dominant orientation is estimated by summing all responses within a 60 degree sliding orientation window.

For describing the features SURF again makes use of wavelet responses in the horizontal and vertical directions for a 20sX20s neighborhood around the key point. This region is further broken up into sub regions of 4sX4s. The horizontal and vertical wavelet response for each of these sub regions are used to form a vector:

$$v = (\sum d_x, \sum d_y, \sum |d_x|, \sum |d_y|)$$

This vector representation endows the SURF feature descriptor with 64 dimensions. To enhance feature distinction the SURF feature descriptor has a 128 dimension version, where the sums of $d_x \; and \; |d_x|$ are individually computed for $d_y < 0 \; and \; d_y > 0$. The the sums of $d_y \; and \; |d_y|$ are segregated by the sign of d_x, thereby returning twice the number of features. A comparative analysis reveals that though SIFT is most invariant to rotational transforms, it is slower to execute and that SURF performs faster and yields comparable results for the application [7].

The SURF algorithm was chosen as the final feature extractor, owing to its increased computational speed, and reduced complexity. The detector was implemented using the Python bindings of the OpenCV library [8]. The resultant vectors from the SURF feature extractor were stored in memory, to be used as the input vector for the machine learning classifier.

2.3 The Machine Learning Classifier

The present case is that of a multi-class classification problem. There are several ways to implement such a classifier, the most common ones being a Support Vector Machine or SVM, a Logistic Regression based classifier utilizing the one-versus-all principle for each class, clustering analysis classifiers, or neural network based classifiers. The approach undertaken for this study is a multilayer perceptron based classification algorithm.

The multilayer perceptron is a feedforward neural network; it has no cycling between two layers resulting in a unidirectional movement of information [9]. The perceptron is a learning algorithm for a binary classifier, which classifies an input based on the return value of the activation function, which is typically assigned as one for a class, and zero for other classes. A number of layers of such perceptrons, appropriately weighted, may be concurrently used, and are eventually simplified into two layer input-output systems by linear algebra simplification. A multilayer perceptron however, in addition to the linear activation function of a perceptron, also contains neurons with non-linear activation functions such as sigmoid function or hyperbolic tangent function neurons. A typical multilayer perceptron consists of three or more layers, indicating one or more hidden layers in addition to the input and output layers. Since a multilayer perceptron is a fully connected feedforward network, each connection is a weighted one. These weighted connections are adjusted through a backpropogation in order to perform supervised learning. Typically the change in the weights at each learning iteration is found using a gradient descent algorithm in order to minimize the error between the prediction and the target variable.

The multilayer perceptron was used as a classifier in this study, to differentiate between the bacterial species based on the feature vector input from the images. The feature vector extracted from the image was applied as the input vector for the multilayer perceptron. The activation function for the hidden layers was a rectified linear unit function, which acts as a ramp function, analogous to finding the maximum among the input of the neuron and zero, along with a unit addition to the neuron input. The 'Adam' algorithm was used for weight optimization and convergence. This algorithm is a first order gradient based optimization algorithm, which computes individual adaptive learning rates for various patterns [10]. A departure from 'Adam' would be the SEBOOST algorithm which boosts the standard SGD algorithm using memory of previous descent steps [11]. A maximum of 250 iterations was set. The algorithm was to be stopped if the iterations reached their maximum count or the convergence score did not improve by a tolerance of 0.0001 for two consecutive iterations. The dataset was split into a 70:30 training data to test data ratio for performing initial model validation. The model was further cross validated with a leave one out cross validation method.

3. RESULTS

The initial model validation was performed on a conventional 70:30 dataset split for training versus testing respectively. The result of the convergence loss for each iteration is represented graphically. The model is said to have converged when the

loss in two consecutive epochs do not improve by more than the set tolerance value, which in this case was 0.0001.

The convergence score for the training set was at a 100% fit, while that of the testing set was at 74.7826%.

