# Machine Perception
# COMP3007

# ASSIGNMENT

**Due Date**: Sunday 15 October 2023 at 11:59pm.
**Weight**: **30%** of the unit mark.

**Note**: *This document is subject to corrections and updates. Announcements will be made promptly on Blackboard and during lectures. Always check for the latest version of the assignment. Failure to do so may result in you not completing the tasks according to the specifications.*

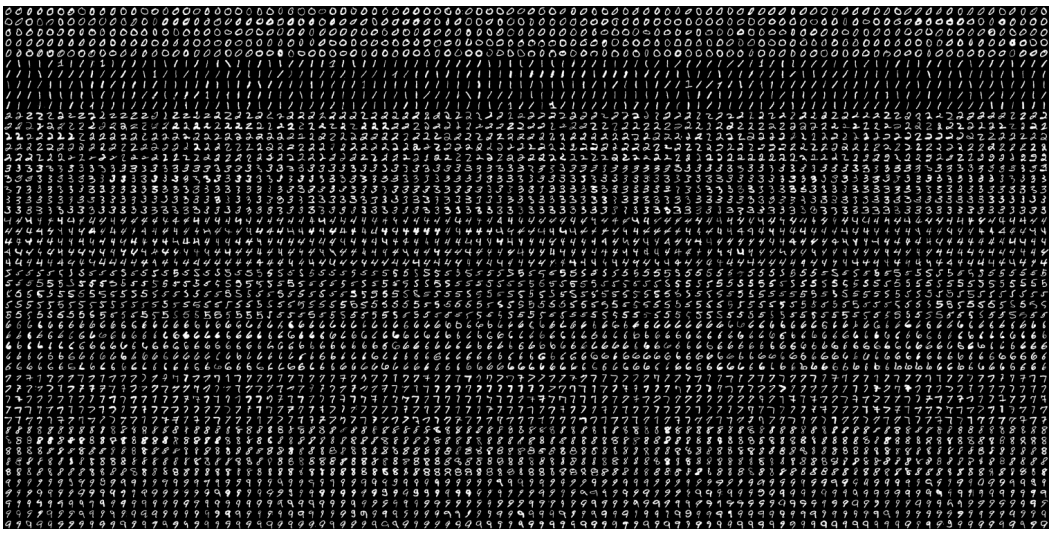
Your total score for this assignment will need to be at least **30 marks** out of the total 100 marks. You will fail this unit if you cannot meet this basic pass mark, regardless of your scores in the mid-semester test and final exam.

## 1 Overview

This assignment provides an opportunity for you to demonstrate how you can use what you have learned from lectures and tutorials to develop computer vision algorithms for image classification problems. For a successful completion of this assignment, you need to understand the fundamental algorithms covered in the lectures, conduct some research into the machine perception problem to design suitable features and classification methods, and use the skills that you have developed through completing practical exercises to build essential components of a computer vision algorithm. External codes are allowed to be used in your assignment, but **the sources must be cited in both your written report and your source codes.** Feel free to use the work you have done in your practical exercises.

A substantial attempt for this assignment is required to pass this unit. A mark of 30% or more is considered a substantial attempt. This means you will not pass this unit if your total mark of this assignment is lower than 30%, even if you achieve full marks in your mid-semester test and final exam.

## 2 The Tasks



Consider the image above which includes 2000 tiny digital images of size 20x20. In this assignment, you are required to 1) extract the tiny digital images and split them into a training dataset (80% of the tiny images) and a testing dataset (20% of the tiny images), and 2) use various machine learning and computer vision methods to classify the tiny digital images and compare their performances. The tasks and requirements are specified as follows.

### 2.1 Task 1: Data Preparation (10 marks)

Extract the tiny images from the big image, split them into training images (80%) and testing images (20%), and write them as images (e.g. jpeg files) and store the training images in the "Train" folder and store the testing images in the "Test" folder.

