

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
федеральное государственное бюджетное образовательное учреждение высшего образования  
«Тольяттинский государственный университет»

Институт математики, физики и информационных технологий  
(наименование института полностью)

Кафедра «Прикладная математика и информатика»  
(наименование)

01.03.02 Прикладная математика и информатика  
(код и наименование направления подготовки / специальности)

Компьютерные технологии и математическое моделирование  
(направленность (профиль) / специализация)

## ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему «Сравнение алгоритмов машинного обучения для задачи распознавания лиц»

Обучающийся

А.В. Мурзинов

(Инициалы Фамилия)

(личная подпись)

Руководитель

к.ф.-м.н. О.В. Лелонд

(ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Консультант

к.п.н., доцент Т.С. Якушева

(ученая степень (при наличии), ученое звание (при наличии), Инициалы Фамилия)

Тольятти 2023

## Аннотация

Тема выпускной квалификационной работы – «Сравнение алгоритмов машинного обучения для задачи распознавания лиц».

Помимо конфигураций и вычислительной мощности используемых для запуска программы одним из критериев выбора конкретного алгоритма машинного обучения является эффективность в каждом отдельном взятом случае.

Исследование и особенности практического применения машинного обучения для систем распознавания лиц представляют актуальность и научно–практический интерес.

Объектом исследования бакалаврской работы являются алгоритмы машинного обучения для систем распознавания лиц.

Предметом исследования бакалаврской работы будет сравнение и реализация алгоритмов машинного обучения.

Цель бакалаврской работы являются алгоритмы машинного обучения.

Методы исследования – методы и алгоритмы машинного обучения, технологии реализации алгоритмов на языках высокого уровня.

Практическая значимость бакалаврской работы заключается в разработке и тестировании программы, реализующей эффективный алгоритм машинного обучения.

Результаты бакалаврской работы представляют научно–практический интерес и могут быть рекомендованы для анализа и программной реализации методов и алгоритмов машинного обучения для систем распознавания лиц.

Бакалаврская работа состоит из 43 страниц текста, 17 скриншотов, одной диаграммы, одной гистограммы и 26 источников.

## **Abstract**

The theme of the final qualification work is “Comparison of machine learning algorithms for the problem of face recognition”.

In addition to the configurations and computing power used to run the program, one of the criteria for choosing a particular machine learning algorithm is efficiency in each individual case.

The study and features of the practical application of machine learning for face recognition systems are of relevance and scientific and practical interest.

The object of study of the bachelor's work is machine learning algorithms for face recognition systems.

The subject of research of the bachelor's work is machine learning algorithms.

The purpose of the bachelor's work is the comparison and implementation of machine learning algorithms.

Research methods – methods and algorithms of machine learning, technologies for implementing algorithms in high-level languages.

The practical significance of the bachelor's work lies in the development and testing of a program that implements an effective machine learning algorithm.

The results of the bachelor's work are of scientific and practical interest and can be recommended for the analysis and software implementation of machine learning methods and algorithms for face recognition systems.

Bachelor's thesis consists of 43 pages of text, 17 screenshots, one diagram, one histogram and 26 sources.

## Оглавление

Введение.....	5
Глава 1 Задачи систем распознавания лиц .....	6
1.1 Задачи и примеры использования .....	6
1.2 Недостатки.....	8
Глава 2 Системы распознавания лиц .....	10
2.1 Алгоритмы классификации.....	10
2.2 Нейронные сети.....	12
2.3 Алгоритмы глубокого обучения.....	14
2.4 Алгоритмы сопоставления шаблонов.....	15
2.5 Алгоритмы на основе SVM.....	17
2.6 Метод главных компонент .....	18
2.7 Сравнение алгоритмов .....	20
Заключение .....	39
Список используемой литературы и используемых источников.....	41

## Введение

Система распознавания лиц — это технология, способная сопоставлять человеческое лицо из цифрового изображения или видеокadra с базой данных лиц. Такая система обычно используется для аутентификации пользователей с помощью служб проверки личности и работает путем точного определения и измерения черт лица на данном изображении. Системы распознавания лиц являются важным инструментом в современной технологической среде. Они нашли широкое применение в различных областях, таких как безопасность, аутентификация, видеонаблюдение и социальные сети. Однако, эффективность таких систем во многом зависит от алгоритмов машинного обучения, применяемых для распознавания лиц.

Разработка подобных систем началась в 1960-х годах, начиная с формы компьютерного приложения. С момента своего создания системы распознавания лиц в последнее время получили более широкое применение на смартфонах и в других технологиях, таких как робототехника. Поскольку компьютеризированное распознавание лиц включает измерение физиологических характеристик человека, системы распознавания лиц относятся к категории биометрических. Хотя точность систем распознавания лиц как биометрической технологии ниже, чем распознавание радужной оболочки глаза и распознавание отпечатков пальцев, она широко применяется благодаря бесконтактному процессу. Системы распознавания лиц были развёрнуты в расширенном взаимодействии человека с компьютером, видеонаблюдении и автоматической индексации изображений.

Основываясь на анализе и сравнении различных алгоритмов, будет произведена оценка их применимости и эффективности для систем распознавания лиц. Это позволит исследователям и разработчикам выбрать наиболее подходящий алгоритм в зависимости от конкретных требований и контекста применения.

## Глава 1 Задачи систем распознавания лиц

### 1.1 Задачи и примеры использования

Системы распознавания лиц используются в различных сферах, таких как безопасность, маркетинг, медицина и т.д. В зависимости от конкретной области применения, задачи перед системами распознавания лиц могут варьироваться. Ниже приведены некоторые из основных задач, которые могут стоять перед системами распознавания лиц:

- идентификация личности: главная задача систем распознавания лиц – это идентификация личности по фотографии или видеозаписи,
- аутентификация личности: системы распознавания лиц также могут использоваться для аутентификации личности, то есть для подтверждения, что конкретный человек является тем, за кого он себя выдает,
- определение эмоций: системы распознавания лиц могут также определять эмоции, которые испытывает человек по его выражению лица,
- определение возраста и пола: системы распознавания лиц могут определять пол и возраст человека по его лицу,
- обнаружение деформаций: системы распознавания лиц могут обнаруживать деформации лица, такие как брекеты, очки и т.д., что может помочь в повышении точности распознавания личности,
- определение настроения и поведения: системы распознавания лиц могут также определять настроение и поведение человека по его выражению лица и жестам.
- обнаружение преступников: системы распознавания лиц могут использоваться в правоохранительных органах для обнаружения преступников и поиска пропавших людей,
- мониторинг: системы распознавания лиц могут использоваться для мониторинга присутствия людей на определенной территории, например, в магазинах или на рабочих местах.

Распознавание лиц – это технология, которая может принести множество преимуществ и улучшений в жизнь людей. Некоторые из них включают в себя:

– улучшенная безопасность: Системы распознавания лиц могут помочь обеспечить безопасность в различных областях, таких как охрана общественных мест, государственных учреждений, транспортных систем и т.д. Они могут помочь в идентификации подозрительных личностей, запрете доступа несанкционированных лиц и пресечении других нарушений безопасности,

– улучшенный процесс идентификации: Распознавание лиц может помочь улучшить процесс идентификации в различных областях, таких как медицинское обслуживание, государственная регистрация, аэропорты и другие сферы, где необходимо подтверждение личности,

– улучшенный процесс обработки и анализа данных: Системы распознавания лиц могут помочь ускорить и улучшить процесс обработки и анализа данных. Например, они могут использоваться для автоматической классификации фотографий в социальных сетях или для идентификации мошеннических операций на банковских счетах,

– удобство использования: Распознавание лиц может помочь улучшить удобство использования в различных областях, таких как автоматизация систем доступа, использование в биометрических системах аутентификации или улучшение функциональности домашней техники,

– улучшенное взаимодействие: Системы распознавания лиц могут помочь улучшить взаимодействие людей в различных сферах, таких как социальная сфера и развлечения. Например, использование технологии распознавания лиц может позволить людям быстро находить своих друзей на фотографиях в социальных сетях или автоматически распознавать их на мероприятиях [9].

