

ПЕНЗЕНСКАЯ ОБЛАСТЬ, г. Кузнецк  
Муниципальное бюджетное общеобразовательное учреждение  
лицей № 21 города Кузнецка

**Тема: Распознавание образов в системах  
видеонаблюдения**

**Автор: Кузнецов Арсений Алексеевич,  
Титеев Рамиль Маратович**

**Руководитель: Кузнецов Алексей  
Александрович,  
учитель информатики**

# Оглавление

1. Введение.....	3
2. Основная часть.....	5
2.1.Объект исследования – нейронная сеть.....	5
2.1.1Многослойный перцептрон.....	5
2.1.2.Рекуррентный перцептрон.....	7
2.1.3.Ассоциативная память.....	8
2.1.4.Спайковые сети.....	9
2.2.Предмет исследования.....	11
2.2.1.Метод гибкого сравнения на графах (Elastic graph matching).....	11
2.2.2.Основные проблемы, связанные с разработкой систем распознавания лиц.....	14
3. Исследовательская часть.....	15
3.1.Составление программного продукта на базе языка python при использовании библиотек (os, numpy, time, OpenCv).....	15
4. Заключение.....	20
5. Библиографический список.....	21

## **1. Введение**

Сегодня самым важным становится безопасность. В школах устанавливают турникеты с пропуском по карточки, но из-за ненадёжности сохранения карточек, которые могут не только забыть дома, но и потерять или сломать. Такая проблема может возникнуть у взрослых, не говоря о детях. Потерянной карточкой могут воспользоваться злоумышленники.

Решением этой проблемы является установка видеокамеры с внедрением нейронной сети по распознаванию лиц в пропускные пункты. Такая система увеличит надёжность охраны, позволяя исключить человеческий фактор.

На текущий момент существует множество различных методов и алгоритмов, которые позволяют обработать изображение и получить информацию о присутствующих на ней лицах. Для выбора оптимального решения в условиях решаемой задачи распознавание лиц.

Распознавание лиц – задача, которую человек выполняет без особых усилий несколько раз в день. Это одна из ключевых биометрических технологий. Распознавание лиц имеет ряд преимуществ перед другими биометрическими технологиями: это естественность, доступность и простота в использовании. Исходя из этого, вопрос улучшения и модернизации данного типа систем распознавания является актуальным.

**Цель работы:** повысить уровень безопасности путем интеграции идентификации личности посредством портретной экспертизы в системы видеонаблюдения.

**Задача:**

- Структурировать предметную область и выявить основные направления развития при выборе алгоритмов определения и захвата лица.
- Познакомиться с программными продуктами, обеспечивающих решение задач по распознаванию образов.
- Проектирование системы и средств получения и хранения видеоматериалов.

**Гипотеза:**

Возможно ли улучшить с минимальными затратами охранную систему.

**Методы исследования:**

- теоретические (поиск, изучение и анализ справочной литературы и интернет - ресурсов, обобщение полученной информации);
- практические (Реализация системы, апробация и анализ полученных результатов).

## **2. Основная часть**

### **2.1. Объект исследования – нейронная сеть.**

В настоящее время существует около десятка разновидностей нейронных сетей (НС). Одним из самых широко используемых вариантов является сеть, построенная на многослойном перцептроне, которая позволяет классифицировать поданное на вход изображение/сигнал в соответствии с предварительной настройкой/обучением сети.

Обучаются нейронные сети на наборе обучающих примеров. Суть обучения сводится к настройке весов межнейронных связей в процессе решения оптимизационной задачи методом градиентного спуска. В процессе обучения НС происходит автоматическое извлечение ключевых признаков, определение их важности и построение взаимосвязей между ними. Предполагается, что обученная НС сможет применить опыт, полученный в процессе обучения, на неизвестные образы за счет обобщающих способностей.

Наилучшие результаты в области распознавания лиц (по результатам анализа публикаций) показала Convolutional Neural Network или сверточная нейронная сеть (далее – СНС) [29-31], которая является логическим развитием идей таких архитектур НС как когнитрона и неокогнитрона. Успех обусловлен возможностью учета двумерной топологии изображения, в отличие от многослойного перцептрона.

Отличительными особенностями СНС являются локальные рецепторные поля (обеспечивают локальную двумерную связность нейронов), общие веса (обеспечивают детектирование некоторых черт в любом месте изображения) и иерархическая организация с пространственным сэмпингом (spatial subsampling). Благодаря этим нововведениям СНС обеспечивает частичную устойчивость к изменениям масштаба, смещениям, поворотам, смене ракурса и прочим искажениям.

