Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение

высшего образования

«Пермский национальный исследовательский политехнический университет»

Кафедра информационных технологий и автоматизированных систем

**Лабораторные работы**

по дисциплине **«Современные проблемы информатики и вычислительной техники»**

**Выполнили**

студенты гр. АСУ2-16-1м, АСУ3-16-1м

Климов Сергей Николаевич

Казанцев Александр Сергеевич

Федосеева Ольга Николаевна

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*(подпись студентов)*

**Проверил** доцент кафедры ИТАС

Курушин Даниил Сергеевич

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*(отметка о зачете)*

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*(дата, подпись преподавателя)*

Пермь 2016

**Содержание**

[Лабораторная работа №1 «Нейронные сети» 3](#_Toc463470752)

[1. Постановка задачи 3](#_Toc463470753)

[2. Ход решения 3](#_Toc463470754)

[3. Алгоритм программирования 4](#_Toc463470755)

[4. Результаты экспериментов 6](#_Toc463470756)

[5. Заключение 9](#_Toc463470757)

[6. Список используемых источников 10](#_Toc463470758)

# Лабораторная работа №1 «Нейронные сети»

1. **Постановка задачи**

Реализовать многослойный персептрон для следующей задачи: на вход нейронной сети подается изображение лица человека 20х20 пикселей, необходимо отличить изображения женщин от изображений мужчин.

1. **Ход решения**

Для реализации нейронной сети было установлено следующее программное обеспечение: Python3, OpenCV, Pybrain, Eric IDE.

Процесс решения поставленной задачи поделен на следующие этапы:

1. Установка ПО и подключение к Pytnon3 необходимых библиотек;
2. Подготовка обучающей выборки. Выборка содержит 213 изображений мужчин и 83 изображений женщин. Выборка получена из открытой базы данных FERET [3]. Полученные из базы изображения необходимо обработать и обрезать до размера 20х20 пикселей.
3. Создание нейронной сети. Многослойный персептрон состоит из 3 слоев, имеет 1596000 входов (10\*400\*399), один выход (женский пол, мужской пол).
4. Написание программы для обучения выборки и ее тестирование (см. пункт 3. Алгоритм программирования);
5. Тестирование программы.
6. Оценка полученных результатов.

1. **Алгоритм программирования**

Первоначальной задачей является подготовка исходных данных для работы. С помощью библиотеки OpenCV с применением функции face\_cascade.detectMultiScale(img, 1.3, 5), которая использует признаки Хаара определяем лица на фотографиях. После чего изображение обрезается и ужимается до размера 20х20 пикселей с помощью функции cv2.imwrite('20\_' + sys.argv[1] , resized\_img). Полученное скорректированное изображение сохраняется в файл. Данный скрипт применяется ко всем фотографиям при помощи shell.

Для упрощения дальнейшего обучения и тестирования нейронной сети, полученные изображения были вручную отсортированы и размещены в разных папках (Female, Male). Изображения мужского лица были поименованы с префиксом f, женские – m.

Входными данными для нейронной сети являются характеристики изображения, представляющие собой разность яркости каждой пары пикселей изображения. Данная разность сравнивается с некоторыми значениями, [2]:

│Pixeli – Pixelj │> 0

│Pixeli – Pixelj │ < 5

│Pixeli – Pixelj │ < 10

│Pixeli – Pixelj │ < 25

│Pixeli – Pixelj │ < 50

Таким образом, мы получаем вектор характеристик [0 0 0 1 1]. К этим пяти характеристикам добавляются обратные им характеристики – [1 1 1 0 0]. Объединяя эти векторы, получаем набор двоичных характеристик для пары пикселей. Получившиеся характеристики записываются в список, который далее будет передаваться нейронной сети как входные данные, а также использоваться для обучения. Так как наши изображения имеют размер 20х20 пикселей, список имеет 399 х 400 х 10 (1 596 000) значений.

Работа с подготовленными ранее изображениями осуществляется функцией do\_train\_dataset следующим образом: каждый пиксель изображения загружается в градациях серого и помещается в список, который передается в функцию pixel\_camparision(img), которая возвращает список с готовыми характеристиками. Получившийся список добавляется в структуру dataset, необходимую для тренировки нейросети, в которую и передается в виде первого значения. Второй значение – требуемый результат реакции нейронной сети на изображение (женское лицо – 0, мужское – 1). В результате своей работы функция возвращает обучающую выборку (dataset) в функцию main.

Главная функция представляет собой следующий набор действий: первым делом строится нейронная сеть с помощью net = buildNetwork(1596000, 3, 1), где первый параметр – число входов, второй – число скрытых слоев, третий – число выходов. Далее указываем, что сеть обучается методом «с учителем»: dataset = SupervisedDataSet(1596000, 1). После чего главная функция получает обучающую выборку из dataset = do\_train\_dataset(dataset). В следующем действии указываем, что сеть тренируется методом обратного распределения ошибки. После чего происходит непосредственно процесс обучения с помощью функцией train. [5] Процесс обучения может происходить либо по указанному числу эпох, либо до того момента пока ошибка не сводится к нулю. После проведения обучения программа выводит значение получившейся ошибки. Далее происходит тестирование работы с получившеюся нейронной сетью. На вход сети подается список характеристик изображений, не участвующих в обучении. После чего сеть выдает результат своей работы, что происходит при помощи функции net.activate(list). По получившимся данным оценивается результат работы нейронной сети.

Основная идея второго метода – нахождение на лице глаз, носа, рта и определение расстояния между данными частями. Получившиеся результаты являются данными на вход нейросети. Для осуществления данных действий была использована библиотека ОpenCV. Для нахождения черт лица использовались те же методы что и при подготовке обучающей выборки фотографий, с разными каскадами Хаара.