Таким образом, распознавание лиц может принести множество преимуществ и улучшений в различные области жизни людей.

## 1.2 Недостатки

Кроме основных задач, перед системами распознавания лиц стоят и другие вызовы и задачи, которые необходимо учитывать при их разработке и использовании:

- проблема приватности: использование систем распознавания лиц может привести к нарушению приватности и прав на защиту персональных данных. Необходимо обеспечить соответствие систем законодательству и этическим принципам,

- низкая точность распознавания: системы распознавания лиц могут быть не совсем точными, особенно в условиях изменчивости освещения, угла обзора и других факторов. Это может привести к ложным срабатываниям и ошибкам идентификации,

- необходимость большой базы данных: для эффективной работы системы распознавания лиц необходимо иметь большую базу данных с изображениями лиц. Создание такой базы данных может быть сложным и дорогостоящим процессом,

- риски злоупотребления: системы распознавания лиц могут быть использованы для незаконных целей, таких как слежка за людьми или нарушение их прав. Необходимо обеспечить защиту от злоупотреблений и мисс использования,

- необходимость обучения: системы распознавания лиц требуют обучения на большом количестве изображений лиц. Это может занять много времени и ресурсов, а также требует наличия экспертов по распознаванию лиц.

Тем не менее, системы распознавания лиц представляют собой важный инструмент в различных областях и их использование может привести к улучшению эффективности и безопасности в соответствующих сферах. Важно обеспечить правильный баланс между преимуществами и возможными рисками при разработке и использовании таких систем.



## Выводы по главе 1

Системы распознавания лиц становятся все более распространенными в различных сферах деятельности, начиная от охраны безопасности и заканчивая маркетингом и рекламой. Они обладают значительным потенциалом для улучшения эффективности бизнес-процессов и повышения уровня безопасности. Однако, как и любая технология, системы распознавания лиц имеют некоторые недостатки.

Одним из главных недостатков систем распознавания лиц является возможность ложных срабатываний, когда система ошибочно идентифицирует личность. Это может привести к серьезным проблемам, например, задержанию невиновных людей или доступу злоумышленников к защищенным помещениям.

Другой недостаток систем распознавания лиц заключается в их неспособности распознавать лица, скрытые или измененные с помощью различных методов. Кроме того, системы распознавания лиц могут стать объектом злоупотреблений со стороны правительственных органов или частных компаний, что подчеркивает важность этических вопросов, связанных с использованием таких систем. В

заключение, можно сказать, что системы распознавания лиц имеют значительный потенциал для повышения эффективности и безопасности в различных сферах деятельности, однако их применение также подразумевает серьезные этические и законодательные вопросы, которые нужно учитывать при разработке и внедрении таких систем [14].

## Глава 2 Системы распознавания лиц

Рассмотрим популярные алгоритмы машинного обучения в системах распознавания лиц.

### 2.1 Алгоритмы классификации

Алгоритмы классификации: Эти алгоритмы принимают входные данные и классифицируют их в соответствии с заданным набором классов. В системах распознавания лиц, эти алгоритмы могут быть использованы для классификации изображений лиц на основе определенных параметров, таких как расстояние между глазами, форма лица и т.д [6].

Ниже приведены некоторые математические формулы, которые используются в алгоритмах классификации:

– линейная регрессия:

$$y = \beta_0 + \beta_{1x_1} + \beta_{1x_2} + \dots + \beta_{pxp} + \epsilon, \quad (1)$$

где  $y$  – зависимая переменная;

$x_1, x_2, \dots, x_p$  – независимые переменные;

$\beta_0, \beta_1, \beta_2, \dots$ ;

$\beta_p$  – коэффициенты регрессии,  $\epsilon$  – ошибка.

– функция стоимости:

$$J(\theta) = \frac{1}{m} \sum_{i=1}^m (-y^i \log(h_0(x^i)) - 1(1 - y^i) \log(1 - h_0(x^i))), \quad (2)$$

где  $m$  – количество образцов в обучающей выборке;

$y$  – истинное значение метки класса для  $i$ -го образца;

$x$  –  $i$ -й образец;

$h$  – предсказанная вероятность, что  $i$ -й образец относится к классу  $l$  [15].

Алгоритмы классификации применяются там, где необходимо разделить данные на определенные классы на основе их признаков. Вот некоторые примеры, когда лучше всего использовать алгоритмы классификации:

– классификация текстов. Алгоритмы классификации могут использоваться для классификации текстовых документов на определенные категории, такие как новости, отзывы или электронные письма,

– распознавание образов. Алгоритмы классификации могут использоваться для распознавания образов на изображениях, таких как лица, животные или автомобили,

– анализ тональности. Алгоритмы классификации могут использоваться для анализа тональности текстов, таких как отзывы или комментарии на социальных сетях,

– медицинская диагностика. Алгоритмы классификации могут использоваться для диагностики заболеваний на основе медицинских данных, таких как симптомы, тесты и история болезни,

– финансовый анализ. Алгоритмы классификации могут использоваться для анализа финансовых данных, таких как кредитный риск, финансовые мошенничества и инвестиционные возможности,

– прогнозирование. Алгоритмы классификации могут использоваться для прогнозирования событий или результатов на основе исторических данных, таких как предсказание погоды или результатов спортивных матчей.

В целом, алгоритмы классификации широко используются в различных областях, где необходимо классифицировать данные на основе их признаков. Они позволяют автоматически обрабатывать большие объемы данных и находить скрытые закономерности, что делает их очень полезными инструментами в анализе данных [3].

## 2.2 Нейронные сети

Нейронные сети: Нейронные сети – это многократно связанные нейроны, которые могут обрабатывать сложные данные, включая изображения. В системах распознавания лиц, нейронные сети могут быть использованы для обучения на большом наборе изображений и распознавания новых лиц на основе этого обучения [5].

Нейронные сети являются мощным инструментом для обработки и анализа сложных данных. Они состоят из множества нейронов, которые работают вместе, чтобы выполнить задачу. Вот некоторые из математических формул, которые используются в нейронных сетях:

– линейное преобразование:

$$z = Wx + b, \quad (3)$$

где  $W$  – матрица весов;

$x$  – входной вектор;

$b$  – вектор смещения;

$z$  – выходной вектор.

– функция активации:

$$a = g(z), \quad (4)$$

где  $g$  – функция активации;

$z$  – выходной вектор из линейного преобразования;

$a$  – выходной вектор после применения функции активации.

– градиентный спуск:

$$W = W - \alpha \nabla_{w} L, \quad (5)$$

где  $W$  – матрица весов;

$\alpha$  – скорость обучения;

$\nabla_{w^L}$  – градиент функции потерь по весам  $W$ .

Нейронные сети являются мощным инструментом машинного обучения, который может использоваться в различных областях. Вот несколько примеров, когда лучше всего применять нейронные сети:

– обработка изображений и видео. Нейронные сети могут использоваться для распознавания объектов на изображениях или в видео, а также для генерации изображений и видео,

– обработка естественного языка. Нейронные сети могут использоваться для анализа и генерации естественного языка, такого как обработка текстовых документов или создание чат-ботов,

– прогнозирование временных рядов. Нейронные сети могут использоваться для прогнозирования временных рядов, таких как цены на акции, продажи товаров или трафик на сайте,

– рекомендательные системы. Нейронные сети могут использоваться для создания рекомендательных систем, которые предлагают пользователям наиболее подходящие товары, услуги или контент на основе их предпочтений,

– распознавание речи. Нейронные сети могут использоваться для распознавания речи и преобразования ее в текст,

– анализ данных. Нейронные сети могут использоваться для анализа больших объемов данных, включая прогнозирование, классификацию и кластеризацию,

– анализ социальных сетей. Нейронные сети могут использоваться для анализа социальных сетей, таких как Twitter и Facebook, для выявления трендов, мнений и настроений пользователей,

– обработка сигналов. Нейронные сети могут использоваться для обработки сигналов, таких как звук, радиоволны или электроэнцефалограмма (ЭЭГ).

