**ЛАБОРАТОРНАЯ РАБОТА «2D РАСПОЗНАВАНИЕ ЛИЦА»**

**Исходные данные**:

* База данных фотографий лиц людей азиатской наружности в количестве по 5 фотографий на каждого. Группы именуются по маске \d\d\d, фотографии – по маске \d\d\d\\_\d\d, где первые три цифры – префикс группы.

**Задача**:

* Подобраться алгоритм и построить графики FAR и FRR для системы.

**Ход работы:**

1. Ввиду ограниченного кол-ва времени было решено ограничиться половиной базы данных фотографий (однако были проведены попытки работы и с полным объемом – программа работает). Каждую группу фотографий делим 50/50 на данные для обучения (3 шт.) и данные для теста (2 шт.).
2. Все тестовые фотографии путем работы скрипта перемещаем в отдельный каталог.
3. Ищем/создаем свое ПО для распознавания лица. Мой выбор пал на написание GUI приложение на языке C# с применением библиотеки Emgu.CV (<http://www.emgu.com/wiki/index.php/Main_Page>). За основу была выбрана заготовка программы отсюда - <https://www.codeproject.com/Articles/239849/Multiple-face-detection-and-recognition-in-real> и существенно переработана.
4. Имающийся алгоритм базируется на признаках Хаара для детектирования лица на фотографии и на алгоритме распознавания Eigenface.
5. Алгоритм работы программы:
   1. *Обучающий этап*: все обучающие изображение подвергаем переработке, в ходе которой на фото определяется область с лицом человека, далее эта область извлекается с общей фотографии, переводится в черно-белый цвет и сохраняется в отдельной директории в размере 200x200 пикселей. Отдельно в текстовый файл записываются ярлыки изображений (т.е. их имена) – признаки детектирования верных/неверных распознаваний. В дальнейшем, эти фото и ярлыки будут использоваться в качестве имитации зарегистрированных в системе пользователей.
   2. *Работа алгоритма*: ПО устроено таким образом, что в ответ на загруженное пользователем изображение выдает ярлык (строку) наиболее похожего изображения, если параметр eigenDistanceThreshold находится в допустимых пределах, иначе – пустую строку. Стало быть, для наших целей можно формализовать такой вывод программы:
      1. *FAR* – если алгоритм выдал ярлык изображения из другой группы (пропустил, но не того, кого надо).
      2. *FRR* – если алгоритм не выдал ответа (когда к нему обратился зарегистрированный пользователь).
   3. *Процесс*: обработка происходит по всем фотографиям в группе теста. Имена файлов в обучающей выборке и в группе теста оставлены без изменений. Формально, процесс выглядит так:
      1. Подается тестовое изображение на вход. Алгоритм выдает ярлык максимально похожего изображения, либо пустую строку. Если пустая строка – увеличиваем значение FRR, если префикс группы ярлыка != префиксу группы тестового изображение – увеличиваем значение FAR.
6. Результат работы программы представлен в таблице:

|  |  |  |  |
| --- | --- | --- | --- |
| MAX THRESHOLDING | FAR | CORRECT | FRR |
| 1000 | 0 | 0 | 483 |
| 2000 | 0 | 7 | 477 |
| 3000 | 32 | 54 | 398 |
| 4000 | 162 | 114 | 208 |
| 6000 | 321 | 147 | 16 |
| 7000 | 327 | 150 | 7 |
| 9000 | 332 | 151 | 1 |
| 10000 | 333 | 151 | 0 |
| 12000 | 333 | 151 | 0 |

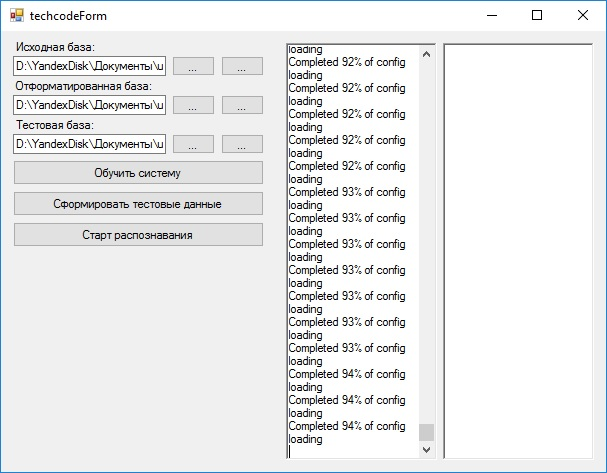
Диаграммы зависимостей FAR и FRR от порогового значения имеют вид:

(расчёт приведен в документе graph.xls)

Директория с обработанными извлечёнными фотографиями лиц:

|  |  |
| --- | --- |
|  |  |

**Руководство по работе с программой:**

1. Необходимо заполнить пути требуемых директорий с исходными изображениями, отформатированными изображениями и тестовыми изображениями.
2. После этого нажать на кнопку «Обучить систему». Если запуск происходит в первый раз и папка с обработанными изображениями пустая, то появится соответствующее сообщение. В таком случае, нажимаем на кнопку «Сформировать тестовые данные» и дожидаемся окончания операции.
3. После этого можно нажимать на кнопку «Старт распознавания». По завершению операции в окне вывода появятся посчитанные значения FAR, FRR, CORRECT и таблица вида ярлык\_тестового\_изображения: ярлык\_выданный\_алгоритмом lastDistance, где lastDistance – точность, с которой распозналось изображение.