CW1

Song Liu (song.liu@bristol.ac.uk)

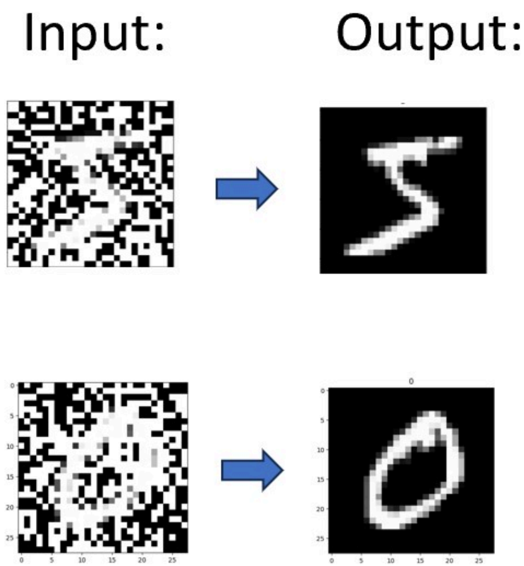
GA 18, Fry Building,

Microsoft Teams (search "song liu").

Assessed Coursework 1

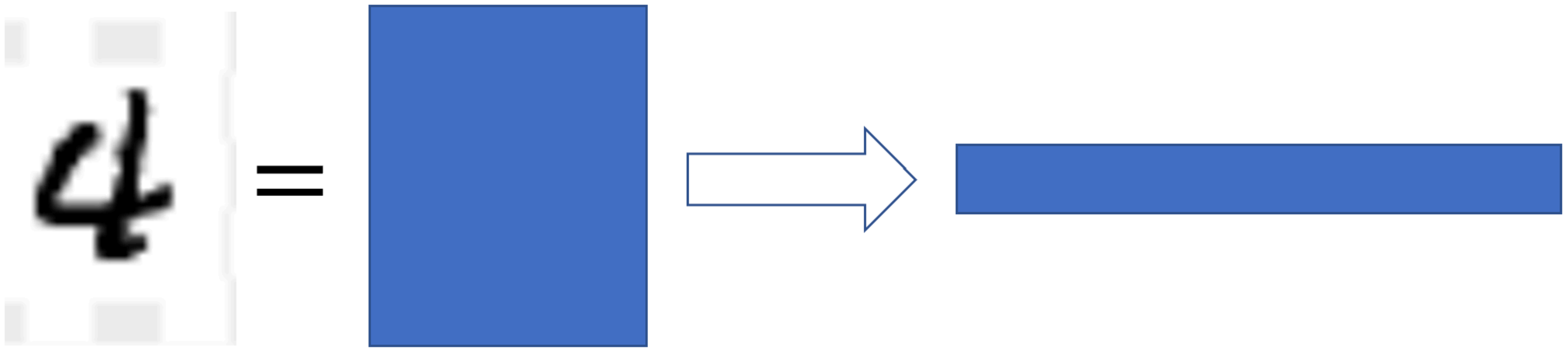
Deadline: Monday, 28th Oct.

Task: Given a data set (MNIST) containing images of handwritten digits, implement a k-nearest neighbor algorithm, which recovers an image from a corrupted version of it.