В целом, нейронные сети являются мощным инструментом машинного

обучения, который может быть использован во многих областях. Они особенно эффективны в областях, где данные имеют сложную структуру или много признаков, а также в областях, где необходимо выявлять скрытые закономерности [10].

### **2.3 Алгоритмы глубокого обучения**

Алгоритмы глубокого обучения: Это тип нейронных сетей, который использует множество слоев для обработки и анализа данных. В системах распознавания лиц, алгоритмы глубокого обучения могут быть использованы для обучения на большом количестве изображений лиц и распознавания новых лиц на основе этого обучения [11].

Нейронные сети являются одним из видов алгоритмов глубокого обучения, так когда лучше использовать глубокие нейронные сети вместо более простых моделей машинного обучения [18].

Нейронные сети являются мощным инструментом для обработки и анализа сложных данных. Они состоят из множества нейронов, которые работают вместе, чтобы выполнить задачу.

Вот несколько примеров, когда лучше использовать глубокие нейронные сети:

- когда имеется большой объем данных. Глубокие нейронные сети могут извлекать более сложные закономерности из больших объемов данных, чем простые модели машинного обучения,

- когда данные имеют сложную структуру. Глубокие нейронные сети могут обрабатывать данные с более сложной структурой, такие как изображения, видео, текст и звук,

- когда необходимо выделять признаки автоматически. Глубокие нейронные сети могут автоматически выделять признаки из данных, что может быть сложно или невозможно в простых моделях машинного обучения,

- когда необходимо учитывать зависимости между признаками.

Глубокие нейронные сети могут учитывать зависимости между признаками и использовать их для более точного прогнозирования,

– когда необходимо решить задачу классификации или регрессии с высокой точностью. Глубокие нейронные сети могут достичь высокой точности в решении задач классификации или регрессии, особенно если обучены на больших объемах данных [17].

В целом, глубокие нейронные сети лучше всего подходят для задач, которые требуют извлечения сложных закономерностей из больших объемов данных с более сложной структурой. Если же задача относительно простая и данных немного, то можно использовать более простые модели машинного обучения [12].

## 2.4 Алгоритмы сопоставления шаблонов

Алгоритмы сопоставления шаблонов: Эти алгоритмы используются для сопоставления новых изображений лиц с уже известными шаблонами. В системах распознавания лиц, эти алгоритмы могут быть использованы для сопоставления новых лиц с уже имеющимися в базе данных изображениями [16].

Одним из наиболее распространенных методов сопоставления шаблонов является метод наименьших квадратов (Least Squares Method). Он использует следующую формулу:

$$a = \frac{\sum y - b \sum x}{n}, \quad (6)$$

где  $a$  - то, что мы называем точкой пересечения линии с осью  $y$  и она равна приблизительным фиксированным затратам на любом уровне деятельности;

$b$  - наклон линии и представляет собой затраты не единицу продукции [26].

Алгоритмы сопоставления шаблонов (pattern matching) обычно используются в задачах распознавания образов или объектов на изображениях или в звуковых сигналах. Эти алгоритмы работают путем сравнения входных данных с заранее заданными шаблонами и выбирают наиболее близкий шаблон [2].

Одним из примеров применения алгоритмов сопоставления шаблонов является распознавание лиц на изображениях. Здесь шаблоном может быть изображение лица, а алгоритм сопоставления будет искать на изображении входного потока лицо, наиболее близкое к заданному шаблону [4].

Одним из главных преимуществ алгоритмов сопоставления шаблонов является их относительная простота и быстрота работы. В отличие от более сложных алгоритмов машинного обучения, таких как нейронные сети, алгоритмы сопоставления шаблонов не требуют длительного процесса обучения на больших объемах данных. Они также не требуют мощных вычислительных ресурсов для работы [13].

Однако, применение алгоритмов сопоставления шаблонов может быть ограничено в задачах с большими объемами данных или сложной структурой данных. В этих случаях более сложные алгоритмы машинного обучения, такие как нейронные сети, могут показать более высокую точность и эффективность [1].

Таким образом, лучше всего применять алгоритмы сопоставления шаблонов в задачах, где необходима быстрая обработка данных с относительно простой структурой. В более сложных задачах или при работе с большими объемами данных, лучше использовать более сложные алгоритмы машинного обучения [7].



## 2.5 Алгоритмы на основе SVM

Алгоритмы на основе SVM (Support Vector Machines): Эти алгоритмы используются для классификации данных, основанных на разделении двух классов точками на графике. В системах распознавания лиц, SVM могут быть использованы для определения, принадлежит ли изображение лица к заданной базе данных или нет.

Алгоритмы на основе SVM (Support Vector Machine) часто используются в задачах классификации и регрессии. Они особенно полезны в случаях, когда имеется большой объем данных с множеством признаков.

Вот несколько примеров, когда лучше применять алгоритмы на основе SVM:

- классификация текстовых документов: SVM часто используются для классификации текстовых документов по категориям, таким как новости, научные статьи, спортивные новости и т. д. Это можно сделать, используя алгоритмы на основе SVM, которые могут различать различные категории текста на основе их содержания и определенных признаков,

- обнаружение спама: Алгоритмы на основе SVM также могут использоваться для обнаружения спама в электронной почте. Они могут классифицировать электронные письма как спам или не спам, используя признаки, такие как наличие определенных слов или фраз,

- анализ изображений: SVM может использоваться для классификации изображений, например, для распознавания объектов на изображении. Алгоритмы на основе SVM могут использоваться для классификации изображений, определяя, содержит ли изображение объект определенного класса,

- регрессия: Алгоритмы на основе SVM также могут использоваться для решения задач регрессии, например, для прогнозирования цен на недвижимость на основе различных признаков.

В целом, алгоритмы на основе SVM могут быть полезны в любой задаче,

где необходимо классифицировать данные или решить задачу регрессии, особенно в случаях, когда имеется большой объем данных с множеством признаков. Однако, как и в случае с другими алгоритмами машинного обучения, эффективность SVM зависит от тщательного выбора признаков и настройки параметров модели для достижения наилучшей производительности [9].

## 2.6 Метод главных компонент

Метод главных компонент (Principal Component Analysis, PCA): Этот алгоритм используется для сокращения размерности данных путем поиска главных компонент в данных. В системах распознавания лиц, PCA может быть использован для сокращения размерности изображений лиц перед их классификацией или сопоставлением с шаблонами, что может повысить эффективность распознавания лиц и ускорить процесс обработки данных [19].

Некоторые из математических формул, используемых в алгоритме PCA, включают:

Вычисление среднего значения исходных данных:

$$\mu = \frac{1}{n} \sum_{i=1}^n x_i, \quad (7)$$

где  $\mu$  - среднее значение;

$N$  - количество исходных данных;

$x_i$  = вектор  $i$ -го элемента данных.

Центрирование данных:

$$X_{centered} = X - \bar{X}, \quad (8)$$

где  $X$  - матрица данных;

$\bar{X}$  - среднее значение каждого столбца матрицы.

Для вычисления матрицы ковариации необходимо умножить центрированную матрицу данных на ее транспонированную версию:

$$\text{Cov}(X_{\text{centered}}) = \frac{1}{n-1} X_{\text{centered}}^T X_{\text{centered}}, \quad (9)$$

где  $n$  - количество объектов в матрице данных.

Вычисление собственных векторов и собственных значений матрицы ковариации:

Далее необходимо найти собственные векторы и собственные значения матрицы ковариации. Собственные векторы являются новым ортогональными базисными векторами, на которые будут проектироваться исходные данные [8].

$$\text{Cov}(X_{\text{centered}})v = \lambda v, \quad (10)$$

где  $v$  - собственный вектор;

$\lambda$  - собственное значение.

Метод главных компонент (PCA) – это метод, который используется для снижения размерности данных, сохраняя при этом как можно больше информации. Когда количество признаков в данных очень большое, метод главных компонент может быть очень полезен для упрощения их анализа [20].

