**CS464: Introduction to Machine Learning**

**Project Progress Report**

Bora Bardük | Muhammed Çavuşoğlu | Duygu Durmuş | Mert İnan | Ekin Uyumaz

{bora.barduk, m.cavusoglu, duygu.durmus, mert.inan, ekin.uyumaz}@ug.bilkent.edu.tr

**Detailed and Concrete Description of Datasets**

The aim of the project is to match the artworks with the movements they belong to. WikiArt data from Kaggle is used in the project. However, since WikiArt data does not have same number of images for each art movement, we selected 15 art movements (Impressionism, Realism, Romanticism, Post-Impressionism, Surrealism, Art Nouveau, Baroque, Symbolism, Abstract Expressionism, Naive Art (Primitivism), Neoclassicism, Cubism, Northern Renaissance, Rococo) with high number of artworks and we are going to evaluate 2800 images for each movement during the project. For preliminary results, unbalanced 5000 images for chosen art movements were evaluated. All images were resized to reduce the number of HOG features since kNN was affected by the curse of dimensionality.

**Concrete Description of Approach**

The approach in our project is the comparison of three different machine learning algorithms on the same classification problem. The methods which will be used are: k-Nearest Neighbour (kNN), Convolutional Neural Network (CNN), Multi-Dimensional Long Short-Term Memory (LSTM).

* The kNN procedure consists of model building for the optimal k and using k-Means for cluster building.
* CNN procedure consists only of model-building by setting up the number and types of neural layers, proceeded by further customizing the filters associated with each layer. Feature selection is done automatically during CNN procedure.
* During the LSTM procedure, we will only specify the number of recurrence layers to be used during model building. Our model will then analyze the long-term dependencies between information.

The comparisons are going to be done based on loss functions, and accuracy. After these values are calculated for each procedure, we are going to make conclusions to assess and compare each algorithm.

**Work Done**

We have downloaded our WikiArt dataset. We have selected 15 art movements and eliminated images from unused art movements. We have resized every image in this dataset to 100x100 pixels using Python Imaging Library (PIL) hence reducing dataset size to 40 MB from 5 GB.

We had to match chosen 15 art movements with the images. The information about images was in another file. Since the image information file (filename, artist, title, style, genre, date) was not sorted, first we sorted the csv file based on filename. Then, by comparing the filenames of images in directory with the filenames in information file, we matched each image to its art movement.

**Preliminary Results**

In order to use categorization algorithms, paintings were fed into a feature extraction routine. HOG (Histogram of Oriented Gradients) features were extracted from each painting. These features represent the color histograms and the edge distribution and descriptions of pictures. MATLAB’s extractHOGFeatures function was used to get 4356 features.

For the preliminary tests, 5000 training images were used, which were then downsized to 3492 because of elimination of unused art movements. This training set was unbalanced in terms of its classes, yet, for preliminary results this caveat was presumed to not affect the results heavily.

For the first preliminary work, the training set was run through a k-Means clustering algorithm. Main aim of this step was to question whether the features would congregate by themselves into separate clusters. Yet, multiple different art movements got clustered into the same cluster by the kmeans function of MATLAB. This showed that the features may not contain sufficient information intrinsically, hence a feature selection procedure should be applied.

In the second part of the preliminary work, k-NN classification algorithm was used to classify the sample training data. fitcknn function of MATLAB is used. In this stage, multiple different k values have been tested, and for the training data that we used, accuracy was 35 %.

**Discussion of Challenges and Solutions**

* Unaccustomed Technologies: Throughout our project, we were subjected to use some technologies we had no prior knowledge upon. These technologies include using Python image library, and implementing CNN networks. We had to make some comprehensive research to continue with our project.
* Obtaining Preprocessed Data: Since data is essential for any machine learning project, we were nitpicky with the data to use with our models' training. The WikiArt data we obtained was highly unprocessed, that is it was not balanced (not every class is represented equally), and the sizes of images were not homogenous. As a solution, we produced MATLAB and Python software to automatically convert the data to a suitable form.
* CNN Data Compatibility: The data we had to train the models with were not in a suitable format to use with our selected CNN model building tool Tensor Flow. As a solution, we carefully studied the example data of popular Tensor Flow projects and understood in what format they input the data.
* Cross-Validation Long Runtime: The software we produce to train kNN model took a long time to do cross-validation, which imposed an issue if we want to run it many times during our testing. As a solution, we thought of decreasing the dataset size or rewrite the algorithm in a more efficient way.