Homework 4 Questions

Document Instructions

- 7 questions **[6 + 6 + 6 + 4 + 4 + 4 + 10 = 40 points]**.
- Fill all your answers within the answer boxes, and **please do NOT remove the answer box outlines**.
- Include code, images, and equations where appropriate.
- To identify all places where your responses are expected, search for 'TODO'.
- Please make this document anonymous.

Gradescope Instructions

- When you are finished, compile this document to a PDF and submit it directly to Gradescope.
- You will be required to assign the appropriate answers to the right pages on Gradescope. **Failure to assign pages correctly will lead to a deduction of 2 points per misaligned page (capped at a maxmimum 6 point deduction).**

Q1: [3 × 2 points] The performance of trained classifiers is determined by the data we train them upon, and our perception of that performance is determined by how we evaluate them. One concern is that evaluation using overall accuracy can mask poor performance in specific subgroups caused by dataset bias.

Buolamwini and Gebru's 2018 paper Gender Shades: Intersectional Accuracy Disparities in Commercial Gender Classification found that Microsoft's Face API trained gender classifier achieved 94% overall accuracy, with 100% accuracy on lighter-skinned male faces and 79.2% accuracy on darker-skinned female faces. In response to the report, Microsoft significantly improved their classifier's performance. This occurred through "expanded and revised training and benchmark datasets, new data collection efforts to further improve the training data by focusing specifically on skin tone, gender and age, and an improved classifier to produce higher precision results." These steps helped, but in general it is impossible to completely mitigate all bias in a real-world dataset.

In this homework, we will train a scene classifier using data from Lazebnik et al. 2006. Please review the data to check for potential biases: look at images in the data/train and data/test directories and consider their class labels.

Please list at least three potential biases in the dataset, and describe a potential consequence for each for an application that trained with this data.

For example, if the street images in the dataset were used to train a pedestrian detector for a self-driving car, variation in clothing styles between training data and deployment environments might bias the classifier and fail to detect people. [1–2 sentences each]

TODO: Your answer here.

Q2: [6 points] It can be hard and expensive to find and collect data. One approach that researchers and companies have used is Web scraping, which downloads data across many websites to more easily create large datasets.

Web scraping has come under increased scrutiny as computer vision systems have been deployed in real-world applications, because the technical capability to download publicly-accessible data does not imply consent for the use of that data. In 2021, France found that Clearview AI, a company that operates a facial recognition platform for law enforcement, violated privacy laws by scraping 10 billion images of people's faces from Facebook, YouTube, and other websites without consent.

(a) **[1 point]** How was the Lazebnik et al. 2006 dataset collected?

TODO: Your answer to (a) here.

(b) **[2 points]** Dataset collection might merge multiple sources to fill in gaps in sampling a data distribution. Please find additional scene data that could be added to the Lazebnik et al. 2006 dataset. Here are two places to find datasets: Google and Kaggle.

Find a dataset and provide a URL to it. Given your answers to Q1, describe how the new data addresses gaps in the homework dataset and any limitations of the new data. [3–4 sentences]

TODO: Your answer to (b) here.

(c) **[1 + 2 points]** How was this new data collected? If you had to collect new scene data yourself, how would you do it? [6–7 sentences]

TODO: Your answer to (c) here.

Q3: [6 points]

(a) Define these common terms in machine learning:

(i) **[1 point]** Bias

TODO: Your answer to (a) (i) here.

(ii) **[1 point]** Variance

TODO: Your answer to (a) (ii) here.

(b) Define these terms in the context of evaluating a classifier:

(i) **[1 point]** Overfitting

TODO: Your answer to (b) (i) here.

(ii) **[1 point]** Underfitting

TODO: Your answer to (b) (ii) here.

(c) **[2 points]** How do overfitting and underfitting relate to bias and variance?

TODO: Your answer to (c) here.

Q4: **[4 points]** Suppose we create a visual word dictionary using SIFT and k-means clustering for a scene recognition algorithm. Examining the SIFT features generated from our training database, we see that many are almost equidistant from two or more visual words.

(a) **[2 points]** Why might this affect classification accuracy?

TODO: Your answer to (a) here.

(b) **[2 points]** Given the situation, describe *two* methods to improve classification accuracy, and explain why they would help.
These can be for k-means, or otherwise.

TODO: Your answer to (b) here.

Q5: **[4 points]** The way that the bag of words representation handles the spatial layout of visual information can be both an advantage and a disadvantage.

(a) **[2 points]** Describe an example scenario for each of these cases.

TODO: Your answer to (a) here.

(b) **[1 point]** Describe a modification or additional algorithm which might overcome the disadvantage.

TODO: Your answer to (b) here.

(c) **[1 point]** How might we determine whether bag of words is a good model?

TODO: Your answer to (c) here.

Q6: [4 points] Given a linear classifier such as SVM which separates two classes (binary decision), how might we use multiple linear classifiers to create a new classifier which separates k classes?

Below, we provide pseudocode for a linear classifier. It trains a model on a training set, and then classifies a new test example into one of two classes. Please edit the pseudo-code to convert this into a multi-class classifier.

Hints: See slides in supervised learning crash course deck, plus your own research. You can take either the one vs. all (or one vs. others) approach or the one vs. one approach in the slides; please declare which approach you take.