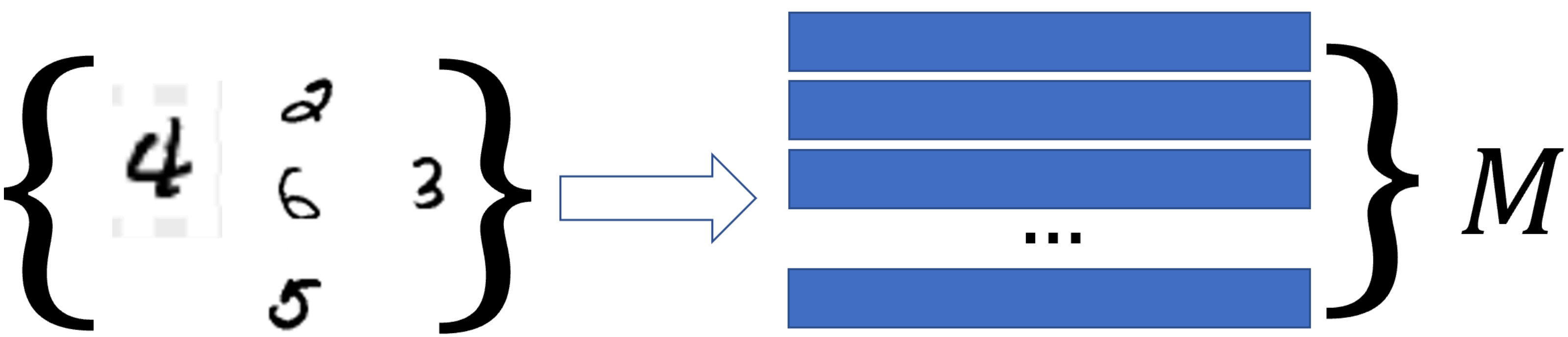
Assessed Coursework 1

- As we mentioned in previous lectures/labs, images are stored as flattened matrices (in row major order) in the memory.
- Each image is stored as a vector in this coursework.



Assessed Coursework 1

- In this CW, we are dealing with sets of images. Stacking all the image vectors together, you get a matrix.

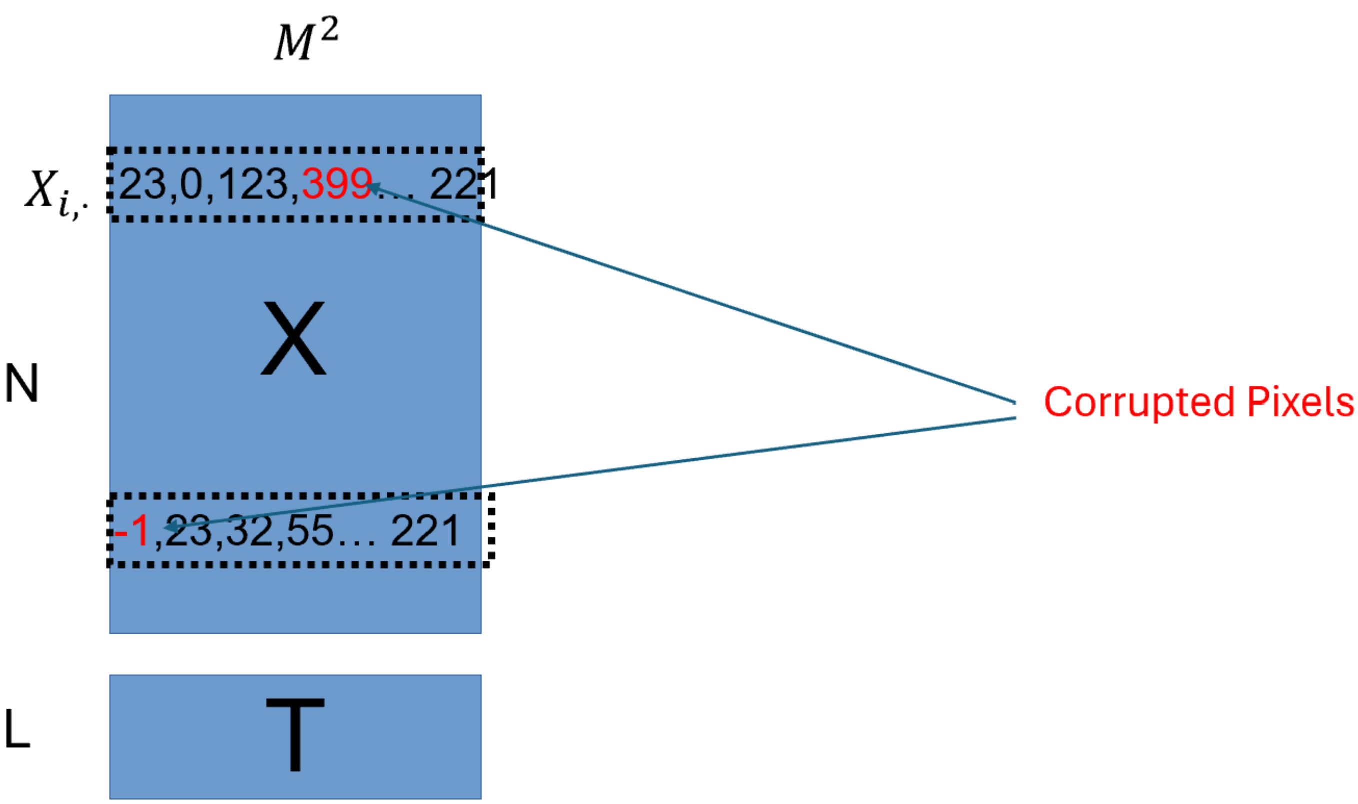


- Specifically, there are two sets of images in this CW, so they are represented by two matrices X, T.