#### **2.1.1. Многослойный перцептрон**

Самая известная и очень старая архитектура, в которой идут подряд несколько слоев нейронов — входной, один или несколько скрытых слоев, и выходной слой. Почти всегда обучается методом обратного распространения ошибки — что

автоматически означает, что мы должны предоставить для обучения набор пар «входной вектор — правильный выход». Тогда входной вектор отправится на вход сети, последовательно будут рассчитаны состояния всех промежуточных нейронов, и на выходе образуется выходной вектор, который мы и сравним с правильным. Расхождение даст нам ошибку, которую можно распространить обратно по связям сети, вычислить вклад в итоговую ошибку каждого нейрона, и скорректировать его веса, чтобы ее исправить. Повторив эту процедуру много тысяч раз, возможно выйдет обучить сеть.

Сеть такого типа обычно очень хорошо справляется с задачами, где:

1. ответ действительно зависит только от того, что мы даем на вход сети, и никак не зависит от истории входов (т.е. это не динамический процесс, или, по крайней мере, мы дали на вход исчерпывающую информацию об этом процессе в форме, пригодной для обработки сетью).
2. ответ не зависит/слабо зависит от высоких степеней и/или произведений параметров — функции этого типа сеть строить почти не умеет.
3. в наличии есть достаточно много примеров (желательно иметь не менее сотни примеров на каждую связь сети), или у вас есть большой опыт борьбы с эффектом специализации. Это связано с тем, что имея много коэффициентов, сеть может банально запомнить много конкретных примеров, и выдавать на них отличный результат — но ее прогнозы не будут иметь ничего общего с реальностью в случае, если дать на вход примеры не из обучающей выборки.

Сильные стороны — изучена со всех сторон, хорошо работает на своих задачах, если на некоторой задаче не работает (действительно не работает, а не по криворукости, как это бывает чаще всего) — то это повод утверждать, что задача сложнее, чем казалось. Слабые стороны — неумение работать с динамическими процессами, необходимость большой обучающей выборки. Перспективы — никаких существенных. Большинство серьезных задач, которые все еще требуют решения, не входят в класс задач, решаемых многослойным перцептроном с методом обратного распространения ошибки.

### 2.1.2. Рекуррентный перцептрон

На первый взгляд похож на обычный перцептрон, единственное существенное отличие состоит в том, что его выходы попадают ему же на входы, и участвуют в обработке уже следующего входного вектора. То есть, в случае рекуррентного перцептрона имеет место не набор отдельных, ничем не связанных образов, а некоторый процесс, и значение имеют не только сами входы, но и то, в какой последовательности они поступают. Из-за этого возникают отличия в методе обучения — используется то же самое обратное распространение ошибки, но для того, чтобы ошибка попала по рекуррентной связи в прошлое, используются разные ухищрения (если подойти к задаче «в лоб» — то возникнет проблема ухода ошибки на бесконечное число циклов назад). В остальном же ситуация похожа на обычный перцептрон — для обучения нужно иметь достаточно длинную последовательность пар вход-выход, которую нужно много раз прогнать через сеть, чтобы ее обучить (или же иметь под рукой мат. модель искомого процесса, которую можно гонять во всевозможных условиях, и в реалтайме давать результаты сети для обучения).

Сеть такого типа обычно хорошо решает задачи управления динамическими процессами (начиная от классической задачи стабилизации перевернутого маятника, и до любых систем, которыми вообще хоть как-то получается управлять), предсказания динамических процессов, кроме курса валют :), и вообще всего, где помимо явно наблюдаемого входа у системы есть некоторое внутреннее состояние, которое не совсем понятно как использовать.

Сильные стороны: сеть очень хороша для работы с динамическими процессами. Слабые стороны: если все же не работает, понять в чем проблема — очень затруднительно, в процессе обучения может вылететь в самовозбуждение (когда сигнал, полученный с выхода, забивает все, что приходит по входам), если решение все же получено — сложно понять, можно ли добиться лучших результатов, и каким путем. Другими словами, плохо изучена. Перспективы: этот подход явно не исчерпал себя в вопросах управления — собственно, на данный момент рекуррентные перцептроны используются довольно редко, хотя их потенциал высок. Интересные результаты может дать подход с непрерывно

адаптирующей к объекту управления сетью, хотя для этого еще необходимо решить проблему неустойчивости обучения.