Вот несколько примеров, когда лучше применять метод главных компонент:

– визуализация данных: Если данные содержат большое количество признаков, трудно визуализировать их в пространстве меньшей размерности. Метод главных компонент может быть использован для снижения размерности данных и получения нового набора признаков, которые могут

быть визуализированы на двумерном или трехмерном графике,

– удаление шума из данных: Метод главных компонент также может быть использован для удаления шума из данных. Когда данные содержат шумовые значения, метод главных компонент может отделить сигналы от шума и позволить извлечь более значимые признаки,

– ускорение обучения моделей: Когда количество признаков в данных очень большое, обучение моделей может занять много времени. Метод главных компонент может быть использован для снижения размерности данных и ускорения обучения моделей, при этом не теряя существенной части информации,

– снижение затрат на хранение данных: Если данные занимают много места в памяти, то метод главных компонент может быть использован для снижения затрат на хранение данных. С помощью метода главных компонент можно сократить количество признаков в данных, что позволит снизить объем памяти, необходимый для их хранения [10].

В целом, метод главных компонент может быть очень полезен в любой ситуации, когда данные содержат большое количество признаков и когда необходимо упростить анализ данных без потери существенной информации. Однако следует иметь в виду, что метод главных компонент также имеет свои ограничения, например, он может потерять информацию, которая не может быть представлена в виде главных компонент [22].

## **2.7 Сравнение алгоритмов**

Сравнение всех алгоритмов машинного обучения, перечисленных в предыдущих ответах, является достаточно сложной задачей, так как каждый из них имеет свои преимущества и недостатки, а также подходит для решения определенных задач. Однако, я могу предоставить общее сравнение основных алгоритмов машинного обучения по следующим критериям:

– результаты: Некоторые алгоритмы могут давать лучшие результаты в

определенных задачах, чем другие. Например, нейронные сети могут давать лучшие результаты в задачах классификации изображений, тогда как SVM может быть более эффективен в задачах классификации текста.

– сложность: Некоторые алгоритмы, такие как SVM, логистическая регрессия и деревья решений, являются относительно простыми для понимания и реализации, тогда как нейронные сети и алгоритмы глубокого обучения могут быть более сложными в реализации и требовать большого количества вычислительных ресурсов,

– объем данных: Некоторые алгоритмы, такие как SVM и K–ближайших соседей, могут работать хорошо на небольших объемах данных, тогда как нейронные сети и алгоритмы глубокого обучения требуют большого объема данных для обучения,

– переобучение: Некоторые алгоритмы, такие как SVM и деревья решений, склонны к переобучению, особенно при использовании большого количества признаков, тогда как нейронные сети и алгоритмы глубокого обучения могут быть более устойчивыми к переобучению,

– время обучения: Некоторые алгоритмы, такие как SVM и K–ближайших соседей, обучаются быстро, тогда как нейронные сети и алгоритмы глубокого обучения могут требовать большого количества времени для обучения,

– скорость обучения и предсказания. Некоторые алгоритмы, например, SVM и логистическая регрессия, обучаются быстрее, чем глубокие нейронные сети. Однако, для больших и сложных наборов данных может потребоваться длительное время на обучение любого алгоритма. Кроме того, некоторые алгоритмы могут быть быстрее при предсказании, что может быть важно в реальном времени приложениях,

– интерпретируемость. Некоторые алгоритмы, такие как логистическая регрессия и решающие деревья, легко интерпретируются, то есть позволяют легко понять, какие признаки влияют на предсказания. Однако, глубокие нейронные сети, в особенности с большим количеством слоев, могут быть

сложными для интерпретации,

- разреженность. Некоторые алгоритмы, например, логистическая регрессия и SVM, могут использовать разреженные матрицы, что может сильно ускорить процесс обучения и уменьшить потребление памяти. В то время как другие алгоритмы, такие как нейронные сети и деревья решений, не могут использовать разреженность, что может снизить их производительность,

- наличие готовых библиотек. Некоторые алгоритмы машинного обучения имеют готовые библиотеки, которые могут значительно упростить процесс их реализации и использования. Например, библиотеки Scikit-learn, TensorFlow и Keras содержат реализации многих алгоритмов машинного обучения.

## Выводы по главе 2

Вторая глава посвящена обзору и анализу алгоритмов машинного обучения для систем распознавания лиц.

Результаты проделанной работы позволили сделать следующие выводы: Алгоритмы машинного обучения в системах распознавания лиц используют различные математические методы, включая:

- матричные операции: для представления и обработки изображений используются матрицы, а операции над ними (например, умножение, вычитание, свёртка) используются для извлечения признаков и создания моделей,

- статистика: используется для анализа распределения данных и извлечения полезной информации. Например, метод главных компонент (PCA) используется для уменьшения размерности данных и выделения наиболее значимых признаков,

- теория вероятностей: используется для моделирования распределений признаков и классов, а также для оценки вероятности правильности

классификации. Методы, такие как гауссовские смеси и байесовский классификатор, основаны на теории вероятностей.

Линейная алгебра: используется для решения систем линейных уравнений, определения собственных значений и векторов, а также для нахождения оптимальных весов модели,

– оптимизация: используется для настройки параметров моделей на основе данных обучения. Методы, такие как стохастический градиентный спуск и метод опорных векторов (SVM), основаны на оптимизации,

– глубокое обучение: используется для создания и обучения нейронных сетей, которые могут моделировать сложные нелинейные зависимости между признаками и классами.

Все эти методы используются в различных алгоритмах машинного обучения в системах распознавания лиц, которые позволяют извлекать признаки, создавать модели и классифицировать изображения на основе обучающих данных [21].

В целом, выбор алгоритма машинного обучения зависит от типа задачи, доступных ресурсов и требуемых результатов.

Когда речь идет о задачах классификации, то алгоритмы на основе SVM могут быть более эффективными, чем нейронные сети или алгоритмы сопоставления шаблонов, особенно когда имеется большой объем данных. Они также могут быть лучше подходят для данных с высокой размерностью. Однако, они могут быть менее эффективными в задачах, где данные имеют сложную структуру или когда данные не являются линейно разделимыми [22].

Метод главных компонент, как правило, используется для уменьшения размерности данных и упрощения моделей. Он может быть полезен, когда данные имеют большую размерность и когда необходимо избавиться от лишних признаков, которые не оказывают влияния на результат. Также этот метод может помочь избежать проблемы переобучения моделей.

Алгоритмы глубокого обучения наиболее эффективны в задачах, связанных с обработкой больших объемов данных, таких как распознавание

речи, компьютерное зрение, обработка естественного языка и т. д. Они могут обнаруживать сложные зависимости между признаками и построить высококачественные модели, которые могут обобщаться на новые данные. Однако, они могут требовать большого объема данных и вычислительных ресурсов для обучения, а также не всегда могут обеспечивать интерпретируемость результата [23].

Алгоритмы сопоставления шаблонов могут быть эффективны в задачах, связанных с распознаванием образов или лиц, где шаблоны могут быть определены заранее. Они могут быть полезны, когда данные имеют небольшую размерность и когда существует возможность создания точных шаблонов. Однако, они могут быть менее эффективными в задачах, где шаблоны сложно определить заранее или когда данные имеют большую размерность [24].

Конечный выбор алгоритма машинного обучения зависит от многих факторов, таких как размер и тип данных, требования к точности, доступные вычислительные ресурсы, время обучения и т. д.



### Глава 3 Программная реализация и тестирование алгоритма машинного обучения в системах распознавания лиц

Кирби и Сирович (1990) впервые разработали метод, известный как Eigenface, который используется для представления и распознавания лиц на основе анализа основных компонентов. С помощью этого метода Терк и Пентланд преобразовали все изображение лица в векторы и вычислили собственные лица с набором выборок. PCA смог получить данные, представляющие лицо на оптимальном уровне с данными, полученными из изображения. Различные уровни лица и освещенности одного и того же человека были оценены как слабое место PCA [25].

Алгоритм для распознавания лиц, который будет использован ниже, основан на работе Терка и Пентланда.