Part I, The Data Set

- CW folder contains 2 `.matrix` files storing 2 matrices.
  - `X.matrix` stores an N by M^2 matrix X where each row is a grayscale **corrupted M by M image** stored in row-major order.
    - The corrupted pixels have values that is out of $[0, 255]$
  - `T.matrix` stores an L by M^2 matrix T where each row is an M by M **clean image** in row-major order.

Part I, Data Structure



Part I, Loading Dataset (15pt)

- The code for loading these matrices from files have been provided to you. Matrices are represented by a **matrix structure** in this coursework.

```
struct matrix
{
    int numrow; //number of rows
    int numcol; //number of columns
    int *elements; // pointer pointing to an integer array
    // storing all entries in the matrix in row major order.
};
typedef struct matrix Matrix;
// now "Matrix" is an alias of "struct matrix"
```

- By simply running the skeleton code, you should see some basic statistics of X and T .
  - What are M, N and L?

Part I, Loading Dataset (15pt)

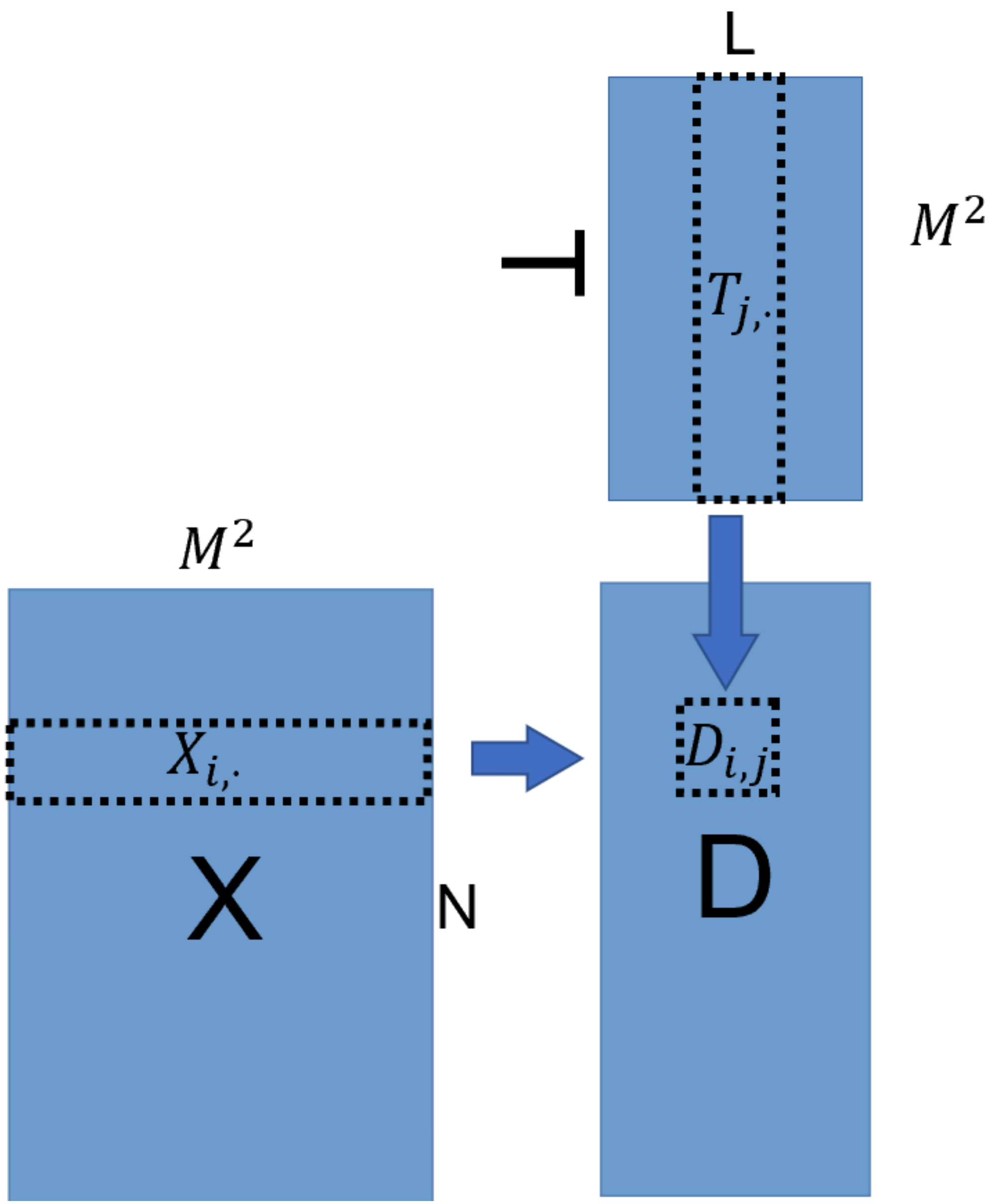
- Plot the first 5 images in X.
- in the following format:

```
Below are five corrupted images:
Image 0:
...
```

