

Classifying Electrical Impedance Tomography images by resistances based on contours

Sudhanva Narayana

Machine Learning Engineer,
Faststream Technologies,
Bengaluru, Karnataka, India
nsudhanva@gmail.com

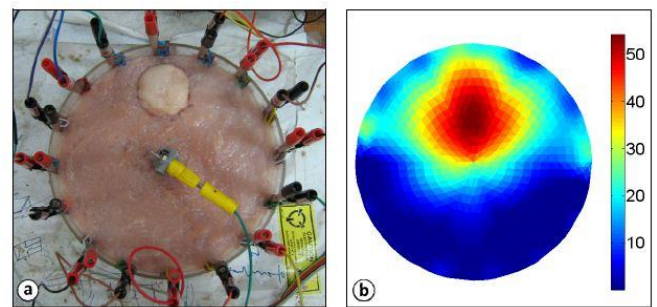
Vinay Bansal

Chief Executive Officer,
Faststream Technologies,
Bengaluru, Karnataka, India
vinay@faststreamtech.com

Vinod Agrawal

Chief Technology Officer,
Faststream Technologies,
Bengaluru, Karnataka, India
vinod.agrawal@faststreamtech.com

Abstract: This paper will be of interest for Health Care Data Analysts, Data Scientists, Doctors and Medical researchers. This paper provides an overview of current practice of Electrical Impedance Tomography (EIT), its imaging and use-cases. Electrical Impedance Tomography is a non-invasive type of medical imaging. These advances are improving our capacity to treat and even prevent cancers. The full implications of the subject remain to be explored. Examples of research techniques used in this project are detailed.



(Figure 1) - Sample Image of an EIT

The resistive properties of the images are assigned colour codes and are plotted.

1. Introduction

Human bodies have electrical properties, specifically the electric conductivity and permittivity. The electric conductivity is a measure of the ease with which a material conducts electricity; the electric permittivity is a measure of how readily the charges within a material separate under an imposed electric field [1]. Highly conductive materials allow both AC and DC currents to pass through them. Highly permissive materials allow only AC current to pass through them. Both of these properties can be used in medical applications as tumours, tissues and other irregularities in human body have different conductive and permissive properties. Other application of EIT include detection of blood clots, pulmonary emboli and gas in human body. Electrical Impedance Tomography – images are typically colour based representing the resistive properties on human bodies [2].

The following image (right) is an experimental EIT image generated using electrodes (Figure 1)

2. Machine Learning

Machine Learning is the science (and art) of programming computers so they can learn from data. For example, your spam filter is a Machine Learning program that can learn to flag spam given examples of spam emails (e.g., flagged by users) and examples of regular (non-spam) emails. The examples that the system uses to learn are called the training set. Each training example is called a training instance (or sample). In this case, the task T is to flag spam for new emails, the experience E is the training data, and the performance measure P needs to be defined; for example, you can use the ratio of correctly classified emails. This particular performance measure is called accuracy and it is often used in classification tasks [3].

Supervised Learning

Machine Learning systems can be classified according to the amount and type of supervision they get during

training. There are four major categories: supervised learning, unsupervised learning, semi-supervised learning, and Reinforcement Learning [4]. In supervised learning, the training data you feed to the algorithm includes the desired solutions, called labels. A typical supervised learning task is classification. The spam filter is a good example of this: it is trained with many example emails along with their class (spam or ham), and it must learn how to classify new emails [5].

3. Proposed Methodologies

The goal of the paper is to validate performance of Electrical Impedance Tomography's performance across various Machine Learning – Classification algorithms. Image is read into code in the form of a three-dimensional matrix where in each dimension represents intensities of the respective colour code. This three-dimensional matrix is then converted to two-dimensional matrix (representation of grayscale image) with intensities ranging from 0 to 1. Image is re generated to observe distribution using contour plots. Based on the data obtained and observation from the graphs, random multidimensional matrices are generated. Using radial basis function on these matrices, values ranging from 0 to 1 are created. 1000 random-related images are created based on the matrices and its values. The generated images are read back into code and are plotted to observe the distribution of intensities. Mean intensity ranges are calculated and are assigned labels (colours) correspondingly. The generated images are parsed and respective intensity ranges, its count of pixels and percentages are calculated. A dataset of 8 intensity ranges (columns) and 1000 values (rows) are created. Mean of pixel count of all ranges are taken in consideration and is used as a criterion for assigning targets. Binary targets are generated and are appended to the existing dataset as a target column [6].

