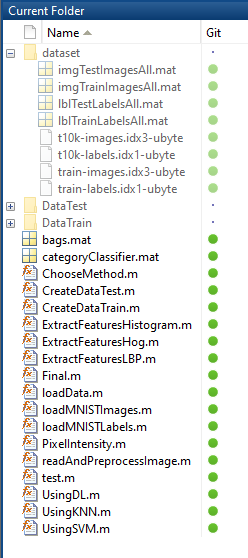
**BÁO CÁO ĐỒ ÁN CUỐI KỲ**

**NHẬN DẠNG THỊ GIÁC VÀ ỨNG DỤNG**

1. **Links**
   1. Github: <https://goo.gl/uQAosj>
   2. Youtube:
2. **Tóm tắt**
3. **Mục tiêu**
4. **Nội dung**
5. **Cách thực hiện**
6. **Nội dung chi tiết**
7. **Cấu trúc chương trình**

* Cấu trúc chương trình gồm thư mục chứa bộ dữ liệu chữ số viết tay được đặt trong thư mục **dataset** gồm bộ dữ liệu test và bộ dữ liệu huấn luyện.
* Các thư mục **DataTest** và **DataTrain** chứa bộ dữ liệu chữ số viết tay được sắp xếp vào trong các thư mục tương ứng với nhãn theo đúng định dạng tên ảnh là thứ tự ảnh “**image\_xxxxx.jpg**” trong bộ dữ liệu.
* **bags.mat** chứa kết quả trung gian trong việc rút trích đặc trưng bag of features.
* **categoryClassifier.mat** chứa kết quả trong quá trình xử lý phân loại ảnh huấn luyện vào trong các lớp tương ứng.
* **ChooseMethod.m** là phương thức cho phép người dùng lựa chọn các phương pháp huấn luyện dữ liệu gồm KNN, SVM và Deep Learning.
* Các tập tin ExtractFeatures có nhiệm vụ rút trích các đặc trưng theo Histogram, Hog, LBP nhằm phục vụ qua việc huấn luyện dữ liệu.
* **Final.m** chứa chương trình chính của đồ án cuối kỳ.
* **loadData.m** là phương thức nạp các bộ dữ liệu chữ số viết tay vào trong chương trình để xử lý, ta có thể nạp dữ liệu bằng cách cung cấp tên bộ dữ liệu cần nạp.
* **loadMNISTImages.m** và **loadMNISTLabels.m** là thư viện MNIST để đọc dữ liệu chữ số viết tay.
* **CreateDataTest.m** và **CreateDataTrain.m** chứa phương thức tạo ra hai bộ dữ liệu phát sinh từ dữ liệu chữ số viết tay vào trong hai thư mục DataTest và DataTrain.

1. **Mô tả features – machine learning methods**
   1. **Dữ liệu thô - Raw (pixel intensity)**

Pixel intensity sử dụng đặc trưng histogram để huấn luyện cho bộ dữ liệu với tham số **Bin** được người dùng cung cấp sử dụng **fitcknn** để huấn luyện dữ liệu với các tham số **“NumNeighbors”** (mặc định k = 1) cho phép xác định số lượng láng giềng gần nhất để phân lớp khi dự đoán. Bên cạnh đó cho phép người dùng lựa chọn **“Standardize”** (mặc định false)để chuẩn hoá các dự đoán bằng cách tuỳ chỉnh các cột dữ liệu dự báo theo giá trị trung bình cột và độ lệch chuẩn tương ứng.