### **2.1.3. Ассоциативная память**

Это широкий класс сетей, которые в той или иной степени напоминают архитектуру Хопфилда, которая состоит из одного слоя нейронов, выходы которого поступают на его входы в следующий момент времени. Этот слой служит и входом сети (в начальный момент выходы нейронов принимаются равными входному вектору), и ее выходом — значения на нейронах, образовавшиеся в конце работы, считаются ответом сети. Эта сеть меняет свои состояния с течением времени до тех пор, пока состояние не перестанет меняться. Свойства весовой матрицы выбраны таким образом, чтобы устойчивое состояние всегда гарантированно достигалось (и обычно это происходит за несколько шагов). Такая сеть помнит некоторое количество векторов, и при подаче на вход любого вектора, может определить, на какой из запомненных он более всего похож — отсюда и название. Двухслойная модификация этой сети (гетеро ассоциативная память) может запоминать вектора не по одному, а по парам разной размерности. Сети такого типа хорошо справляются с задачами, где нужно определить похожесть вектора на один из стандартных запомненных. Собственно, это единственный класс задач, где они хороши. Также конкретно сеть Хопфилда может использоваться для решения задач оптимизации (например, задачи коммивояжера), однако ее эффективность в этой области под вопросом.

**Сильные стороны** — очень быстрое обучение (т.к. вместо градиентного спуска решается система уравнений), возможность удалить образ из памяти или добавить в память, не затронув остальные, некоторые свойства такой памяти напоминают свойства мозга, и их изучение интересно с такой позиции. Слабые стороны — очень узкий класс решаемых задач, неумение обобщать примеры, максимальный объем памяти жестко связан с размерностью запоминаемого вектора (ввиду особенностей построения).

**Перспективы:**— разработана ядерная (от слова kernel) ассоциативная память, которая способна к обобщению образов, и имеет неограниченный объем памяти (сеть растет по мере заполнения) — разработана динамическая ассоциативная

память, которая запоминает не отдельные образы, а определенные последовательности образов, и поэтому может применяться для распознавания элементов динамических процессов — динамическая ассоциативная память демонстрирует способность к генерации отклика, содержащего разные элементы запомненных последовательностей при подаче входного сигнала, соответствующего одновременно разным последовательностям, что, возможно, является некоторой грубой моделью творчества человека — гибрид ядерной и динамической ассоциативной памяти может дать новое качество в распознавании последовательностей — например, в распознавании речи.

#### **2.1.4. Спайковые сети**

Это особый класс сетей, в которых сигнал представлен не вещественным числом, как во всех ранее рассмотренных сетях, а набором импульсов (спайков) одинаковой амплитуды и длительности, и информация содержится не в амплитуде, а в интервалах между импульсами, в их паттерне. Спайковые нейроны на выходе генерируют спайки, либо одиночные (если суммарный вход не очень большой), или пакеты (если суммарный вход достаточно велик). Этот тип сетей почти полностью копирует процессы, проходящие в мозгу человека, единственное серьезное отличие — для обучения не придумано ничего лучше правила Хебба (которое звучит примерно так: если второй нейрон сработал сразу после первого, то связь от первого ко второму усиливается, а если сразу перед первым — то ослабевает), для которого был придуман ряд небольших усовершенствований, но, к сожалению, повторить свойства мозга в области обучения пока толком не получилось. Сети такого типа умеют приспособивать для решения различных задач, решаемых другими сетями, но редко результаты оказываются существенно лучше. В большинстве случаев удастся только повторить уже достигнутое.

**Сильные стороны:** очень интересны для изучения как модели биологических сетей. **Слабые стороны:** почти любое практическое применение выглядит необоснованно, сети других типов справляются не хуже.