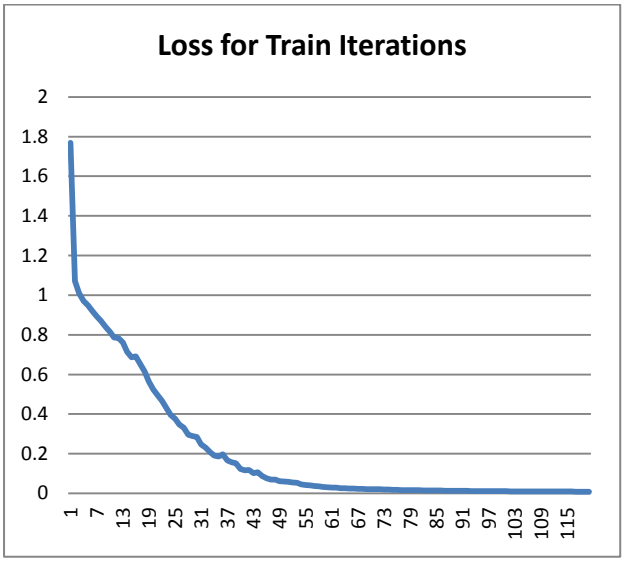

Figure 1: The iteration losses for the testing phase, indicated by the blue line

The leave one out cross validation [12] performed on the training set of 692 samples yielded a complete convergence on 472 samples. The results are represented in the pie chart as follows:

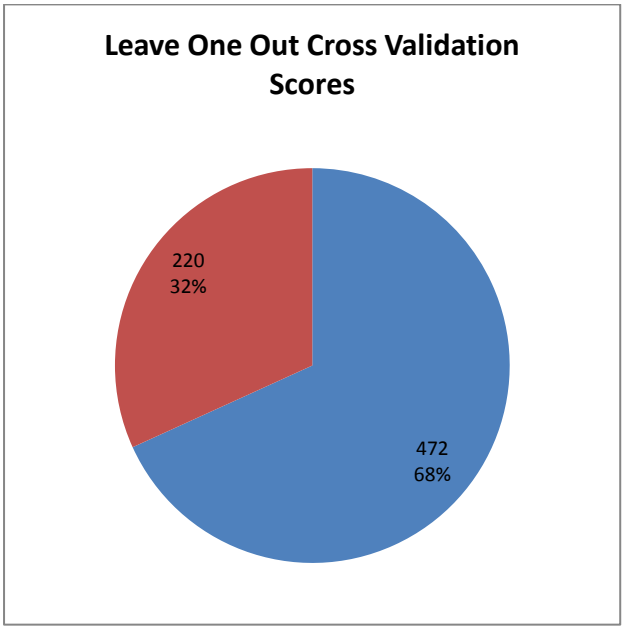

Figure 2: Leave one out cross validation results represented as a pie chart. Correctly classified data points are represented by the blue slice, incorrect ones by the red slice.

This shows a 68% convergence on the training set on the leave one out cross validation for the model under consideration. Although the leave one out cross validation was computationally expensive, owing to the n^2 iterations over an n length dataset, it was carried out to obtain a true picture of the model accuracy [13].

Aside from the conventional model validation, and the leave one out cross validation steps, a third method was also undertaken to test the model. This was a species wise validation of the model. The results of this validation step are as follows:

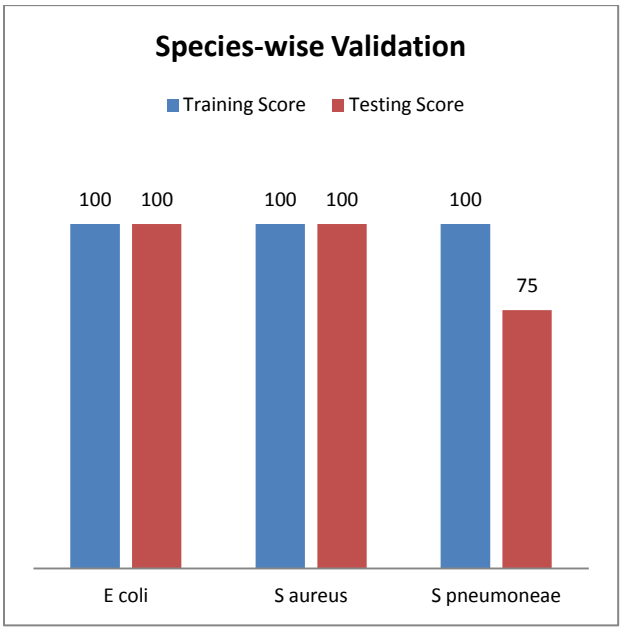

Figure 3: Species wise validation of the model.