More hints: Be aware that:

1. The input labels in the multi-class case are different, and you will need to match the expected label input for the `train_linear_classifier` function
2. You need to make a new decision on how to aggregate or decide on the most confident prediction

Note: A more efficient software application would separate the classifier training and testing into two different functions so that the model could be reused without retraining. Feel free to ignore this for now.

TODO: Select the implementation you chose.

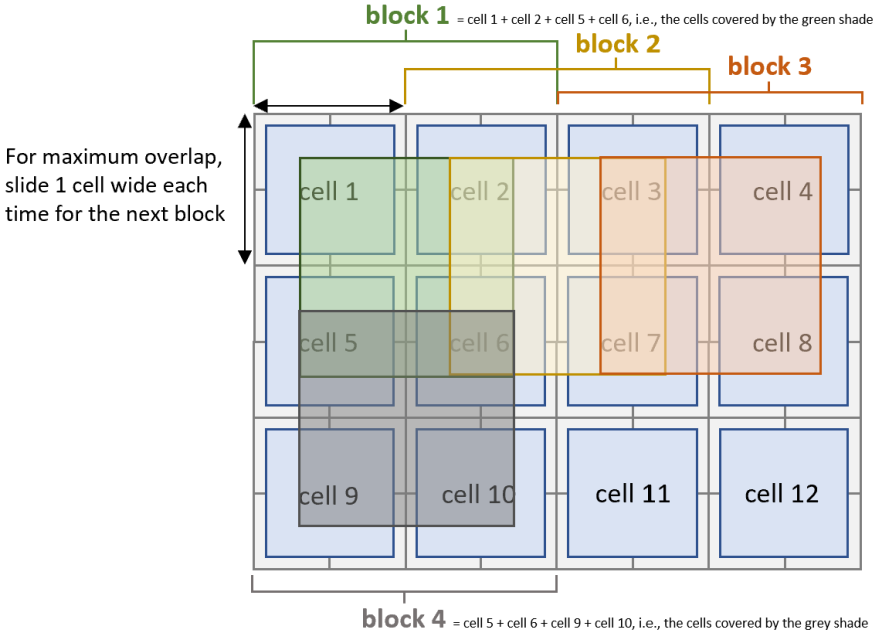
One vs One	☐
One vs Many	☐

```
# Inputs
#   train_feats: N x d matrix of N features each d descriptor long
#   train_labels: N x 1 array containing values of either -1
#                 (class 0) or 1 (class 1)
#   test_feat: 1 x d image for which we wish to predict a label
#
#   Outputs: -1 (class 0) or 1 (class 1)
#
# Inputs:
#    As before, except
#    train_labels: N x 1 array of class label integers from 0 to k-1
# Outputs:
#    A class label integer from 0 to k-1
#

# TODO: Turn this into a multi-class classifier for k classes.
def classify(train_feats, train_labels, test_feat)
    # Train classification hyperplane
    weights, bias = train_linear_classifier(train_feats, train_label)
    # Compute distance from hyperplane
    test_score = weights * test_feats + bias

    return 1 if test_score > 0 else -1
```

Q7: [10 points] In this homework, we will use a feature descriptor called HOG—'Histogram of Oriented Gradients'. As its name implies, it works similarly to SIFT. In classification, we might extract HOG features across the entire image (not just at interest points).

HOG creates one feature descriptor per image 'block'. Each block is split into 'cells' covering pixels. HOG outputs a 9-bin histogram of oriented gradients per cell. We append these together to obtain the feature descriptor for each block. As a result, if we have (z, z) cells per block, the feature descriptor for each block will be of size $z \times z \times 9$. Computing HOG over the whole image, we might produce a matrix where each row is a descriptor, or if we wish to extract a single feature vector per image, we would concatenate all block-level descriptors.

Blocks can overlap as displayed in the diagram below.



(a) **[6 points]** Given a 72×72 image, calculate the number of cells, blocks, and final feature vector size that will occur when we extract HOG features over the whole image with the following parameters using maximum overlap between blocks.

(i) **[3 $\times$ 1 points]** Scenario 1: Pixels per cell = 4×4, cells per block = 4×4

1. Number of cells:

   TODO: Your answer to (a) (i) 1. here.

2. Number of blocks:

   TODO: Your answer to (a) (i) 2. here.

3. Dimensions of feature vector for the whole image:

TODO: Your answer to (a) (i) 3. here.

(ii) **[3 $\times$ 1 points]** Scenario 2: Pixels per cell = 8×8, cells per block = 2×2.

1. Number of cells:

   TODO: Your answer to (a) (ii) 1. here.

2. Number of blocks:

   TODO: Your answer to (a) (ii) 2. here.

3. Dimensions of feature vector for the whole image:

   TODO: Your answer to (a) (ii) 3. here.

(b) **[2 + 2 points]** When using HOG, the parameters such as pixels per cell and cells per block impact the resulting feature descriptor and so our performance on a classification task.

What are the pros and cons of the two parameter combinations? Which might you expect to have better performance?

Feedback? (Optional)

Please help us make the course better. If you have any feedback for this assignment, we'd love to hear it!