Machine Learning Project 4

By

Gautham Mudunuru

Paul Carrillo

John Agan

March 2021

**1. Introduction**

This report describes the design and implementation of a machine learning program which uses logistic regression to classify images potentially containing C. elegans worms into one of two classes: those with and those without the worm present.

**2. Design**

**2.1 Input Data**

The input data for this project was supplied by the previous work of another team and consisted of 5500 images labeled as containing an individual of C. elegans and 5500 labeled as not containing any individual of the species. The initial dimensions of the given grayscale images were 101 x 101 pixels.

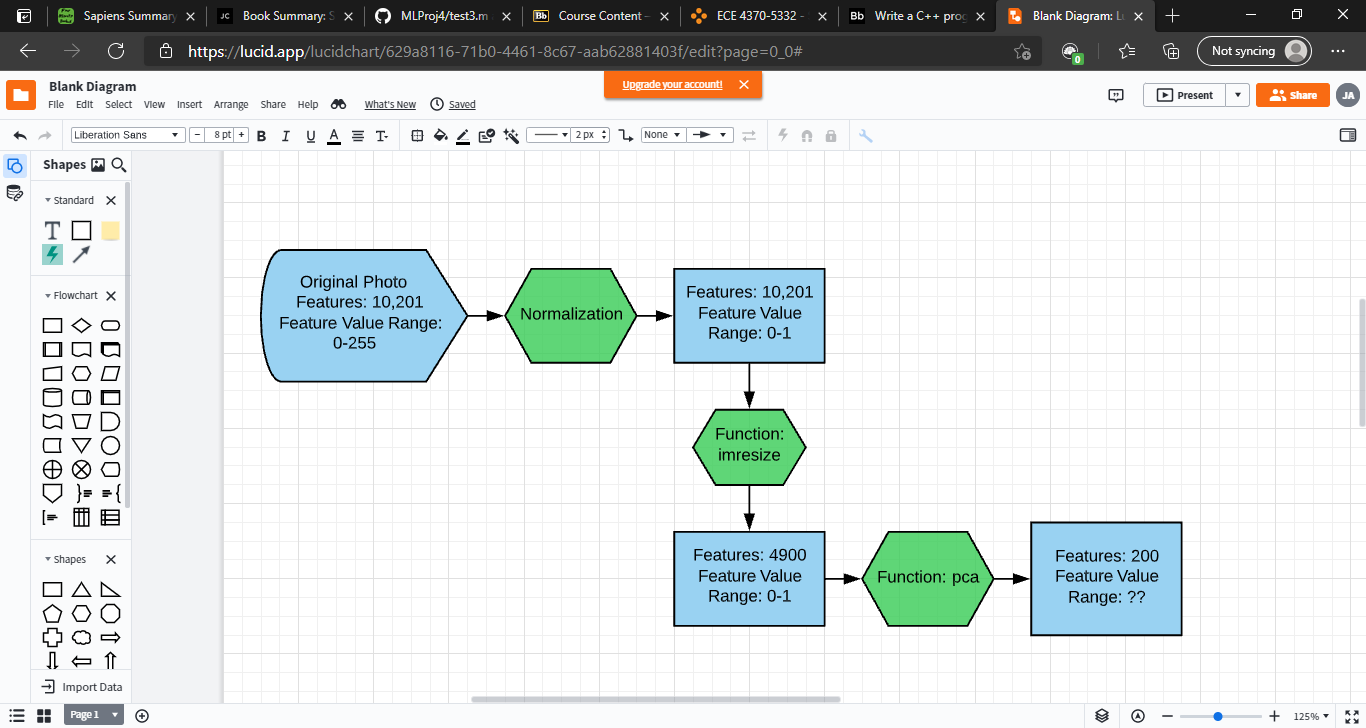
A visual inspection of the data revealed that while the images were generally well labeled, the set may contain errors. The proportion of these possible errors is apparently much greater in the no-worm subset. In the worm-containing subset, only two images (5282 and 5284) were considered highly questionable. This represents less than 0.04% of the subset. The no-worm subset was less well-curated. Clearly, even for a human (who isn’t a trained biologist), there are images in which it is very difficult to identify whether or not a given object is a worm. However, 65 images (1.18%) were identified in the no-worm subset, in which the presence of an individual seems very likely. Some cases are so strong as to be unquestionable (e.g., 2033 and 2799). The algorithm which originally classified these images appears to have had greater difficulty classifying images in which the worm appeared only partially and near the edge.

**2.2 Data Split**

Both subsets of the data were divided using holdout. Holdout was judged to be a sufficiently rigorous method of sampling since the data was observed to be relatively homogenous. Data was divided as follows:

* 60% training set
* 30% validation set
* 10% test set

**2.3 Input Image Processing**



The imported greyscale images are first normalized, dividing their pixel values by 255. They are subsequently rescaled to 0.7 times their initial size using bicubic interpolation, realized through Matlab’s built-in *imresize* function. Finally, principal component analysis using eigenvalue decomposition is used to reduce the dimensionality to just 200 features (from the original 10,201), which are organized into the feature matrix and sent as inputs into the logistic regression function.

**2.4 Logistic Regression Parameters**

Learning Rate:

This learning rate of the logistic regression algorithm was chosen after holdout validation of a logarithmically spaced set of (insert number here) possible parameter values.

**2.5 Optimizer**

This logistic regression algorithm employed batched gradient descent to determine the optimal weight values. As stated, the rho, or learning rate was (insert number here).

**2.6 Termination Criteria**

Number of iterations before termination: 200

**3. Results**

**3.1 Confusion Matrix**

**3.2 Execution Times**