Набор данных Оливетти

Краткая информация о наборе данных Olivetti:

- изображения лиц, сделанные в период с апреля 1992 года по апрель 1994 года.
- есть десять разных изображений каждого из 40 разных людей.
- в наборе данных 400 изображений лиц.
- изображения лиц были сделаны в разное время, с разным освещением, выражением лица и деталями лица.
- все изображения лиц имеют чёрный фон
- изображения имеют уровень серого
- размер каждого изображения 64x64.
- значения пикселей изображения были масштабированы до интервала [0, 1]
- имена 40 человек были закодированы в целое число от 0 до 39.

Давайте проверим эту информацию:

Программный код который выводит эту информацию представлен на рисунке 1.

```

data=np.load(r"C:\Users\Alex\Desktop\facerec\olivetti_faces.npy")
target=np.load(r"C:\Users\Alex\Desktop\facerec\olivetti_faces_target.npy")
print("Имеется {} фотографий в наборе данных".format(len(data)))
print("Имеется {} уникальных объектов в наборе данных".format(len(np.unique(target))))
print("Размер каждого изображения {}x{}".format(data.shape[1],data.shape[2]))
print("Значения пикселей были масштабированы до интервала [0,1] :{}".format(data[0][0,:4]))
print("Уникальные номера объектов:",np.unique(target))
print("unique target number:",np.unique(target))

```

✓ 0.6s

```

Имеется 400 фотографий в наборе данных
Имеется 40 уникальных объектов в наборе данных
Размер каждого изображения 64x64
Значения пикселей были масштабированы до интервала [0,1] :[0.30991736 0.3677686 0.41735536 0.44214877]
Уникальные номера объектов: [ 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39]
unique target number: [ 0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23
 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39]

```

Рисунок 1 – Исходные данные

Рисунок 2 представляет программный код, который выводит уникальные объекты.

```

def show_40_distinct_people(images, unique_ids):
    #Creating 4X10 subplots in 18x9 figure size
    fig, axarr=plt.subplots(nrows=4, ncols=10, figsize=(18, 9))
    #For easy iteration flattened 4X10 subplots matrix to 40 array
    axarr=axarr.flatten()

    #iterating over user ids
    for unique_id in unique_ids:
        image_index=unique_id*10
        axarr[unique_id].imshow(images[image_index], cmap='gray')
        axarr[unique_id].set_xticks([])
        axarr[unique_id].set_yticks([])
        axarr[unique_id].set_title("face id:{}".format(unique_id))
    plt.suptitle("There are 40 distinct people in the dataset")

```

✓ 0.6s

```

show_40_distinct_people(data, np.unique(target))

```

Рисунок 2 – Уникальные объекты

Результат работы кода представлен на рисунке 3

There are 40 distinct people in the dataset



Рисунок 3 – Фотографии уникальных объектов

Как видно из фотогалереи выше, набор данных содержит 40 различных изображений лиц, принадлежащих людям.

Каждый из этих 40 людей предоставил 10 фотографий тем самым создав набор данных состоящий из 400 изображений.

Ниже на рисунке 4 представлен код результатом которого будет вывод всех 10 изображений выбранных 5 человек.

```
def show_10_faces_of_n_subject(images, subject_ids):
    cols=10# each subject has 10 distinct face images
    rows=(len(subject_ids)*10)/cols #
    rows=int(rows)

    fig, axarr=plt.subplots(nrows=rows, ncols=cols, figsize=(18,9))
    #axarr=axarr.flatten()

    for i, subject_id in enumerate(subject_ids):
        for j in range(cols):
            image_index=subject_id*10 + j
            axarr[i,j].imshow(images[image_index], cmap="gray")
            axarr[i,j].set_xticks([])
            axarr[i,j].set_yticks([])
            axarr[i,j].set_title("face id:{}".format(subject_id))

✓ 0.2s

#You can playaround subject_ids to see other people faces
show_10_faces_of_n_subject(images=data, subject_ids=[0,5, 21, 24, 36])
```

Рисунок 4 – Код, который выводит 10 изображений выбранных 5  
ЧЕЛОВЕК

Результат работы кода представлен на рисунке 5

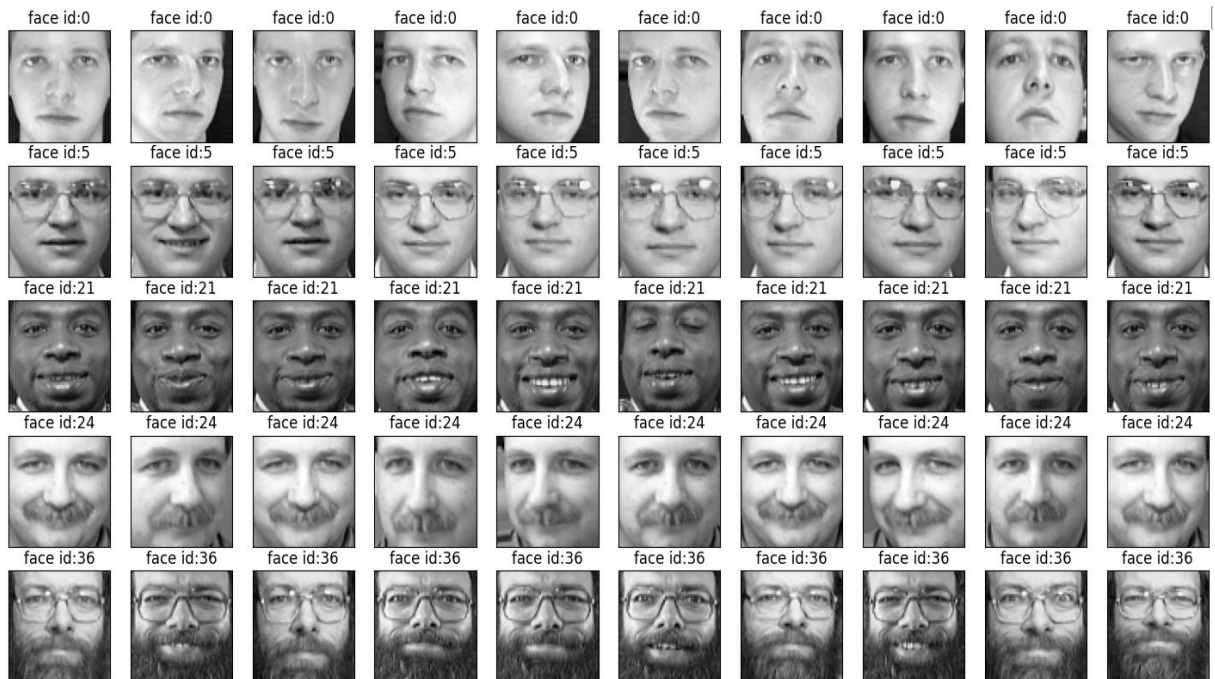


Рисунок 5 – 10 фотографий выбранных 5 человек

Каждое лицо субъекта имеет разные характеристики в зависимости от освещения, выражения лица и деталей лица (очки, борода).

Модели машинного обучения могут работать с векторами. Поскольку данные изображения представлены в матричной форме, их необходимо преобразовать в вектор.

Программный код реализующий векторное преобразование представлен на рисунке 6

```
#Поскольку данные изображения представлены в матричной форме, их необходимо преобразовать в вектор.  
X=data.reshape((data.shape[0],data.shape[1]*data.shape[2]))  
print("X shape:",X.shape)  
✓ 0.1s  
X shape: (400, 4096)
```

Рисунок 6 – Код реализующий векторное преобразование

Набор данных содержит 10 изображений лиц для каждого субъекта. Из изображений лиц 70 процентов будут использоваться для обучения, 30 процентов — для тестирования. Можно использовать разное соотношение тренировочных и обучающих наборов, 70/30 наиболее распространённый и универсальный вариант. Программный код реализующий деление данных на обучающую и тестовую выборки представлен на рисунке 7.

```
X_train, X_test, y_train, y_test=train_test_split(X, target, test_size=0.3, stratify=target, random_state=0)  
print("X_train Размер:",X_train.shape)  
print("y_train Размер:{}".format(y_train.shape))  
✓ 0.7s  
X_train Размер: (280, 4096)  
y_train Размер:(280,)
```

Рисунок 7 – Код реализующий деление данных

Напишем код результатом которого будет диаграмма которая отображает количество тренировочных образцов для каждого объекта. Он

представлен на рисунке 8.

```
X_train, X_test, y_train, y_test=train_test_split(X, target, test_size=0.3, stratify=target, random_state=0)
print("X_train Размер:",X_train.shape)
print("y_train Размер:{}".format(y_train.shape))
```

✓ 0.7s

Рисунок 8 – Код который выводит диаграмму

Как и предполагалось исходя из выбранной нами выборки 70% на обучение 30% процентов на проверку, для каждого объекта, который содержит в себе 10 фотографии выделено 7 фото на тренировку. Результат на рисунке 9.