#### **Перспективы:**

- моделирование масштабных спайковых сетей в ближайшие годы вероятно даст много ценной информации о психических расстройствах,

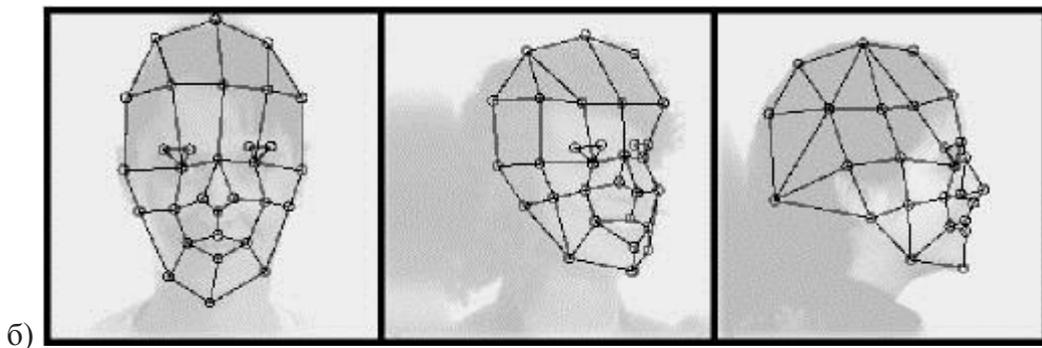
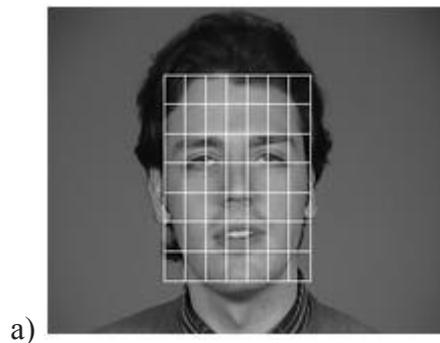
- позволит классифицировать нормальный и ненормальный режимы работы различных отделов мозга.

В более отдаленной перспективе, после создания подходящего алгоритма обучения, такие сети по функциональности сравняются или даже превзойдут другие типы нейросетей, а еще позднее на их основе можно будет собирать структуры, пригодные для прямого подключения к биологическому мозгу, для расширения возможностей интеллекта.

## 2.2. Предмет исследования - нейронная сеть по определению лица.

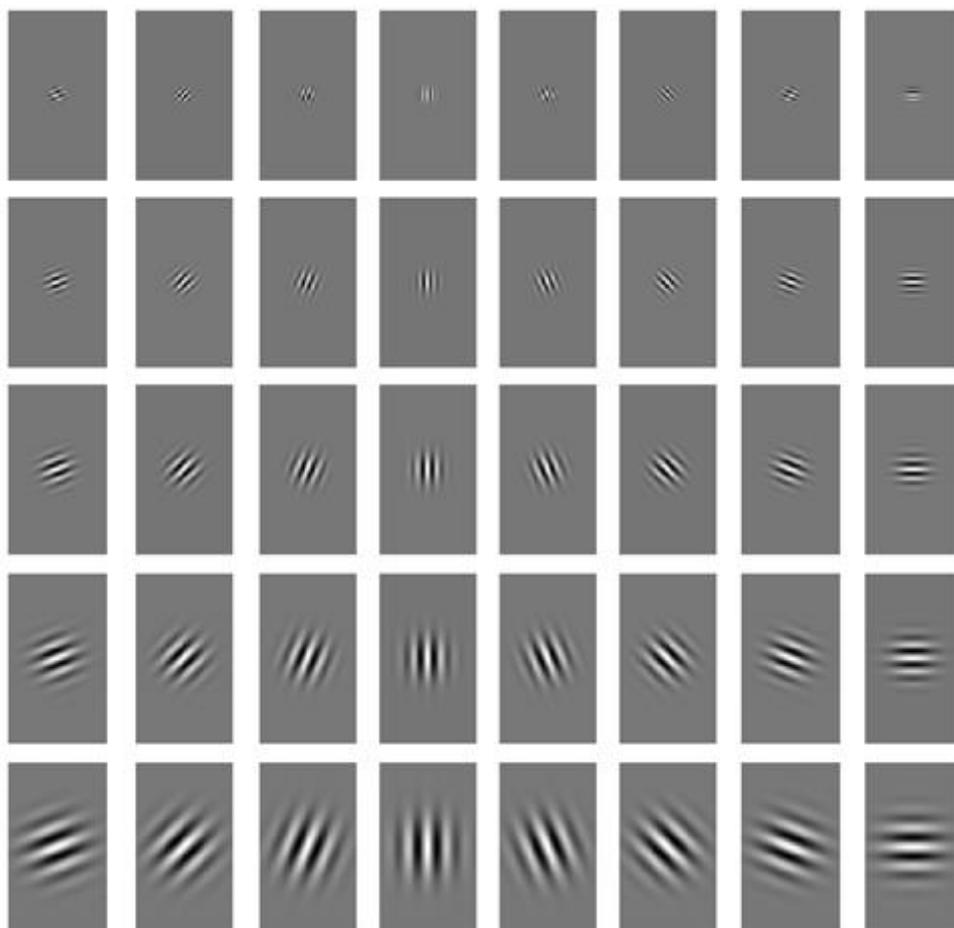
### 2.2.1. Метод гибкого сравнения на графах (Elastic graph matching) .