Part II Computing Distance Matrix D (25pt in total)

- Construct an N by L matrix D, where the i, j-th element $D_{ij} = \text{dist2}(X_i, T_j)$
  - X_i is the i-th row of X
  - T_j is the j-th row of T.
- $\text{dist2}(a, b)$ computes the squared euclidean distance between two vectors a and b, excluding elements that are corrupted,
  - $\text{dist2}(a, b) := \sum_{k: a_k, b_k \in [0,255]} (a_k - b_k)^2$.

Part II (Computing D)



Part II.1 Constructing D (10pt)

Before your `main` function,

1. Write a few helper functions:

   ```
   int get_elem(Matrix M, int i, int j)
   ```

   - returns the `i, j` th element of matrix `M`.

   ```
   void set_elem(Matrix M, int i, int j, int value)
   ```

   - assign `value` to the `i, j` th element of matrix `M`
     In this coursework, `i, j` are **zero-based indices**.

In your `main` function,

2. Allocate HEAP memory for D.
3. Declare and initialize a new `matrix` variable `D`.

Part II.2 Computing D (15pt)

Now, populate the matrix D with correct values.

- **Hint**: Compare the computation of D and the matrix multiplication. What are the similarities and what are the dissimilarities?
    - Can you modify the matrix multiplication code to compute matrix D?
- **Hint**, you can write a function

```
void pairwise_dist(Matrix T, Matrix X, Matrix D)
```

    - where D is the output, storing the outcome.
- Partial points will be given for correctly written code for computing $\mathrm{dist2}(a, b)$.

Part III. Restoring the Images (30pt)