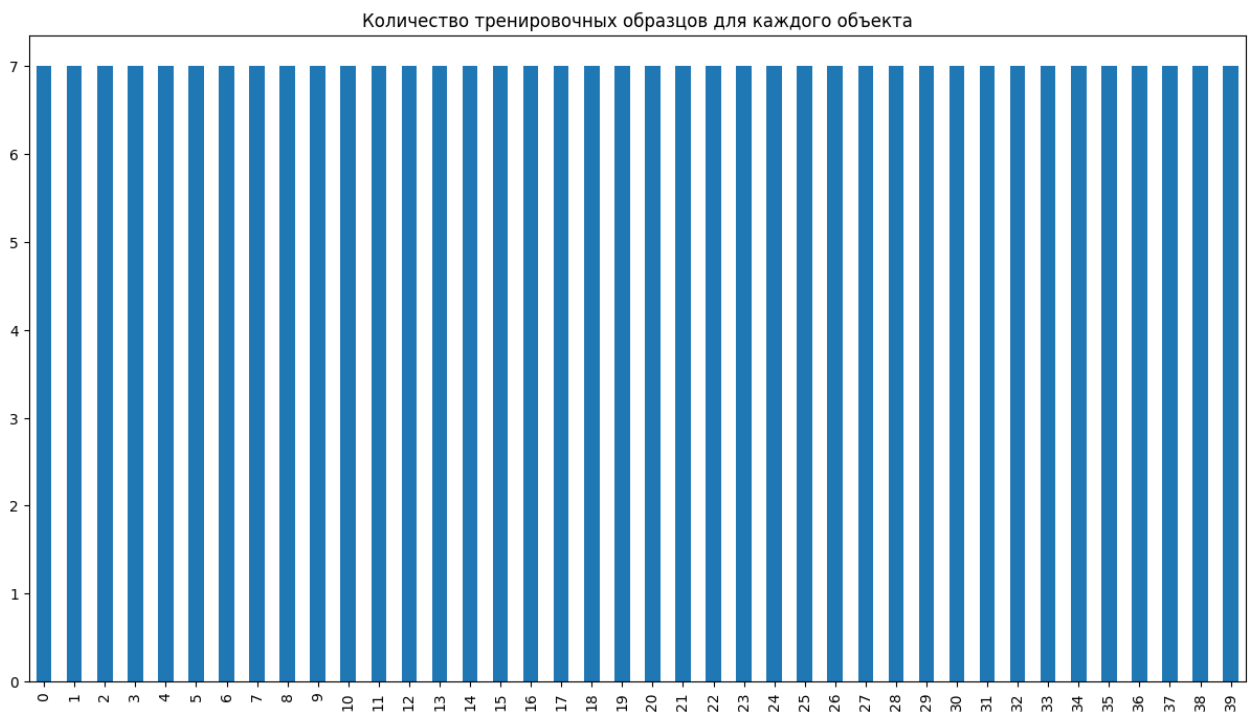


Рисунок 9 – Диаграмма, которая отображает количество тренировочных образцов для каждого объекта

Методы машинного обучения делятся на два: обучение с учителем и обучение без учителя. При обучении с учителем набор данных делится на две основные части: «данные» и «выход». Данные содержат значения выборки в наборе данных, а «выходные данные» содержат класс (для классификации) или целевое значение (для регрессии). При неконтролируемом обучении набор

данных состоит только из раздела данных [17].

Обучение без учителя обычно делится на две части: преобразование данных и кластеризация. В этом исследовании преобразование данных будет осуществляться с использованием неконтролируемого обучения. Неконтролируемые методы преобразования позволяют упростить интерпретацию данных компьютерами и людьми.

Наиболее распространёнными приложениями неконтролируемого преобразования являются уменьшение размера данных. В процессе уменьшения размера размер данных уменьшается.

Анализ основных компонентов (PCA) — это метод, который позволяет представлять данные в меньшем размере. Согласно этому методу данные преобразуются в новые компоненты, а размер данных уменьшается за счёт выбора наиболее важных компонентов [16].

Код реализующий проекцию PCA для 10 объектов представлен на рисунке 10.

```
from sklearn.decomposition import PCA
pca=PCA(n_components=2)
pca.fit(X)
X_pca=pca.transform(X)
✓ 0.2s

number_of_people=10
index_range=number_of_people*10
fig=plt.figure(figsize=(10,8))
ax=fig.add_subplot(1,1,1)
scatter=ax.scatter(X_pca[:index_range,0],
                  X_pca[:index_range,1],
                  c=target[:index_range],
                  s=10,
                  cmap=plt.get_cmap('jet', number_of_people)
                  )

ax.set_xlabel("First Principle Component")
ax.set_ylabel("Second Principle Component")
ax.set_title("PCA projection of {} people".format(number_of_people))

fig.colorbar(scatter)
```

