# Introduction to Computer Vision (ECSE 415)
# Assignment 4: Classifiers

Due date: 11:59PM, November 22, 2019

Please submit your assignment solutions electronically via the myCourses assignment dropbox. The submission should include: a single jupyter notebook. More details on the format of the submission can be found below. Submissions that do not follow the format will be penalized **10%**. Attempt all parts of this assignment. The assignment will be graded out of total of **100 points**. Note that you can use any of the OpenCV and scikit-learn functions shown during tutorial sessions for this assignment, unless stated otherwise.

Students are expected to write their own code. (Academic integrity guidelines can be found here).

**Note: Assignments received up to 24 hours late will be penalized by 30%. Assignments received more than 24 hours late will not be graded.**

## Instruction for the Report

1. Submit one single jupyter notebook for the whole assignment. It should contain everything (code, output, and answers to reasoning questions) [1]
2. DO NOT submit dataset along with your code. Explicitly mention how your code is accessing the dataset in the comments.
3. Write answers section-wise. The numbering of the answers should match the questions exactly.
4. Comment your code appropriately.
5. The answers to the reasoning questions should be brief but comprehensive. Unnecessarily lengthy answers will be penalized. Use *markdown cell* type in jupyter notebook to write the answers.
6. If external libraries were used in your code please specify its name and version in the README file.

[1] If you are facing some memory and RAM issues you are free to submit multiple notebooks as long as you clearly mention the reason for it.

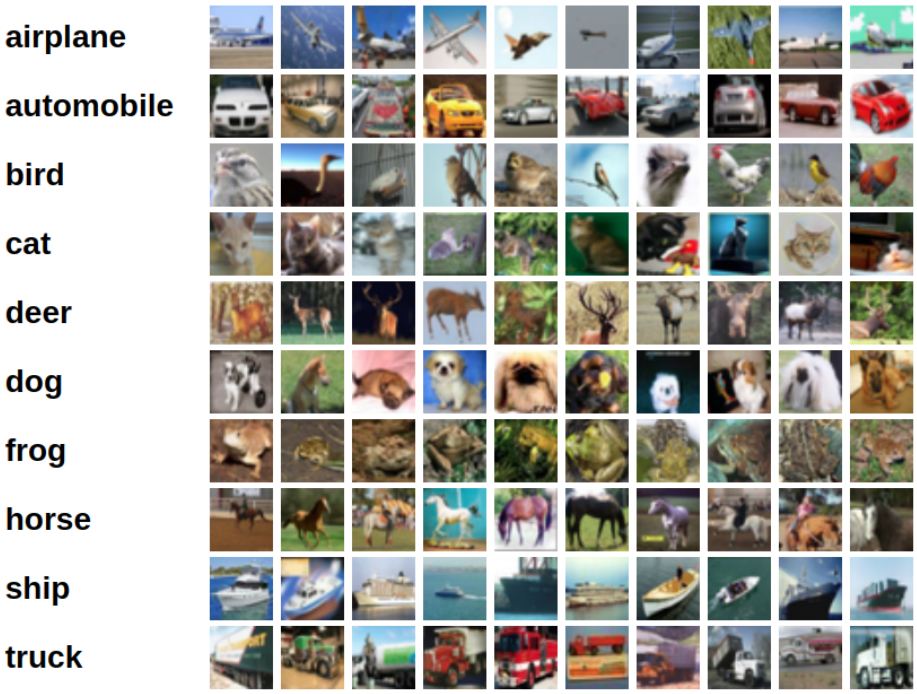


Figure 1: 10 random images from 10 classes of CIFAR10

7. Run the whole jupyter notebook one last time using *Cell-> Run All*. This will run all cells in the jupyter notebook and show all outputs. We expect that when we run your jupyter notebook, we should be able to reproduce your results.

# 1 Dataset Overview

In this assignment, you will explore the classification task. You will use the CIFAR-10 dataset. The dataset can be downloaded from the challenge website using this link (Use CIFAR-10 python version).

This dataset consists of 60,000 32x32 colour images in 10 classes, with 6000 images per class. There are 50,000 training images (5 folds of 10,000 images) and 10,000 test images. The dataset is divided into five training batches and one test batch, each with 10,000 images. The test batch contains exactly 1000 randomly-selected images from each class. The training batches contain the remaining images in random order, but some training batches may contain more images from one class than another. Between them, the training batches contain exactly 5000 images from each class. The classes in the dataset, as well as 10 random images from each are given in Fig:1.

Each batch has following two attributes: Data and Labels. More information about the same is given below.

- **Data:** It contains a 10000x3072 numpy array of uint8s. Each row of the array stores a 32x32 colour image. The first 1024 entries contain the red channel values, the next 1024 the green, and the final 1024 the blue. The image is stored in row-major order, so that the first 32 entries of the array are the red channel values of the first row of the image.
- **Labels:** a list of 10000 numbers in the range 0-9. The number at index i indicates the label of the $i^{th}$ image in the array data.