Cú pháp:

**Mdl = fitcknn(featureDataTrain', lblDataTrain, 'NumNeighbors', [value], 'Standardize', [true/false]);**

**Bảng kết quả RAW sử dụng fitcknn**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Bin** | **K** | **Standardize** | **Mẫu đúng** |
| 1 | 16 | 1 | false | 2635 |
| 2 | 64 | 1 | false | 2799 |
| 3 | 128 | 1 | false | 2870 |
| 4 | 256 | 1 | false | **3039** |
| 5 | 16 | 2 | false | 2632 |
| 6 | 64 | 2 | false | 2777 |
| 7 | 128 | 2 | false | 2829 |
| 8 | 256 | 2 | false | 2908 |
| 9 | 16 | 2 | true | 2583 |
| 10 | 64 | 2 | true | 2420 |
| 11 | 128 | 2 | true | 2115 |
| 12 | 256 | 2 | true | 1674 |
| 13 | 16 | 3 | false | 2668 |
| 14 | 64 | 3 | false | 2874 |
| 15 | 128 | 3 | false | 2829 |
| 16 | 256 | 3 | false | **3039** |
| 17 | 16 | 3 | true | 2667 |
| 18 | 64 | 3 | true | 2430 |
| 19 | 128 | 3 | true | 2075 |
| 20 | 256 | 3 | true | 1651 |

**Bảng kết quả RAW sử dụng SVM**

|  |  |  |
| --- | --- | --- |
| **STT** | **Bin** | **Mẫu đúng** |
| 1 | 16 | 2635 |

**Bảng kết quả RAW sử dụng Deep Learning**

|  |  |  |
| --- | --- | --- |
| **STT** | **Bin** | **Mẫu đúng** |
| 1 | 16 | 2635 |

* 1. **Histogram of Oriented Gradients (HOG)**

Đặc trưng HoG dựa vào sự phân bố về cường độ và hướng điểm xám trên ảnh với các tham số **“CellSize”** cho biết kích thước ô để tính giá trị HoG (mặc định [8 8]), nếu giá trị nhỏ thì có thể gây mất mát thông tin khi rút trích đặc trưng; **“BlockSize”** cho biết số lượng cell cấu thành một khối (mặc định [2 2]), nếu kích thước quá lớn có thể làm giảm khả năng ngăn chặn những thay đổi sáng cục bộ trên ảnh, bởi vì nếu số lượng điểm ảnh trong một khối lớn, có thể làm mất giá trung bình do đó, giá trị “**BlockSize**” nhỏ có thể hỗ trợ ngăn chặn những thay đổi sáng cục bộ khi rút trích đặc trưng; bên cạnh đó **“NumBins”** cho biết số lượng bin của đặc trưng.

**Bảng kết quả sử dụng fitcknn**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **STT** | **CellSize** | **BlockSize** | **NumBins** | **K** | **Standardize** | **Mẫu đúng** |
| 1 | 8 | 2 | 9 | 1 | 0 | 9709 |
| 2 | 8 | 2 | 9 | 2 | 1 | 9720 |
| 3 | 8 | 2 | 4 | 1 | 0 | 9631 |
| 4 | 8 | 2 | 4 | 2 | 1 | 9632 |
| 5 | 4 | 2 | 9 | 1 | 0 | **9767** |
| 6 | 4 | 2 | 9 | 2 | 1 | 9727 |
| 7 | 4 | 2 | 4 | 1 | 0 | 9717 |
| 8 | 2 | 2 | 9 | 1 | 0 | 9772 |
| 9 | 2 | 2 | 9 | 2 | 1 | 9536 |
| 10 | 2 | 2 | 4 | 1 | 0 | 9754 |

**Bảng khảo sát với SVM**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **CellSize** | **BlockSize** | **NumBins** | **Mẫu đúng** |
| 1 | 8 | 2 | 9 | 9794 |
| 2 | 8 | 2 | 4 | 9721 |
| 3 | 4 | 2 | 9 | **9883** |
| 4 | 4 | 2 | 4 | 9861 |
| 5 | 2 | 2 | 9 | 9874 |
| 6 | 2 | 2 | 4 | 9852 |

* 1. **Local Binary Pattern (LBP)**

Đặc trưng LBP dùng để đo độ tương phản cục bộ trong ảnh với các tham số **“****NumNeighbors”** để tính toán giá trị LBP cho mỗi pixel của ảnh nguồn (mặc định 8); **“Radius”** dùng để lựa chọn lân cận của mỗi pixel ảnh nguồn (mặc định 1, giá trị từ 1 🡪 5); **“CellSize”** cho biết kích thước ô tính giá trị LBP (mặc định Size(I)) và **“Normalization”** để xác định loại dạng chuẩn cho mỗi biểu đồ xám trên ô tính giá LBP, có hai giá trị “L2” hay “None”.