Суть метода сводится к эластичному сопоставлению графов, описывающих изображения лиц. Лица представлены в виде графов со взвешенными вершинами и ребрами. На этапе распознавания один из графов – эталонный – остается неизменным, в то время как другой деформируется с целью наилучшей подгонки к первому. В подобных системах распознавания графы могут представлять собой как прямоугольную решетку, так и структуру, образованную характерными (антропометрическими) точками лица.

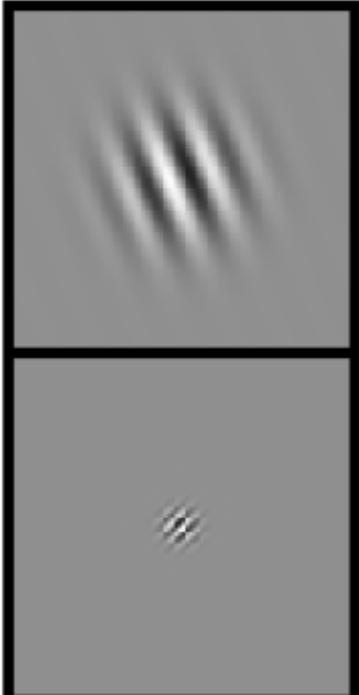
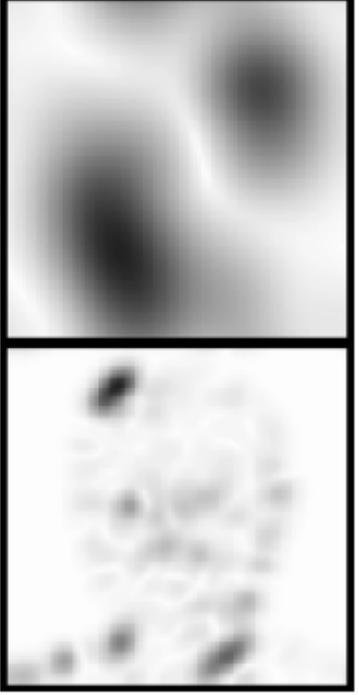


*Пример структуры графа для распознавания лиц: а) регулярная решетка б) граф на основе антропометрических точек лица.*

В вершинах графа вычисляются значения признаков, чаще всего используют комплексные значения фильтров Габора или их упорядоченных наборов – Габоровских вейвлет (строи Габора), которые вычисляются в некоторой локальной области вершины графа локально путем свертки значений яркости пикселей с фильтрами Габора.



*Набор (банк, jet) фильтров Габора*

Входное изображение	Пример двух фильтров Габора	Результат свертки входного изображения лица и фильтров Габора
		

*Пример свертки изображения лица с двумя фильтрами Габора*

Ребра графа взвешиваются расстояниями между смежными вершинами. Различие (расстояние, дискриминационная характеристика) между двумя графами вычисляется при помощи некоторой ценовой функции деформации, учитывающей как различие между значениями признаков, вычисленными в вершинах, так и степень деформации ребер графа.

Деформация графа происходит путем смещения каждой из его вершин на некоторое расстояние в определённых направлениях относительно ее исходного местоположения и выбора такой ее позиции, при которой разница между значениями признаков (откликов фильтров Габора) в вершине деформируемого графа и соответствующей ей вершине эталонного графа будет минимальной. Данная операция выполняется поочередно для всех вершин графа до тех пор, пока не будет достигнуто наименьшее суммарное различие между признаками деформируемого и эталонного графов. Значение ценовой функции деформации при таком положении деформируемого графа и будет являться мерой различия между входным изображением лица и эталонным графом. Данная «релаксационная» процедура деформации должна выполняться для всех эталонных лиц, заложенных в базу данных системы.