The dataset contains another file, called batches.meta. It too contains a Python dictionary object. It has the following entries: label_names – a 10-element list which gives meaningful names to the numeric labels in the labels array described above. For example, label_names[0] == "airplane", label_names[1] == "automobile", etc.

Use any one random data batch and convert it into images of size 32x32x3 [2] and display at least 10 images from each class along with their class label. (Use Fig:1 as an example)**(10 points)**

For this task, you will train your classifier on **only one** training batch (data_batch_1) and test it on the test batch, unless specified otherwise.[3]

## Feature Extraction

For this assignment, you will use two different methods for feature extraction.

- Mean intensity values from 3 colour channels (R,G,B)
- Histogram of Oriented Gradients (HoG). Use cell sizes of $8 \times 8$ pixels, and block size of $2 \times 2$ cells and 9 orientation bins. (Suggestion: Make a function which takes list of images as arguments and delivers list of HoG features as output.)

## Classifiers

For this assignment, you will explore the performance of two different classifiers on the task of object classification.

- Support Vector Machines (SVM)

[2] You can refer to the code given on the dataset website to read pickle files.

[3] Make sure you convert data array (10000x3072) into images (10000x32x32x3) before you extract features.

- Random Forest (RF)

**Note: Training time can be upto 10 mins for a single classifier depending on the number of features, number of examples and hyperparameters of the classifier.**

## Evaluation Metrics

Report following two metrics to evaluate the performance of classifiers.

- Accuracy: The number of correct predictions divided by the total predictions.
- Confusion Matrix: A visual representation of misclassification. A 2D histogram of true vs. predicted labels (Refer Tutorial-9).

# 2 SVMs

### 2.1 Linear SVMs - Exploring Features

Extract the two different features and train two different linear SVMs (one for each feature) on the training set. Use gamma='scale' and keep other hyperparameters as default [4]. Which features give better results (in terms of the performance metrics)? Is it expected? Give a hypothetical scenario (i.e. image dataset) in which both feature extraction method can work well for classification? **(15 points)**

### 2.2 Non-linear SVMs - Exploring SVM Kernels

Extract HoG features, and train two different SVM classifiers with two different kernels: (i) Polynomial (ii) Radial Basis Function (RBF). Use gamma='scale' and keep other hyperparameters as default. Compare the accuracy of these two with linear SVM. Which one is giving better results? Is it expected? **(10 points)**

# 3 Random Forest Classifiers

### 3.1 RF - Exploring Performance of HoG Features

Extract HoG features, and train a Random Forest classifier of scikit-learn with following parameters: n_estimators=100, criterion = 'entropy'. Keep all other parameters to default. Report the performance of the classifier. **(5 points)**

[4] Sanity check: if you have implemented everything correctly than you should get around 60% accuracy for HoG features with Linear SVM. We will use this as a baseline and evaluate your implementation. You may or may not get higher accuracy for questions in the following sections.

### 3.2 RF - Exploring Different Number of Trees (estimators)

Extract HoG features, and train RF classifiers with different number of trees (try at least 3 experiments with tress = 40, 200, 250). Write your observations. Do you see an increase in classification performance with an increase in the number of trees? Report the number of trees for which RF gives the best performance. **(15 points)**

## 4 Comparison of Classifiers

### 4.1 Performance Analysis

Describe the performance and the best performing RF classifier (number of trees) and SVM (type of kernel) from the previous sections. Compare both accuracy and Confusion matrices for both classifiers on both training[5] and testing sets. Write your observations. Is there any particular class which is hard to classify for SVM but easier for RF or vice versa? **(10 points)**

### 4.2 Exploring Different Parameters of HoG

Change the cell size, block size, and orientation bins for the HoG features. Try experiments with either SVM (best performing kernel) or RF (best performing trees). Report results for at least 3 different experiments. **(15 points)**

### 4.3 Folded Validation

Take best performing HoG feature parameters, SVM (best performing kernel), and RF (best performing trees) from the previous section and do 3 fold validation on any 3 different training batches (data_batch_1 - data_batch_5). Report accuracies for each fold on the test set for both SVM and RF. Do you see higher accuracy for any particular fold? If yes, what might be the reason. **(10 points)**

### 4.4 Ensemble Classifiers

Build an ensemble classifier that aggregates the results of the 6 classifiers (3 SVM and 3 RF) trained in the previous section. Compare the accuracies and confusion matrices of the ensemble classifiers with the best performing individual classifier. Which one do you expect to perform better? Why? **(10 points)**

[5] To measure the performance on the training set, you train your classifier on the training set and use the same set as the test set and measure the accuracy and the confusion matrix.