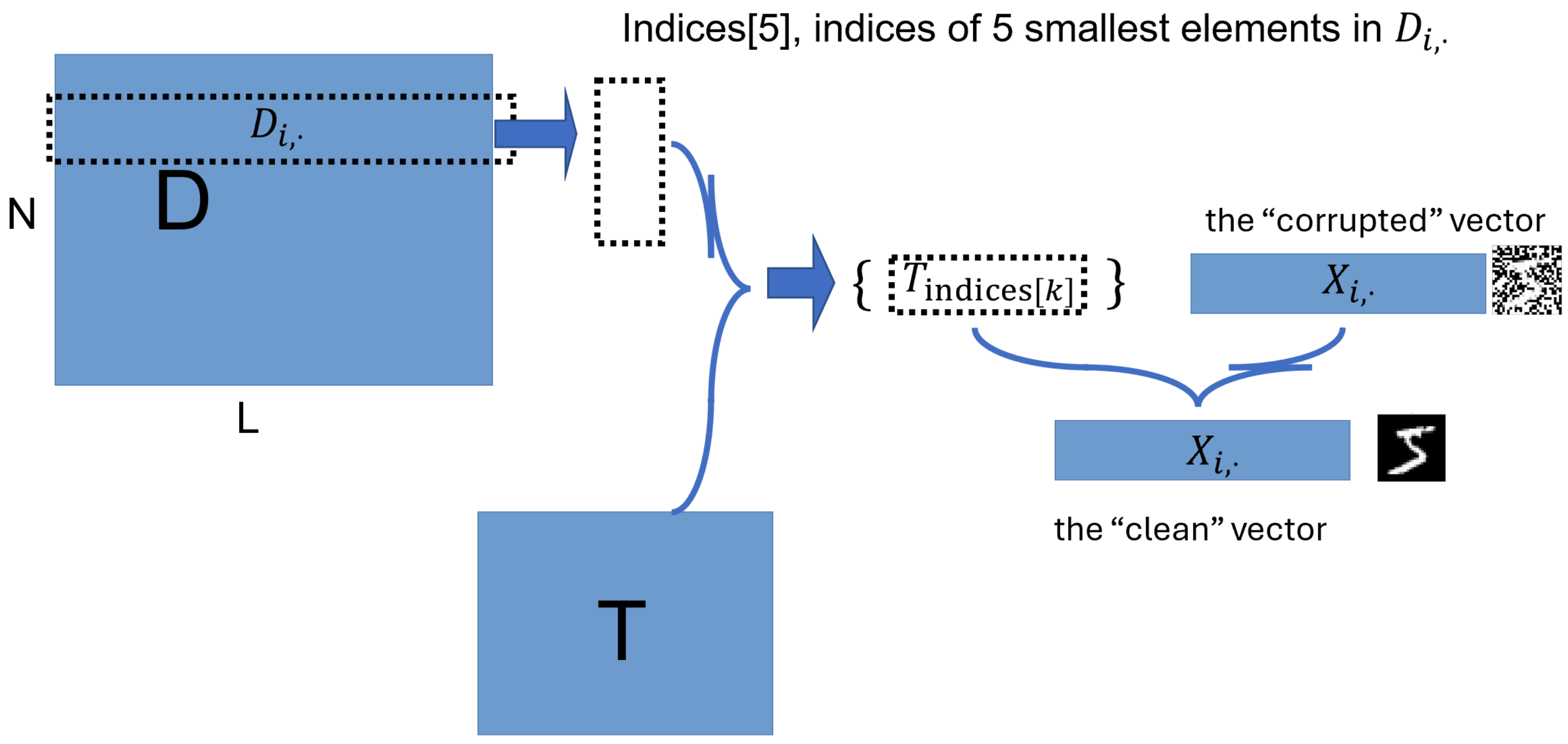
- For each row of matrix D, find **the indices** of the five smallest elements.
    - Suppose the array `indices` contains the indices of the five smallest elements in row i of matrix D.
    - For each **corrupted** column j in the row vector X_i:
        - Set X_{ij} to be v_j,
        - where v is the average of row vectors $\{T_{\text{indices[k]}}\}_{k=0}^{4}$.

Part III. Restoring the Images (30pt)

- For example:
  - Suppose the i-th row of D is $[3, 2, 5, 1, 2, 5, 13, 46, 32]$,
  - The indices of the five smallest elements are $[3, 1, 0, 5, 6]$.

Part III. Restoring the Images

For each row in D, do:



Part III Helper Function

- Hint: Write a helper function

  ```c
  void minimum5(int len, int a[], int indices[])
  ```

  It takes an array `a` with length `len` as input, then fills `indices[]` with the indices of the five smallest elements.

- You might want to test your functions in a separate c file to ensure that they are correctly written.

Part III. k-Nearest Neighbour Algorithm

- At each row D_i, you guess the corrupted pixels in X_i using 5-nearest neighbour algorithm.
  - https://en.wikipedia.org/wiki/K-nearest_neighbors_algorithm
- After your guess, print the first 5 restored image X_i to the console for validation, in the following format:

```
Below are five restored images:
Image 0:
...
```

Final Project: Marking Criteria

- Submitting correct code (10%)
    - Submitting a C file with **the correct name**.
    - Your code compiles and runs **without major error** such as **crash**, **infinite loop**.
        - It will be tested using `gcc` in the lab pack.
- Part I 15%
- Part II 25% (10% + 15%)
- Part III 30% (10% + 20%)
- Good Coding Practice (20%)
    - Good code format
    - Good variable naming scheme.
    - Apt comments

Final Project: Dos and Don'ts

- You can discuss with your classmates about general strategies but write your own code!
- Don't give your code to other students.
- Review relevant previous lab sessions before you start.
- You need to add a reference in the comments. Copy and pasting code from internet (including chatGPT) without citation is not allowed.
- You are only allowed to use standard features of C.
  - You can use `stdio.h`, `stdlib.h`, `limits.h` and `math.h`.
  - If you want to use other libraries, consult with the lecturer or TA beforehand.

Final Project: Q&A

- We will answer questions posted on the Blackboard forum or answering them during the lab sessions.
    - https://www.ole.bris.ac.uk/ultra/courses/_259201_1/engagement
- We will inspect the forum regularly and try to respond in 24 hours.