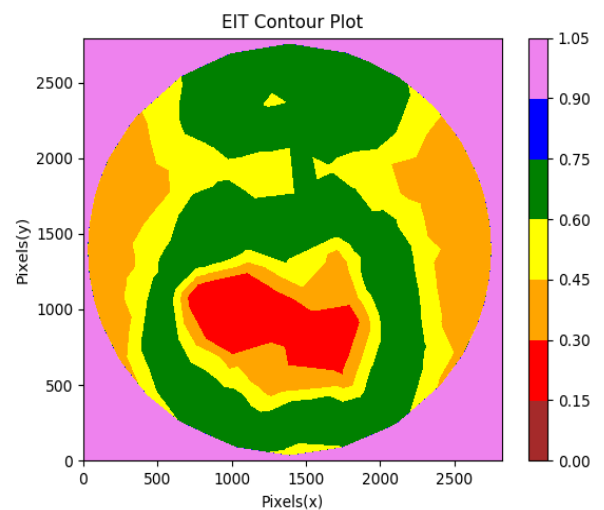
The paper works across any images in general but is concentrated on images generated by contours and sine, cosine functions. A sample EIT Image is read into the code in the form of a 2-dimensional matrix. This matrix represents intensities of various colour gamut. The project revolves around generating images and reading those images into matrices [7]. A dataset of count of pixels of various intensity ranges are created. A machine learning model is created out the dataset [8].

List of classifiers/algorithms used:

- K – Nearest Neighbours
- Decision Tree Classifier
- Kernel Support Vector Machines
- Logistic Regression Classifier
- Naïve Bayes Classifier
- Random Forest Classifier
- Support Vector Machines

4. Identifying pixels by contours

A typical contour image of the same EIT Image looks like this. The colours here are representative and are not related to any colours in resistive properties. The intensity range is from 0 to 1 where 0 being the lowest and 1 being the highest. The x and the y axis are count of pixels of the image. (Figure 2)



(Figure 2) - EIT Contour Plot of the sample image

5. Importing and analysing sample images

An image in processed in numpy (image processing library) is basically a three-dimensional matrix where each dimension represents the intensity of the respective colour value. A couple of few basic assumptions and paradigms are made when processing an image. The image is converted to grayscale. It can also, be a colour filter or a gradient but to simplify the process, it is converted to grayscale. A grayscale image is basically a two-dimensional matrix where each dimension represents the intensity of either black or white. 1000 sample images are generated using the following tools:

Numpy random (random number generator):

- Returns a sample (or samples) from the “standard normal” distribution.
- If positive, int_like or int-convertible arguments are provided, random generates an array of shape (d0, d1, ..., dn), filled with random floats sampled from a univariate “normal” (Gaussian) distribution of mean 0 and variance 1 (if any of the d_i are floats, they are first converted to integers by truncation). A single float randomly sampled from the distribution is returned if no argument is provided

The following code generates random numbers in the form of matrices using normal distribution:

```
# Generate data:
```

```
x, y, z = 10 * np.random.random((3, 50))
```

Numpy meshgrid (random matrices generator):

- Return coordinate matrices from coordinate vectors.
- Make N-D coordinate arrays for vectorized evaluations of N-D scalar/vector fields over N-D grids, given one-dimensional coordinate arrays x1, x2,..., xn.

The following code generates interpolation points using line space method:

```
# Set up a regular grid of interpolation points
```

```
xi, yi = np.linspace(x.min(), x.max(), 100),  
np.linspace(y.min(), y.max(), 100)
```

```
xi, yi = np.meshgrid(xi, yi)
```

Scipy interpolate:

- This sub-package contains spline functions and classes, one-dimensional and multi-dimensional (univariate and multivariate) interpolation classes

The following code interpolates using radial basis function of the ‘scipy’ module with linear parameter.

```
# Interpolate
```

```
rbf = scipy.interpolate.Rbf(x, y, z, function='linear')  
zi = rbf(xi, yi)
```

6. Preparing Dataset

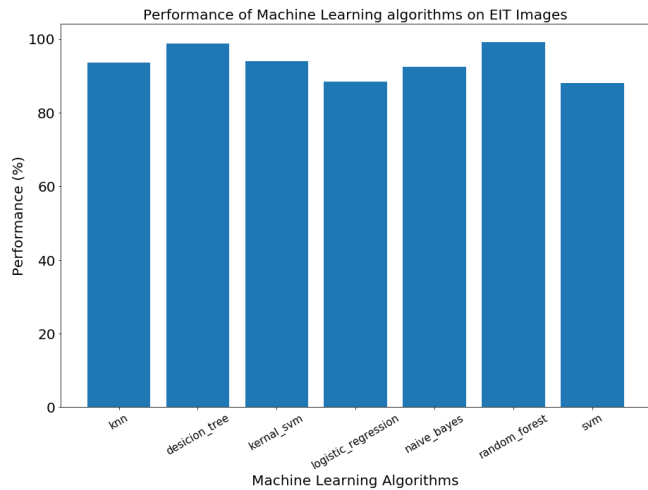
The generated images are parsed and are imported into code. These images are basically matrices as mentioned earlier. The pixel intensity is categorised based on the intensity range. Two datasets are generated. One contains the count of pixels categorised into respective intensity ranges and the other contains percentage of the pixels covered per image. The generated dataset is checked for errors and numerical anomalies. Once the data is pre-processed then mean of each column is calculated. These means are made the basic criteria for classification threshold. If the pixel count is more than the mean, the image is categorised as ‘1’. If the pixel count is less than the mean, the image is categorised as ‘0’. Once the classification is done, the ‘target’ column is appended and is attached to the existing dataset. Another dataset is generated and is considered as the final dataset for machine learning.