### 2.2 Task 2: Nearest Neighbor method for image classification (15 marks)

In this task, you will use the raw pixel values and K-nearest neighbor method to classify the tiny images and test how the hyper-parameter K affects the classification accuracy. Use tables to report the accuracies and use confusion matrices to analyze the errors.

### 2.3 Task 3: Linear classifier for image classification (15 marks)

In this task, you will use raw pixel values and linear SVM to classify the tiny images with the one-versus-the-rest method. Compare the performances of linear SVM and k-nearest neighbor methods.

### 2.4 Task 4: Image classification using bag of visual words (30 marks).

In this task, you are required to read the tutorial https://medium.com/@aybukeyalcinerr/bag-of-visual-words-bovw-db9500331b2f and use bag of visual words method to classify the tiny digit images and compare it performances to linear classifiers.

### 2.5 Task 5: Image classification using convolutional neural networks (30 marks)

In this task, you are required to 1) use three or more pretrained CNN models and fine-tuning to classify tiny digital images and compare their performances**.**

## 3. Specifications and Marking Guide

### 4.1 Report: 70%

A written report must be submitted, in PDF format, to Blackboard by the due date. This submission must contain:

- A completed assignment cover sheet
- Printout of your source code
- A document that includes:
  - Statements on how much you have attempted the assignment for each task and summarize the performances and your findings.
  - The detail of your implementation for each task: this must clearly indicate your approach, and how the features you extract, the methods you use for model selection. It must allow the marker to understand how you approach the machine learning tasks. If hyper-parameters need to be selected, you need to split the training dataset into two subsets: one for training and the other for hyper-parameter selection.
  - The performance of your program on the training and the testing datasets.
  - Supporting diagrams, figures, tables that help describe your programs and performance clearly.
  - References that your implementation is based on or inspired from.

Your report will be marked based on 1) the clarity and presentation (20%); 2) the description of your implementation and the judgements of your design (40%); and 3) experimental results on the training data, validation data and testing data, and discussions (40%).

### 4.2 Demo and Implementation: 30%

Your implementation will be marked based on the quality of your code (20%), the demo (30%) and whether your Jupyter Notebook programs work in Google Colab and produce reasonable performance (50%). Your codes are expected to be well written with comments and good structures. In your demonstration, you will be requested to show the predictions with randomly selected tiny digital images.

#### 4.2.1 Evaluation Environment

- Your implementation will be tested using Google Colab.

#### 4.2.2 Your electronic submission

- Your electronic submission of the source code (i.e., Jupyter Notebook) to Blackboard must be a compressed file (zip) with the following naming convention

  [surname] [given names] [student ID].zip

  For example, if your name is Mike Jordan and your student ID is 123456 then your compressed filename is

  jordan mike 123456.zip

- Your electronic submission should contain
  - Source code with Jupyter Notebooks: submit to Assignment – Supplement Materials in Blackboard
  - Your written report: submit to Assignment – Written Report in Blackboard

## 5. Submission

You are required to submit your assignment, including your written report and source code, by Sunday 15-Oct-2023, 11:59pm Perth time.

Upload your submission electronically via Blackboard, under the Assessments section.

You are responsible for ensuring that your submission is correct and not corrupted. You may make multiple submissions, but only your newest submission will be marked.

The late submission policy (see the Unit Outline) will be strictly enforced. A submission 1 second late, according to Blackboard, will be considered 1 day late. A submission 24 hours and 1 second late will be considered 2 days late, and so on.

You must also submit a completed, signed **"Declaration of Originality"** form.

## 6. Academic Misconduct – Plagiarism and Collusion

Please note the following, which is standard across all units in the department:

Copying material (from other students, websites, or other sources) and presenting it as your own work is plagiarism. Even with your own (possibly extensive) modifications, it is still plagiarism.

Exchanging assignment solutions, or parts thereof, with other students is collusion. Engaging in such activities may lead to a grade of ANN (Result Annulled Due to Academic Misconduct) being awarded for the unit, or other penalties. Serious or repeated offences may result in termination or expulsion.

You are expected to always understand this, across all your university studies, with or without warnings like this.

**END OF ASSIGNMENT.**