# Big Data and Machine Learning Coursework
# Thomas Tasioulis – 998273

## 1. Introduction

The aim of this paper is image classification to the dataset CIFAR-10 [1]. The dataset is split to train and test sample which it has 10,000 and 1,000 images, respectively, followed by their respective labels. The dataset is consisted of 10 type of images airplane, automobile, bird, cat, deer, dog, frog, horse, ship and truck. In order to classify properly the images, in this paper was used supervised learning as labels were given in order to classify optimally each image. Different models were used such as logistic regression, support vector machine (SVM) and neural network (NN). Furthermore, it was used a function to deduct features out of each image due to the original set had 3072 features per image. Moreover, after the deduction of the features, dimensionality reduction techniques were used in order to compress even more the features and reduce the complexity significantly. Overall, we were able to achieve an accuracy of almost 60% using SVM on the test set while other models approached almost to that accuracy as well.

## 2. Method

### *2.1 Features extraction*

The dataset is consisted from 10,000 images which are in stored in an 4D array of size 32x32x3x10000 for train test and 1,000 images for test set. Due to the curse of dimensionality, we used scikit-image package [2] in order to extract features out of each image which ended to a single array consisted of 324 rows for each image.

### *2.2 Unsupervised and Supervised Learning*

After the extraction of the features we ended up with a 2D array 10,000 x 324 for the train set and 1,000 x 324 for the test set. Then, we applied unsupervised learning techniques to see if it is possible for some algorithms to explain the data. We applied k-means and Gaussian Mixture Model (GMM) to the dataset by setting 10 classes and random initial cluster center for both algorithms. Additionally, the labels for each feature was available so we applied supervised learning techniques. Logistic Regression, Support Vector Machine and Neural Network. Many different optimizers and activation functions were tested for those models in order to find the optimal one and to increase accuracy. All the accuracy measurements were done in the test set while neural network was validated on the test set as well.

### *2.3 Dimensionality Reduction*

Even though the extracted features are way less than the original ones 324 over 3072, respectively, some of them might be less significant. Thus, we applied both PCA and LDA analysis to find out how many principal components could explain the variance with or without the labels included in the analysis. Afterwards, we used the projected matrixes individually and in combination with PCA and LDA features and re-train our models to improve accuracy.

### *2.4 Enrich of the original features*

Lastly, an enrichment to our original dataset was done by rotating each image by $90^o$, $180^o$, $270^o$. Afterwards, we extracted features out of each image and we applied again PCA and LDA. After combining those features, we gave the opportunity to our model to have multiple views over each image.

## 3. Results

Findings showed high fluctuations between the models especially between unsupervised and supervised learning. K-means and GMM showed a poor accuracy to identify data correctly with 10% and almost 9%, respectively, to the extracted features. In contradiction to those, logistic regression had an accuracy of

49.3%. On the other hand, we applied 3 different kernels to SVM such as linear, sigmoid and polynomial. Findings showed that polynomial kernel had the best accuracy with 55.8% over 48.6% and 48.4% for linear and sigmoid, respectively. In Neural Networks 3 different models were created with 1,2 and 3 hidden layers consisted of 10 nodes. In this case, best accuracy occurred in case of 1 hidden layer and the best optimizer seemed to be stochastic gradient descent instead of the other ones.

After dimensionality reduction the best performers used to the new projected arrays based on principal components that explain the variance better. In case of PCA explained variance started flattening after almost 140 principal components. In case of LDA all 9 components were kept as the computational complexity is low enough and there was no need for a trade-off between complexity and accuracy. Finally, results showed that logistic regression had an accuracy of 50.9%, SVM 58.9% and NN 50.4% for PCA features. In case of LDA the results were slightly different, SVM had 49.1%, same as logistic regression and NN had 49.6%.