Рисунок 10 – Код реализующий проекцию PCA

Результат работы кода на рисунке 11

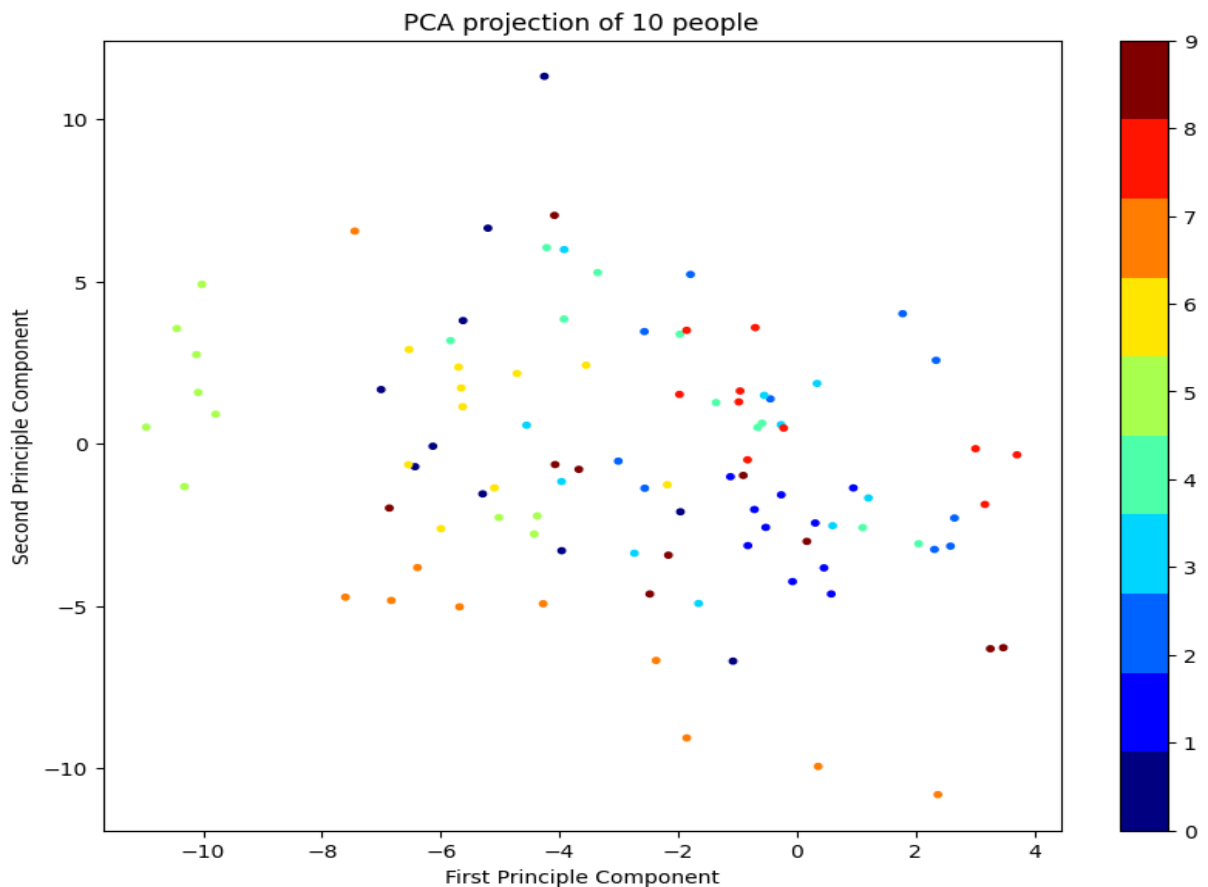


Рисунок 11 – Проекция PCA

Программный код реализующий нахождение оптимального количества основных компонентов представлен на рисунке 12.

```

pca=PCA()
pca.fit(X)

plt.figure(1, figsize=(12,8))

plt.plot(pca.explained_variance_, linewidth=2)

plt.xlabel('Components')
plt.ylabel('Explained Variaces')
plt.show()

```

Рисунок 12 – Код, реализующий нахождение основных компонентов



Результат работы кода представлен на рисунке 13.

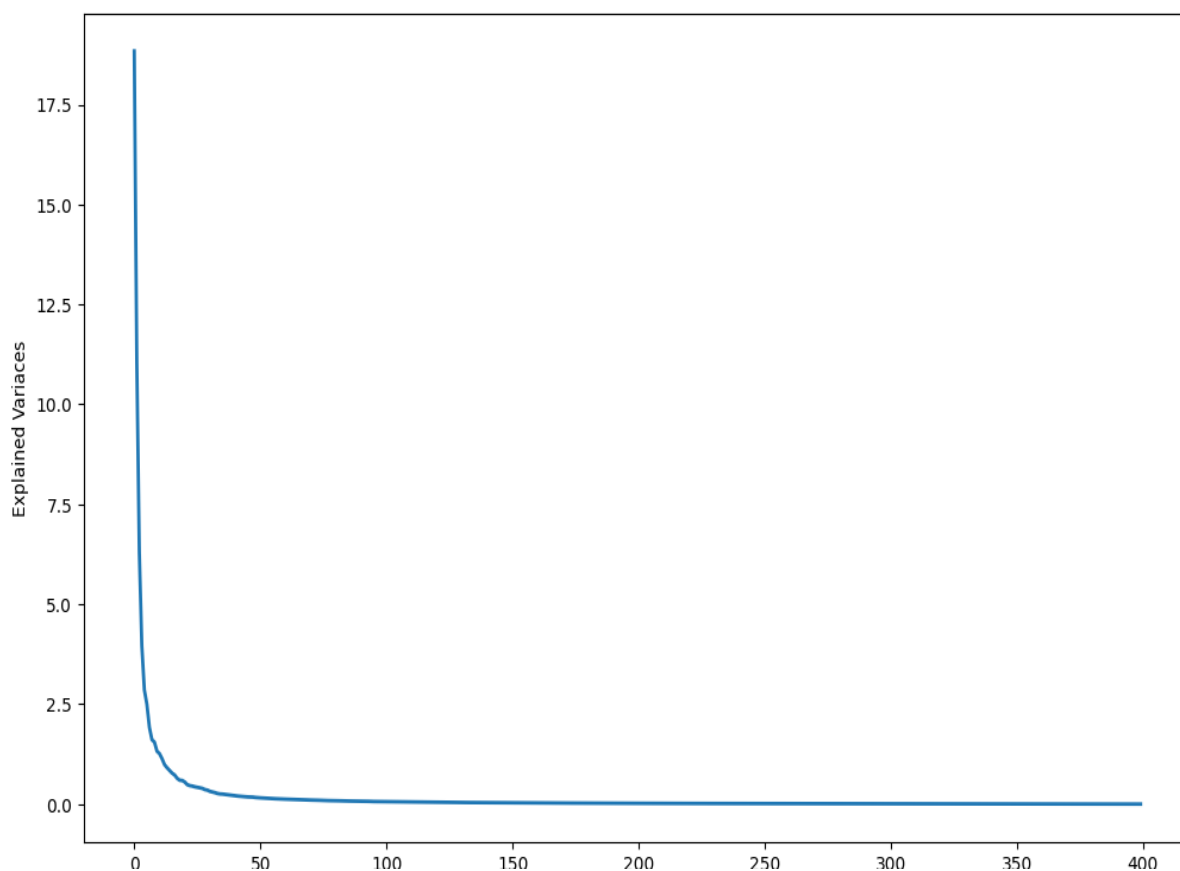


Рисунок 13 – График, отображающий эффективность увеличения КОМПОНЕНТОВ

На рисунке выше видно, что 90 и более компонент PCA представляют одни и те же данные. Теперь давайте проведем процесс классификации, используя 90 компонент PCA. Программный код на рисунке 14.

```
n_components=90
pca=PCA(n_components=n_components, whiten=True)
pca.fit(X_train)
```

Рисунок 14 – Код, реализующий процесс классификации

Покажем “среднее лицо” оно демонстрирует изображение человека

признаки которого максимально универсальны. Программный код реализующий это и результат его работы представлены на рисунке 15.