1. **Результаты экспериментов**

Для тестирования основного метода произведены 3 эксперимента: с использование 5, 10 характеристик, а также без их использования, подавая на вход сети яркость пикселей изображения. Также при тестировании программы обнаружилась нехватка оперативной памяти компьютера при обучающей выборке в 204 изображения и 10 характеристиках, поэтому все тесты обучения проведены с использованием 25 мужских и 20 женских изображений.

В ходе тестов выяснилось, что при количестве эпох более 30 ошибка имеет значение около 0,1 и далее уменьшается на крайне низкое значение. На основе этого принято решение использовать 50 эпох для экономии времени.

В ходе тестирования были получены следующие результаты:

1) 20 f, 25 m, 5 parameters

Working . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

Value of error = 0.116014996231

f\_25.ppm [ 0.6700475]

f\_21.ppm [ 0.36917946]

m\_53.ppm [ 0.6700475]

f\_23.ppm [ 0.36917946]

m\_52.ppm [ 0.36917946]

m\_51.ppm [ 0.71288041]

f\_22.ppm [ 0.63364105]

f\_24.ppm [ 0.37469869]

f\_1.ppm [ 0.41201237]

2) 20 f, 25 m, 10 par

Working . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

Value of error = 0.117524012577

f\_25.ppm [ 0.51748549]

f\_21.ppm [ 0.74359549]

m\_53.ppm [ 0.36951038]

f\_23.ppm [ 0.51748549]

m\_52.ppm [ 0.51748549]

m\_51.ppm [ 0.55372472]

f\_22.ppm [ 0.77983472]

f\_24.ppm [ 0.74359549]

f\_1.ppm [ 0.51748549]

3) 20 f, 25 m, 0 par (берется яркость пикселей, а не их разность)

Working . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . .

Value of error = 0.107943108553

f\_25.ppm [ 0.1921415]

f\_21.ppm [ 0.1921415]

m\_53.ppm [ 0.64154315]

f\_23.ppm [ 0.1921415]

m\_52.ppm [ 0.1921415]

m\_51.ppm [ 0.1921415]

f\_22.ppm [ 0.64154315]

f\_24.ppm [ 0.1921415]

f\_1.ppm [ 0.64154315]

По результатам тестирования метод был признан неэффективным.

При работе с изображениями вторым методом, на больших фотографиях удалось выделать части лица (рот, нос, глаза) как показано на рисунке 1.

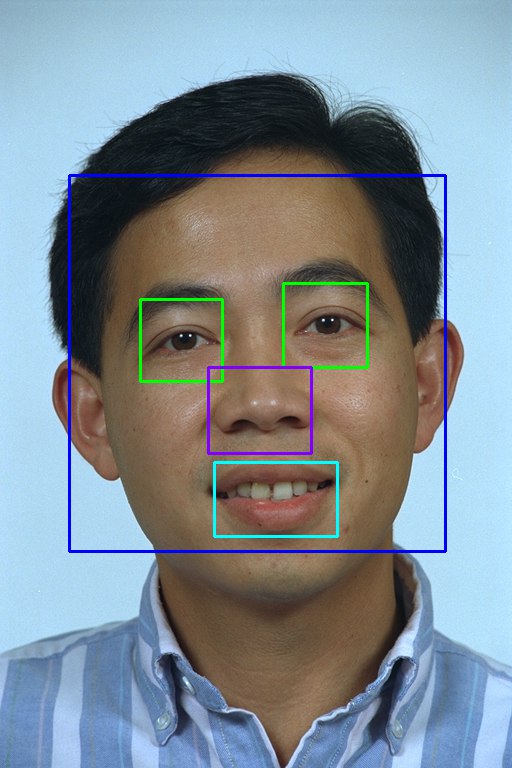


Рисунок 1 – результат распознавания на фото отдельных частей лица

Однако на фото размером 20х20 программа не смогла обнаружить части лица, из-за чего данный способ был признан неудовлетворяющим требованиям исходной задачи.

С работой также можно ознакомиться в репозитории Lab1. <https://github.com/KlimovSN-ASU2-16/Labs/>lab1

**Заключение**

Таким образом, в ходе выполнения лабораторной работы получены следующие результаты:

* создана такая модель нейронной сети как многослойный персептрон;
* подготовлена обучающая выборка, содержащая 296 изображений размером 20x20 пикселей;
* осуществлено обучение многослойного персептрона;
* проведено тестирование нейросети;
* произведен анализ полученных результатов
* по результатам тестирования описанный метод признан неработоспообным
* предложен другой метод решения проблемы, также не дал результатов.

В результате проделанной работы был сделан вывод, что при помощи OpenCV и Pybrain распознавание пола на фото размерам 20х20 пикселей не осуществимо.

**Список используемых источников**

1. Е.В. Долгова, Д.С. Курушин. Компьютерные нейросетевые технологии: учебное пособие. — Пермь: Изд-во Перм. гос. техн. ун-та, 2008 .— 87с.
2. Хабрахабр – Распознавание пола в изображениях и видео.

URL: <https://habrahabr.ru/post/172463/> (Дата обращения 5.10.16.)

1. Открытая база данных FERET. URL: https://face-rec.org/databases

(Дата обращения 5.10.16.)

1. Face Detection using Haar Cascades.

URL: <http://docs.opencv.org/trunk/d7/d8b/tutorial_py_face_detection.html>

(Дата обращения 5.10.16.)

1. PyBrain работаем с нейронными сетями в Python <https://habrahabr.ru/post/148407/> (Дата обращения 5.10.16.)