**Bảng kết quả sử dụng fitcknn (k = 1 & Standardize = 0)**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **STT** | **NumNeighbors** | **Radius** | **CellSize** | **Normalization** | **K** | **Standardize** | **Mẫu đúng** |
| 1 | 8 | 1 | 8 | L2 | 1 | 0 | 9156 |
|  | 8 | 1 | 8 | None | 1 | 0 | 9252 |
|  | 8 | 1 | 4 | L2 | 1 | 0 | 9515 |
|  | 8 | 1 | 4 | None | 1 | 0 |  |
|  | 8 | 2 | 8 | L2 | 1 | 0 |  |
|  | 8 | 2 | 8 | None | 1 | 0 |  |
|  | 8 | 2 | 4 | L2 | 1 | 0 |  |
|  | 8 | 2 | 4 | None | 1 | 0 |  |
|  |  |  |  |  |  |  |  |

**Bảng kết quả sử dụng SVM**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **NumNeighbors** | **Radius** | **CellSize** | **Normalization** | **Mẫu đúng** |
| 1 | 8 | 1 | 8 | L2 |  |
| 2 | 8 | 1 | 8 | None |  |
| 3 | 8 | 1 | 4 | L2 |  |
| 4 | 8 | 1 | 4 | None | 9845 |
| 5 | 8 | 2 | 8 | L2 |  |
| 6 | 8 | 2 | 8 | None |  |
| 7 | 8 | 2 | 4 | L2 |  |
| 8 | 8 | 2 | 4 | None |  |
| 9 | 16 | 1 | 8 | L2 |  |
| 10 | 16 | 1 | 8 | None |  |
| 11 | 16 | 1 | 4 | L2 |  |
| 12 | 16 | 1 | 4 | None |  |
| 13 | 16 | 2 | 8 | L2 |  |
| 14 | 16 | 2 | 8 | None |  |
| 15 | 16 | 2 | 4 | L2 |  |
| 16 | 16 | 2 | 4 | None |  |

**Bảng kết quả sử dụng Deep Learning**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **NumNeighbors** | **Radius** | **CellSize** | **Normalization** | **Mẫu đúng** |
| 1 | 8 | 1 | 8 | L2 |  |

* 1. **Bag of Words (BOW)**

Bag of Words coi những thành phần trên ảnh là một đặc điểm hay đặc trưng để phân loại với các tham số **“VocabularySize”** cho biết số lượng visual words trong đối tượng (mặc định 500); **“****PointSelection”** cho phép chọn phương pháp xác định vị trí điểm cho rút trích đặc trưng SURF (mặc định Grid, giá trị khác là Detector); **“****GridStep”** cho biết kích thước grid step và chỉ có giá trị khi **“****PointSelection”** là Grid và không sử dụng *CustomExtractor*.

**Bảng kết quả sử dụng fitcknn**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **STT** | **VocabularySize** | **PointSelection** | **GridStep** | **PointSelection** | **K** | **Standardize** | **Mẫu đúng** |
| 1 |  |  |  |  | 1 | 0 |  |

**Bảng kết quả sử dụng SVM**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **STT** | **VocabularySize** | **PointSelection** | **GridStep** | **PointSelection** | **Mẫu đúng** |
| 1 |  |  |  |  |  |

**Bảng kết quả sử dụng Deep Learning**

* 1. **Deep Learning**

1. **Kết quả thực nghiệm và tham số**
2. **Báo cáo mở rộng**