*Пример деформации графа в виде регулярной решетки*

Результат распознавания системы – эталон с наилучшим значением ценовой функции деформации 95-97% эффективности распознавания при наличии различных эмоциональных выражений и изменении ракурса лица до 15 градусов. Однако минусом систем эластичного сравнения на графах является высокая вычислительная стоимость данного подхода.

**Недостатки:** высокая вычислительная сложность процедуры распознавания. Низкая технологичность при запоминании новых эталонов. Линейная зависимость времени работы от размера базы данных лиц.

## 2.2.2. Основные проблемы, связанные с разработкой систем распознавания лиц

### Проблема освещенности



Проблема положения головы (лицо – это, все же, 3D объект).



В тестах алгоритма, основанных на гибком сравнении на графах и всевозможных модификаций метода главных компонент (РСА) эффективность алгоритмов была примерно одинаковой. В этой связи трудно или даже невозможно провести четкие различия между ними. Для фронтальных изображений, сделанных в один и тот же день, приемлемая точность распознавания, как правило, составляет 95%. Для изображений, сделанных разными аппаратами и при разном освещении, точность, как правило, падает до 80%. При этом стоит заметить, что даже 50 процентов — это более чем приемлемая точность работы системы подобного рода.

## 2.1.1. Исследовательская часть

### 3.1. Составление программного продукта на базе языка python при использовании библиотек os, numpy, time, OpenCv.

Создание системы директорий (папок), в которых хранятся фотографии лиц, на которых обучается нейронная сеть.

```
def prepare_training_data(data_folder_path):
# -----ШАГ-1-----
# получим список директорий (папок), содержащихся в папке data_folder_path
    dirs = os.listdir(data_folder_path)

# список лиц
    faces = []
# список лейблов (меток)
    labels = []

# считаем изображения в каждой директории по очереди
    for dir_name in dirs:
# директории каждого человека начинаются с буквы 's', поэтому будем рассматривать только их и игнорировать
# остальные папки, если они есть
        if not dir_name.startswith("s"):
            continue

# -----ШАГ-2-----
# получим лейбл из имени папки: просто удалим букву 's' и переведем строку в целое число (integer)
        label = int(dir_name.replace("s", ""))

# запишем в переменную путь до директории, содержащей изображения
# например, subject_dir_path = "014_train_data/s1"
        subject_dir_path = data_folder_path + "/" + dir_name

# получаем список из имен изображений в папке
        subject_images_names = os.listdir(subject_dir_path)

# -----ШАГ-3-----
# считаем каждое изображение
# распознаем лицо и добавим его в список faces
        for image_name in subject_images_names:
# игнорируем системные файлы типа .DS_Store
            if image_name.startswith("."):
                continue
```

```

# запишем в переменную путь до изображения
# например, image_path = 014_train_data/s1/1.jpg
    image_path = subject_dir_path + "/" + image_name

# читаем изображение
    image = cv2.imread(image_path)

# найдем лицо на фото
    face, rect = detect_face(image)

# -----ШАГ-4-----
# в рамках данного tutorials мы будем игнорировать лица, которые не были найдены
    if face is not None:
# добавим лицо в список
        faces.append(face)
# добавим лейбл в список
        labels.append(label)

    return faces, labels

# эта функция идентифицирует человека со входящей фотографии и рисует прямоугольник с именем
def predict(input_img):
    img = input_img.copy()
# определим лицо на фото
    face, rect = detect_face(img)

# если не нашли лицо в кадре вернем исходный кадр
    if face is None:
        return input_img

# идентифицируем лицо
    label, confidence = face_recognizer.predict(face)
# получим имя по лейблу
    label_text = subjects[label]

# рисуем прямоугольник
    draw_rectangle(img, rect)
# пишем имя
    threshold = 50
    if confidence > threshold:
        draw_text(img, label_text, rect[0], rect[1] - 5)
        conf_str = str(round(confidence, 3))
        draw_confidence(img, conf_str, rect[0], rect[1] - 30)