The red bars represent the testing phase accuracy, while blue ones represent training phase accuracy. The species-wise validation model serves to show that the Streptococcus Pneumoneae species was identified to a lesser degree of accuracy than the remaining species. This phenomenon most probably arose due to the quality of the images used for this particular species, which were not as feature rich as the remainder, and therefore presented fewer distinguishable data points in order to facilitate the classification.

4. CONCLUSION

The algorithm conceptualized in this study, has to a satisfactory extent been able to classify bacterial species from their microscopic images. The accuracy of the model on the testing set, being close to 75% is not a cause for major concern, as the accuracy will improve with the addition of data points. Over the course of this study a support vector machine based approach was also tried, the accuracy for this model was 25% on the testing set, which was worse than taking a random guess at the species of a bacterium given three choices, in which case, one will have a 33% accuracy in predicting the correct species. Owing to this poor accuracy issue, the SVM based approach was abandoned. The current approach too, can be further refined with the omission of redundant features from the input vector, so as to optimize the execution of the model. Furthermore, the study was conducted only on Gram sensitive species of bacteria, a separate study for Gram invariant species using similar methods needs to be conducted in the future, in order to verify the accuracy of such a model with those species. The future scope of this study would be to create an extensive database for clinically important bacteria, based on which the model will undergo training, and ultimately be able to classify most of these bacteria to reasonable degrees of accuracy. A unified model for Gram sensitive and Gram invariant bacteria is an area for further research, should the model perform with equal accuracy with those species. The alternative to the unified

model would be a two layer classification model, which will use separate methods to identify species that are Gram sensitive and those that are not. The ultimate goal of the study is to craft a model capable of identifying most clinically important bacteria, in order to provide accurate and swift diagnosis.

5. ACKNOWLEDGEMENTS

The author expresses sincere thanks to all those who have helped this research see the light of day. Special thanks to Mr. Adarsh Ramanathan for his help with the model validation aspect of the paper and to Mr. Kushal Vyas for help with the feature selection aspects. Heartfelt thanks to all those who helped proof read this paper, and helped correct any errors.

6. REFERENCES

[1] Mohamad, N. A., Jusoh, N. A., Htike, Z. Z., and Win, S. L. 2014. Bacteria Identification From Microscopic Morphology: A Survey. International Journal on Soft Computing, Artificial Intelligence and Applications.

[2] Mohamad, N. A., Jusoh, N. A., Htike, Z. Z., and Win, S. L. 2014. Bacteria Identification from Microscopic Morphology Using Naive Bayes. International Journal of Computer Science, Engineering and Information Technology.

[3] Rennie, Jason D. M., Shih, Lawrence., Teevan , Jaime., Karger, David R. 2003. Tackling the Poor Assumptions of Naive Bayes Text Classifiers, International Conference on Machine Learning, Washington DC

[4] Berend, Daniel; Kontorovich, Aryeh. 2015. A Finite Sample Analysis of the Naive Bayes Classifier, Journal of Machine Learning Research

[5] Lowe, David G. 2004. Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision.

[6] Bay, Herbert., Tuytelaars, Tinne., and Van Gool, Luc. 2006. SURF: Speeded Up Robust Features. Graz, Austria : Springer-Verlag, European Conference on Computer Vision.

[7] Juan, Luo., Gwun, Oubong. 2009. A Comparison of SIFT, PCA-SIFT and SURF, International Journal of Image Processing.

[8] Bradski, G. 2000. The OpenCV Library, Dr. Dobb's Journal of Software Tools.

[9] Pal, Sankar K and Mitra, Sushmita. 1992. Multilayer Perceptron, Fuzzy Sets, and Classification. IEEE Transactions on Neural Networks.

[10] Kingma, D. P., and Ba, J. L. 2015. ADAM: A Method for Stochastic Optimization, International Conference on Learning Representations, San Diego.

[11] Richardson, Elad., Herskovitz , Rom., Ginsburg , Boris., Zibulevsky , Michael. 2016. SEBOOST – Boosting Stochastic Learning Using Subspace Optimization Techniques, eprint arXiv:1609.00629

[12] Kearns, Michael and Ron, Dana. 1999. Algorithmic Stability and Sanity-Check Bounds for Leave-One-Out Cross-Validation. Neural Computation. s.l.: Massachusetts Institute of Technology.

[13] Vehtari, Aki., Gelman, Andrew., Gabry, Jonah. 2016. Practical Bayesian model evaluation using leave-one-out cross-validation and WAIC, eprint arXiv:1507.04544v5.