# COMP3204: Scene Recognition

Adethya Srinivasan
University of Southampton
as37g23@soton.ac.uk

Second Author
Institution2
First line of institution2 address

Second Author
Institution2
First line of institution2 address

## Abstract

*Write an abstract???*

## 1. Run 1

## 2. Run 2

### 2.1. Project Overview

The pipeline consists of four stages:

1. **Feature Extraction:** Breaking images into small pixel patches.
2. **Vocabulary Building:** Learning a dictionary of common visual patterns using K-Means.
3. **Quantization:** Converting images into fixed-length histograms based on the vocabulary.
4. **Classification:** Using Linear Support Vector Machines (SVMs) to predict the scene category.

### 2.2. Detailed Pipeline Steps

#### 2.2.1 Step 1: Feature Extraction

- **Dense Sampling:**
  - We use an $8 \times 8$ patch with a stride length of $4$.
- **Vectorization:**
  - The $8 \times 8$ pixel grid is flattened into a 1-dimensional vector of size 64 ($8 \times 8 = 64$).
- **Patch Normalisation:**
  - We subtract the mean of each patch from itself (mean-centring).
  - We divide the vector by its $L_2$ norm. This scales the vector so it has unit length.

#### 2.2.2 Step 2: Vocabulary Building (K-Means Clustering)

We use K-Means to create a Bag of Visual Words (BoVW).

- **Sampling:** To manage memory usage, we take a random subset of patches from the training images.
- **Clustering (K-Means):**
  - We perform K-Means for a single $k$ value.
- The final $k$ cluster centers become our **"Visual Words"**.

#### 2.2.3 Step 3: Quantization (Image Representation)

We translate every image into a single feature vector of length $k$.

1. **Patch Extraction:** Extract all $8 \times 8$ patches from the image.
2. **Nearest Neighbor Assignment:** Each patch is assigned to the cluster of the closest visual word.
3. **Histogram Generation:** We count how many patches were assigned to each word to construct a histogram.
4. **Histogram Normalisation:** We normalise the resulting histogram.

#### 2.2.4 Step 4: Classification (Linear SVM)

We now have a dataset where every image is represented by a $k$-dimensional vector.

- **Algorithm:** Linear Support Vector Machine.
- **Strategy: One-vs-All (OvA):** We train 15 separate one-vs-all classifiers - one for each class.
- **Prediction:**

  - When testing a new image, all 15 classifiers output a confidence score.
  - The class associated with the highest confidence score is selected as the final prediction.

- **Training Accuracy:** We calculate the training accuracy.
- **Optimising:** We compare this to the existing best training accuracy and/or re-run K-Means for a different $k$ value (we test for $k$ values in the range $[475, 525]$) until all values in the range have been tested.
- **Output:** We retrieve the optimal $k$ value and use this $k$ value on the testing data provided.

## 3. Run 3

## References