```

```

else:
    draw_text(img, 'Unknown', rect[0], rect[1] - 5)
    return img
# threshold = 300
# if confidence < threshold:
    #draw_text(img, label_text, rect[0], rect[1] - 5)
    #conf_str = str(round(confidence, 3))
    #draw_confidence(img, conf_str, rect[0], rect[1] - 30)
# else:
# draw_text(img, 'Unknown', rect[0], rect[1] - 5)
    return img
print("Preparing data...")
[faces, labels] = prepare_training_data('C:\\Users\\arseni\\PycharmProjects\\neiroseti\\Neuronka\\Training')
print("Data prepared")

# выведем в длины списков лиц и лейблов
print("Total faces: ", len(faces))
print("Total labels: ", len(labels))

# создаем LBPH face recognizer

face_recognizer = cv2.face.LBPHFaceRecognizer_create()

# тренируем модель
face_recognizer.train(faces, np.array(labels))

cap = cv2.VideoCapture(0)
time.sleep(2.0)

# проверим, успешно ли произошло открытие камеры
if cap.isOpened() == False:
    print("Ошибка при открытии камеры!")

cv2.namedWindow('Window')
cv2.moveWindow('Window', 75, 75)

# будем выполнять цикл, пока не нажмем на кнопку для выхода из программы
while cap.isOpened():
# считаем кадр (будем также называть его frame)
    ret, frame = cap.read()
    if ret:
        predicted_img = predict(frame)
# отобразим кадр

```

```

cv2.imshow('Window', predicted_img)

# если была нажата клавиша "q", то завершим вывод видео
button = cv2.waitKey(1) & 0xFF
if button == ord("q"):
    break

button2 = cv2.waitKey(1) & 0xFF
if button2 == ord("p"):
    new_man()

# когда мы закончили работать с видео, освобождаем переменную
cap.release()

# закрываем все окна
cv2.destroyAllWindows

```

Теперь, после того, как мы научили нашу нейросеть, мы хотим увидеть, как хорошо она работает на тестах. Дан входной тест, нам нужно получить вывод нейронной сети — это делается через запуск процесса прямого распространения через сеть, используя наши полученные значения веса и смещения. Как было сказано ранее, мы выбираем результат выходного слоя через выбор узла с максимальным выводом. Теперь, наконец, мы можем оценить точность результата

## 4. Заключение

С движением прогресса данная технология весьма быстро развивается. Появление более мощных процессоров позволяет обрабатывать большие запасы данных с большой скоростью, а так же более качественный анализ информации на самой камере.

Таким образом, изучение нейронных сетей позволило нам создать программу по определению лиц, предлагаемое решение, которое комбинирует использование метода гибкого сравнения на графах (Elastic graph matching) для определения лиц в кадре, так как этот метод более простой для реализации.

Данное решение основано на том, что для большинства современных систем автоматического распознавания лиц основной задачей является задача сравнения заданного изображения с набором изображений лиц из базы данных. Характеристики систем автоматического распознавания лиц в этом случае оцениваются путем определения вероятностей ошибочного отказа в распознавании и ошибочного распознавания. И предложенный вариант решает эти основные задачи, так как он позволяет хранить в удобном формате информацию (набор признаков), а также имеет возможность быстро и точно определять наличие лиц в кадре при помощи использования каскада признаков.

Направлением дальнейших исследований является задача выбора наиболее подходящего кадра из видеопотока для последующей его обработки.

Мы сделали вывод, что внедрение таких технологий может увеличить безопасность в организациях и усилить их охранную систему

Мы получили результат 86% точности. Звучит довольно неплохо? На самом деле, нет, это довольно низкая точностью. В наше время точность алгоритмов глубинного обучения достигает 99.7%, мы немного отстали.

## 5. Библиографический список

- Соколова Т. Е. «Кодирование и хранение информации» Издательство: Дом Федорова, 2008 г.
- Книги: 1. Книга «Видеоаналитика Мифы и реальность» от авторов Торстен Анштедт, Иво Келлер, Харальд Лутц – основной источник информации
- Хэмминг Р.В. «Теория кодирования и теория информации», 1983 г.  
3.[http://wiki.obr55.ru/index.php?title=Коллективная\\_статья\\_Шифровальщики/Учебный\\_проект\\_Мир\\_кодов](http://wiki.obr55.ru/index.php?title=Коллективная_статья_Шифровальщики/Учебный_проект_Мир_кодов).
- <http://videouroki.net/filecom.php?fileid=98659731>.
- <http://sch10ptz.ru/projects/002/inf/1.7.htm>.
- [http://infourok.ru/tayny\\_koda\\_-\\_uchebnyy\\_proekt\\_po\\_informatike\\_po\\_fgosam-112992.htm](http://infourok.ru/tayny_koda_-_uchebnyy_proekt_po_informatike_po_fgosam-112992.htm)
- <http://www.ru.wikipedia.org>.