After the combination of LDA and PCA features to one dataset with 149 features, SVM reached to 59.8% accuracy while others remained almost the same. Lastly, by enriching the dataset with the rotated images, applying PCA and LDA and combining them there was no important differences. In Table 1 the confusion matrix is shown for the final model of SVM with polynomial kernel for the dataset which is contained of PCA and LDA.

*Table 1. Confusion matrix of SVM (59.8% accuracy)*

| | **Airplane** | **Automobile** | **Bird** | **Cat** | **Deer** | **Dog** | **Frog** | **Horse** | **Ship** | **Truck** |
|---|---|---|---|---|---|---|---|---|---|---|
| **Airplane** | **57** | 2 | 5 | 2 | 7 | 2 | 5 | 4 | 14 | 2 |
| **Automobile** | 2 | **72** | 0 | 0 | 5 | 1 | 3 | 1 | 9 | 7 |
| **Bird** | 5 | 1 | **42** | 4 | 8 | 15 | 16 | 4 | 2 | 3 |
| **Cat** | 0 | 3 | 12 | **35** | 16 | 15 | 11 | 5 | 0 | 3 |
| **Deer** | 1 | 1 | 6 | 5 | **73** | 2 | 8 | 1 | 2 | 1 |
| **Dog** | 0 | 2 | 6 | 8 | 17 | **52** | 7 | 7 | 1 | 0 |
| **Frog** | 2 | 3 | 5 | 3 | 8 | 3 | **73** | 2 | 0 | 1 |
| **Horse** | 0 | 1 | 1 | 5 | 17 | 6 | 2 | **63** | 1 | 4 |
| **Ship** | 11 | 8 | 4 | 1 | 3 | 0 | 1 | 0 | **65** | 7 |
| **Truck** | 3 | 9 | 1 | 0 | 10 | 2 | 1 | 2 | 6 | **66** |

## 4. Conclusion

Ultimately, it is shown that all models were slightly better than the benchmark which is 44.68% accuracy. However, after applying dimensionality reduction techniques we were able to reach an accuracy of almost 60%, showing that not only does dimensionality reduction reduce complexity of calculations significantly, but also it empowers our original model. Furthermore, neural networks showed low accuracy, but it was consisted by only 1 layer, while other studies [3] showed that depth when color channels are concerned are highly affected by the depth of a neural network. In another study [4] though, an accuracy of 78.9% was achieved by squeezing the images into smaller objects and applying weights to deep neural networks.

Thus, based on those evidence there is a large area of improvement compared to the accuracy that was achieved by now. Also, it is observed that Bird and Cat had the lowest true positive cases in the confusion matrix (Table 1) which means that maybe weights could be applied in order to reduce that error. Moreover, it is rare the fact of classifying tangible objects (i.e. truck, airplane, ship etc.) as mammals or vice versa which means that the model can identify the general picture but when little differences are concerned it needs additional details. Thus, maybe feature engineering needs to be done, as it was done in Krizhevsky's, (2012) [4] work. Finally, noise could be applied to the pictures with such way that does not change the picture entirely or hide significant information out of it. Based on that the model would rely more on the actual differences and this might increase the accuracy as well.

**References**

[1] Krizhevsky, Alex, Nair Vinod, and Hinton Geoffrey. CIFAR 10 and CIFAR 100 dataset [Online] at: https://www.cs.toronto.edu/~kriz/cifar.html, last visit: December 2019

[2] Stéfan van der Walt, Johannes L. Schönberger, Juan Nunez-Iglesias, François Boulogne, Joshua D. Warner, Neil Yager, Emmanuelle Gouillart, Tony Yu and the scikit-image contributors. *scikit-image: Image processing in Python*. PeerJ 2:e453 (2014) https://doi.org/10.7717/peerj.453

[3] Krizhevsky, Alex & Sutskever, Ilya & Hinton, Geoffrey. (2012). ImageNet Classification with Deep Convolutional Neural Networks. Neural Information Processing Systems. 25. 10.1145/3065386.

[4] Krizhevsky, Alex. (2012). Convolutional Deep Belief Networks on CIFAR-10.