**CS464: Introduction to Machine Learning**

**ArtMove - Final 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

**1. Overview**

ArtMove is about comparing different machine learning methods' performances at determining associated art movements of given artwork images. The feature extraction methods which were used are Histogram of Oriented Gradients (HOG) and activations of a Convolutional Neural Network (CNN). The classification models which were used are decision trees, linear & quadratic discriminants, multiclass Support Vector Machines (SVM), different types of k-Nearest Neighbour (kNN), and some ensemble models. The comparison of the results were evaluated by their accuracies, since there is a large amount of labels.

**1.1 Summary**

Before attempting our experiments, we were aware that CNNs require a high amount of data to be trained correctly. Because of the lack of data, the expectation was that implementing a CNN feature extraction would give the worst accuracy measures compared to more conventional image processing methods. After the experiment, we saw that features extracted from CNN performed the best compared to HOG while training same models using extracted features.

**1.2 Dataset Description**

Our images belonged to the WikiArt database and we obtained them from Kaggle website [1]. Images were separated according to their art movements, providing us the opportunity to train our data according to their labels.

Each art movement had various amount of images assigned to them in the downloaded data. Therefore, we balanced the data by choosing from only the 15 most used available art movements: Impressionism, Realism, Romanticism, Expressionism,Surrealism,Post-Impressionism, Art Nouveau, Baroque, Symbolism, Abstract Expressionism, Primitivism, Neoclassicism, Cubism, Rococo and Northern Renaissance.

In order to normalize our data, we had to resize them. For kNN and SVM, all training data were resized to 100x100 px. For CNN, they were resized to 227x227 as AlexNet would only accept images with these dimensions.

**2. Changes**

There have been three major changes after our progress report. The first change was that we decided not to implement LSTM as one of the algorithms we were considering. The reason for this was that LSTM and all other RNNs rely on the contextual/sequential properties of the input data. This is irrelevant to our cause, when identifying art movements in a random order, so we decided to focus more on other models we were working on instead.

The second change was the decision to use a pre-trained CNN solely for feature extraction. Because we have a limited amount of training data, we decided to use a pre-trained network. We used AlexNet to extract features, and run various kinds of models on the extracted data.

The final change was the introduction of new models to be applied on extracted features. These models include decision trees, linear & quadratic discriminants, multiclass SVMs, different types of kNN, and some ensemble models.

**3. Work Done**

**3.1 Separating Images**

We seperated our train and test dataset to 15 art styles. The train dataset contained 15000 images and test dataset contained 6000 images. However, we could only use a portion of data for AlexNet, as some images were not compatible with it [2]. As these images were plenty, we could only use select compatible images a few at a time manually.

**3.2 HOG Feature Extraction**

Histogram of Oriented Gradients (HOG) features were extracted from each 15000 training and 6000 testing images to be used as features in the machine learning models.

**3.3 AlexNet Feature Extraction**

We have used a pre-trained CNN AlexNet for feature extraction. Prior to running AlexNet, we have categorized our image set into subfolders of different art movements. We have stored our image set as an ImageDatastore object, since it automatically labels the images based on folder names. We have used 20 layers, layers from ‘data’ layer to fully connected ‘fc7’ layer. We have resized our image set to 227x227, since Image Input layer accepts 227x227x3 images as input. After activations of AlexNet, we have extracted 4096 features. These features were used to train various models and compare the results.