7. Cross validation

Cross-validation, sometimes called rotation estimation, or out-of-sample testing is any of various similar model validation techniques for assessing how the results of a statistical analysis will generalize to an independent data set. It is mainly used in the goal prediction, we estimate how accurately a predictive model will perform in practice. In a prediction problem, a model is usually given a dataset of known data on which training is run (training dataset), and a dataset of unknown data (or first seen data) against which the model is tested (called the validation dataset or testing set). The goal of cross-validation is to test the model’s ability to predict new data that were not used in estimating it, in order to flag problems like overfitting and to give an insight on how the model will generalize to an independent dataset (i.e., an unknown dataset, for instance from a real problem) [9].

8. Results

All algorithms are evaluated based on their performance. The results are interpreted and analysed. (Figure 3) (Figure 4)



(Figure 3) – Results

| No | Algorithms | Algorithms' accuracy |
|----|--------------------------------|----------------------|
| 1 | K Nearest Neighbours | 93.6% |
| 2 | Decision Tree Classification | 98.8% |
| 3 | Kernel Support Vector Machines | 94% |
| 4 | Logistic Regression | 88.4% |
| 5 | Naive Bayes | 92.4% |
| 6 | Random Forest Classification | 99.2% |
| 7 | Support Vector Machines | 88% |

(Figure 4) – Accuracy of Algorithms

9. Conclusion

The performance data is collected based on the confusion matrix produced by the algorithms. In the field of machine learning and specifically the problem of statistical classification, a confusion matrix, also known as an error matrix, is a specific table layout that allows visualization of the performance of an algorithm.

Logistic Regression and SVM showed similar results at 88%. Naïve Bayes, Kernel SVM and K Nearest Neighbours showcased a significant improvement resulting at around 93%. However, Decision Tree and Random Forest classifiers performed better than linear classifiers. Random Forest and Decision Tree both topped at 99% making them the best algorithm to use for this kind of dataset. We come to the conclusion that for a dataset of 1,000 rows, Random Forest and Decision Tree classifier are good algorithms and there

are major differences in accuracy on machine learning models and the model can be used on unseen data. Similar results can be expected on unseen data. However, performance on unseen data is yet to be measured.

10. References

- [1] Robert W. Stacey, Electrical Impedance Tomography, SGP-TR-182, Stanford University (2016)
- [2] Eoghan Dunne and Adam Santorelli, Supervised Learning Classifiers for Electrical Impedance-based BladderState Detection (2018)
- [3] Aurelien Geron (2017) Hands-On Machine Learning with Scikit-Learn and TensorFlow – 2nd ed.
- [4] S. B. Kotsiantis (2007), ‘Supervised Machine Learning: A Review of Classification Techniques’, Department of Computer Science and Technology, University of Peloponnese, Greece, End of Karaiskaki, 22100, Tripolis GR.
- [5] Grzegorz Klosowski, Tomasz Rymarczyk, Using neural networks and deep learning algorithms in electrical impedance tomography, Lublin University of Technology, Lublin (2017)
- [6] Amma, C. Krings, T., Böer, J. and Schultz, T. Advancing Muscle-Computer Interfaces with High-Density. Electromyography. In Proc. CHI '15, 929-938.
- [7] Amento, B., Hill, W., and Terveen, L. The sound of one hand: a wrist-mounted bio-acoustic fingertip gesture interface. In Proc. CHI '02, 724-725.
- [8] Beck, M. S., Dyakowski, T and William, R. A. Process, Tomography–The State of Art. Transactions of the Institute of Measurement and Control. 20(4), 163-177. Oct. 1998
- [9] Guoqiang Zhang and B. Eddy Patuwo, Michael Y. Hu (1997), ‘Forecasting with artificial neural networks: The state of the art’, Graduate School of Management, Kent State University, Kent, Ohio 44242-0001, USA