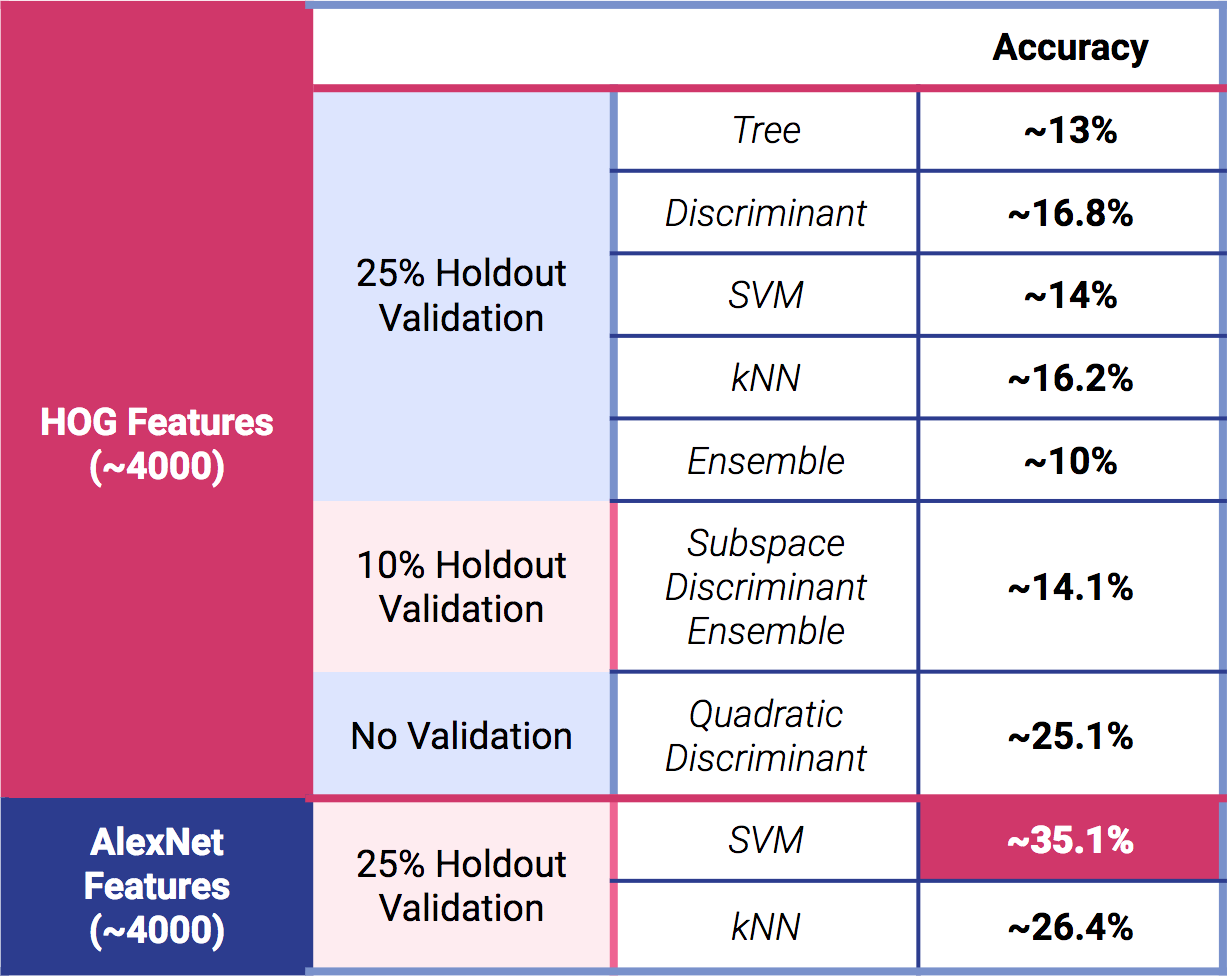
**3.4 Feature Selection**

In order to shorten training time, avoid curse of dimensionality and provide better accuracy, based on suggestions on previous proposal, we benefited from feature selection. Since it would take a lot of time to apply forward selection to all features, 5 best features were selected by using forward selection. After selecting best features, Classification Learner were used to apply several models (KNN, SVM, Ensemble etc.). The best accuracy obtained was 12.8% with Quadratic Discriminant-PCA(95%) model.

**3.5 Using Classification Learner App**

MATLAB provides an app for the purpose of using multiple different classification algorithms. By using the Classification Learner App in MATLAB, a diverse set of models were tested and different validation accuracies were acquired. This app also allows holding out 10-25% of the data for model validation. We have used this to choose the best model. We have used no validation to get the training accuracy. Furthermore, with this app, features were selected based on principal component analysis (PCA).

**4.Results**

****

*Table 1:* This table shows the validation and training accuracies for multiple models with both AlexNet features and HOG Features extracted from the image dataset.

**4.1 HOG Feature Extraction Results**

For the initial stage of the project, we have used the extracted HOG features to train multiple machine learning algorithms. Several validation holdout percentages were also used. Highest accuracy is found with quadratic discriminant model with no validation. Models which gave the highest accuracies are described in the following sections.

**4.1.1 Cosine KNN**

Cosine KNN is a nearest-neighbor classifier that uses the cosine distance metric. It was used with all features. The ROC curve for each feature, confusion matrix for positive predictive values and false discovery rates were observed. The accuracy was 16.2%

**4.1.2 Complex Tree**

Complex Tree is a decision tree with many leaves that makes many fine distinctions between classes (number of splits = 100). It was used with all features. Similarly, the ROC curve for each feature, confusion matrix for positive predictive values and false discovery rates were observed. The accuracy was 16.8%

**4.1.3 Quadratic Discriminant**

Quadratic Discriminant is a fast, easy to interpret classifier that creates elliptical, parabolic or hyperbolic boundaries between classes. It was used with 690 features(PCA). The accuracy was 25.1%. This model also detects Impressionism best so a pure set of 1000 Impressionism images were tested and the accuracy was 64.9%

**4.2 AlexNet Feature Extraction Results**

AlexNet features were extracted from a small amount of images and only a single validation strategy is employed, which is detailed in section 4.2.1.

**4.2.1 Fine kNN (25% Holdout Validation)**

Fine kNN is a knn with finely detailed distinctions between classes. The number of neighbors is set to 1. The accuracy obtained from this model was about 26.4%.

**4.2.2 Cubic SVM (25% Holdout Validation)**

Cubic SVM is classification using a third degree functions to separate hyperplanes and make classification. It was used on 4096 features extracted from AlexNet. This model performed the among other models, with accuracy about 35.1%. Considering there are 15 labels and we did not specifically train a neural network with this purpose, the results are significant.

**5. Challenges & Solutions**

When we obtained the images from Kaggle, we noticed that some images were over 200 MB. Python could not analyze images with such large sizes, therefore we had to manually remove these images from the datasets.

Many images were incompatible with AlexNet’s processing. Since removing these images was not possible due to their large amount, we found the solution in using only a small portion of the dataset and removing the incompatible images in that set.

Our train dataset for AlexNet did not contain many training samples. Therefore the accuracy that we obtained from CNN was not very high.

**References**

[1] P*ainter by Numbers | Kaggle.* [Online]. Available: https://www.kaggle.com/c/painter-by-numbers/data. [Accessed: 25-Dec-2017].

[2] “AlexNet,” *Pretrained AlexNet convolutional neural network.* [Online]. Available: https://ch.mathworks.com/help/nnet/ref/alexnet.html. [Accessed: 25-Dec-2017].