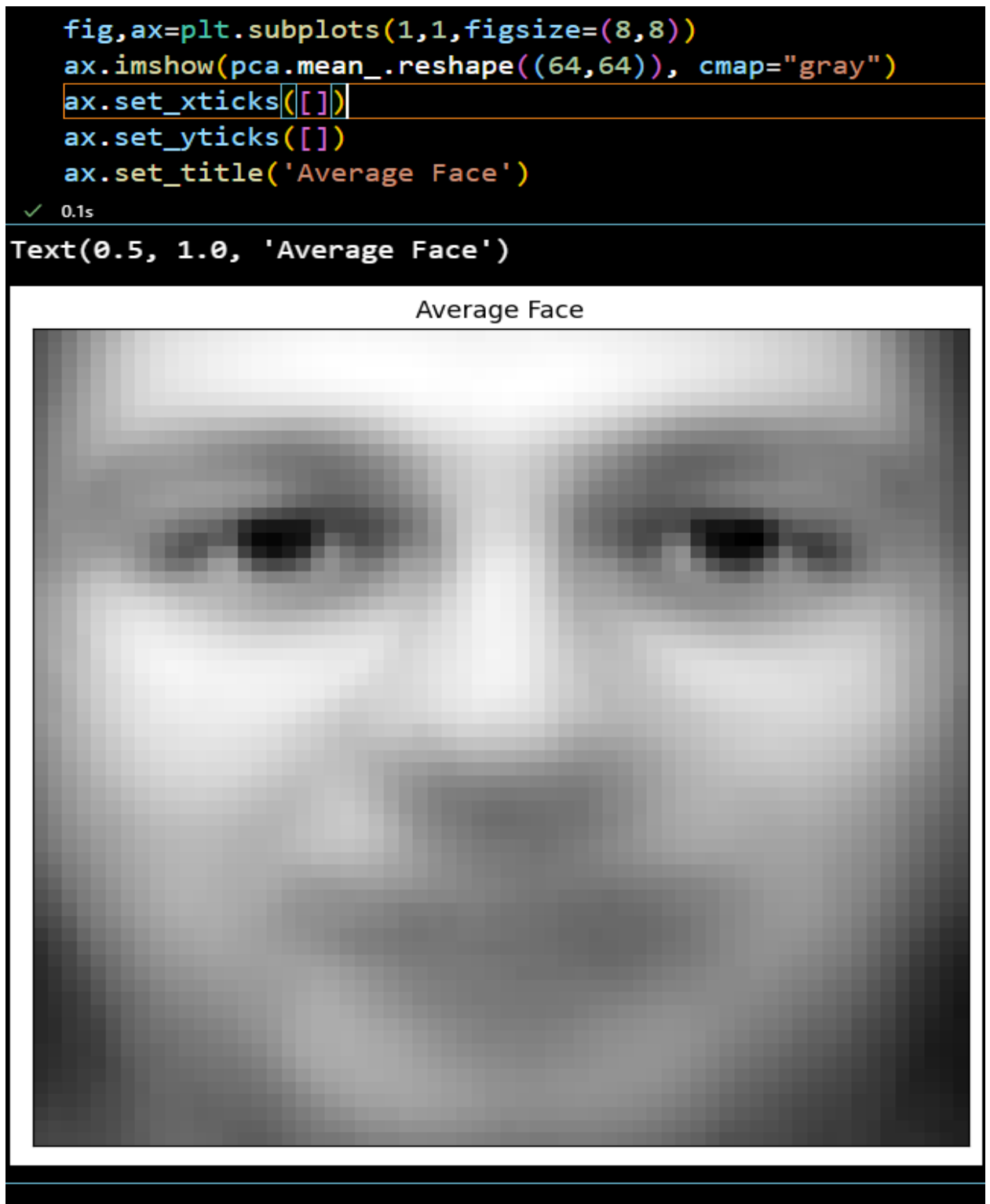


Рисунок 15 – “Среднее лицо”

Неформально собственные лица можно рассматривать как набор «стандартизированных ингредиентов лица», полученных в результате статистического анализа многих изображений лиц. Любое человеческое лицо можно рассматривать как комбинацию этих стандартных лиц. Например, чье-то лицо может состоять из среднего лица плюс 10 % от собственного лица 1, 55 % от собственного лица 2 и даже –3 % от собственного лица.

Мы выяснили что наиболее оптимальным числом для количества компонентов стандартизированных ингредиентов лица будет 90. Выведем 90 “средних/стандартных” лиц. Программный код реализующий это представлен на рисунке 16.

```
number_of_eigenfaces=len(pca.components_)
eigen_faces=pca.components_.reshape((number_of_eigenfaces, data.shape[1], data.shape[2]))

cols=10
rows=int(number_of_eigenfaces/cols)
fig, axarr=plt.subplots(nrows=rows, ncols=cols, figsize=(15,15))
axarr=axarr.flatten()
for i in range(number_of_eigenfaces):
    axarr[i].imshow(eigen_faces[i],cmap="gray")
    axarr[i].set_xticks([])
    axarr[i].set_yticks([])
    axarr[i].set_title("eigen id:{}".format(i))
plt.suptitle("All Eigen Faces".format(10* "=", 10* "="))
```

Рисунок 16 – Код, который демонстрирует 90 “средних лиц”

Результат работы код представлен на рисунке 17.

All Eigen Faces



Рисунок 17 – 90 “средних лиц”

Программный код позволяющий взглянуть на результаты классификации и сами результаты представлены на рисунке 18.

```
X_train_pca=pca.transform(X_train)
X_test_pca=pca.transform(X_test)
✓ 0.3s

clf = SVC()
clf.fit(X_train_pca, y_train)
y_pred = clf.predict(X_test_pca)
print("accuracy score:{:.2f}".format(metrics.accuracy_score(y_test, y_pred)))
✓ 0.3s

accuracy score:0.92
```

Рисунок 18 – Код, который выводит точность алгоритма

Таким образом мы получили точность в 92% процента, но ее можно увеличить различными способами например линейным дискриминантным анализом

Линейный дискриминантный анализ это классификатор с линейной границей решения, созданный путем подгонки условных плотностей класса к данным и использования правила Байеса [15].

Модель подбирает Гауссову плотность для каждого класса, предполагая, что все классы имеют одну и ту же ковариационную матрицу [14].

Подобранную модель также можно использовать для уменьшения размерности входных данных путем проецирования их в наиболее различимых направлениях с использованием метода преобразования.

Программный код реализующий линейный дискриминантный анализ для повышения точности алгоритма и результат его работы представлен на рисунке 19

```

from sklearn.model_selection import LeaveOneOut
loo_cv=LeaveOneOut()
clf=LinearDiscriminantAnalysis()
cv_scores=cross_val_score(clf,
                           X_pca,
                           target,
                           cv=loo_cv)
print("{} Leave One Out cross-validation mean accuracy score: {:.2f}".format(clf.__class__.__name__,
                                                                              cv_scores.mean()))

```

✓ 8.9s

LinearDiscriminantAnalysis Leave One Out cross-validation mean accuracy score:0.98

Рисунок 19 – Код реализующий линейный дискриминантный анализ

Таким образом мы увеличиваем точность работы алгоритма до 98 процентов.

### Выводы по главе 3

Третья глава посвящена программной реализации и тестированию алгоритма машинного обучения для системы распознавания лиц.

Результаты проделанной работы позволили сделать следующие выводы:

- в языке Python имеется множество библиотек, который упрощают и ускоряют разработку в области анализа данных и машинного обучения данные язык занимает первое место по популярности в этой отрасли;
- использование линейного дискриминантного анализа значительно повышает точность модели распознавания лиц.

Как показали результаты тестирования, точность алгоритма составила 92%, а при использовании линейного дискриминантного анализа увеличивается до 98% что вполне соответствует требованиям по эффективности.

## Заключение

Выпускная квалификационная работа посвящена актуальной проблеме исследования и практического применения алгоритмов машинного обучения в системах распознавания лиц.

В настоящее время разработано много всевозможных алгоритмов, которые могут быть применены в системах распознавания лиц.

Помимо конфигураций и вычислительной мощности используемых для запуска алгоритма одним из критериев выбора конкретного метода является эффективность данного способа при реализации конкретной задачи.

Цель бакалаврской работы – исследование, сравнение и реализация алгоритмов машинного обучения в системах распознавания лиц.

Для достижения данной цели в процессе работы над бакалаврской работой решены следующие задачи:

Выполнена постановка задачи исследования и проанализированы различные алгоритмы машинного обучения. Исходя из проведённого сравнения алгоритмов машинного обучения для задачи распознавания лиц, можно сделать следующие выводы:

- алгоритмы на основе глубокого обучения, такие как свёрточные нейронные сети, обладают лучшей точностью в распознавании лиц, по сравнению с другими алгоритмами машинного обучения, такими как РСА, SVM и другими,

- однако, алгоритмы на основе глубокого обучения являются более сложными в реализации и требуют значительно больших вычислительных ресурсов для обучения и работы с системой, чем более простые алгоритмы, такие как РСА,

- алгоритмы на основе глубокого обучения также обладают более высокой устойчивостью к изменениям, таким как изменение положения, освещения и выражения лица, чем другие алгоритмы машинного обучения,

- при работе с большими объемами данных и использовании

распределенных систем, лучше использовать алгоритмы, которые легко масштабируются и могут быть эффективно использованы на большом количестве устройств,

– в задачах, где требуется быстрая обработка изображений, эффективнее использовать алгоритмы, которые работают быстрее и могут быть эффективно использованы на устройствах с ограниченными вычислительными ресурсами.

Таким образом, выбор определенного алгоритма машинного обучения для задачи распознавания лиц должен зависеть от конкретных требований и ограничений, таких как точность, сложность, устойчивость к изменениям, масштабируемость и скорость обработки.

Выполнена программная реализация и тестирования алгоритма машинного обучения для распознавания лиц. В качестве примера использован популярный алгоритм PCA. Выполнена реализация данного алгоритма на языке программирования Python. Как показали результаты тестирования алгоритма его точность составила 98 процентов.

Результаты бакалаврской работы представляют научно–практический интерес и могут быть рекомендованы для анализа и программной реализации методов и алгоритмов машинного обучения в системах распознавания лиц.



## Список используемой литературы и используемых источников

1. Алгоритмы распознавания лиц в машинном обучении [Электронный ресурс]: URL: <https://habr.com/ru/post/336266/>
2. Алгоритмы распознавания лиц в условиях ограниченной выборки [Электронный ресурс]: URL: <https://www.nkj.ru/archive/articles/32371/>
3. Анализ методов распознавания лиц в машинном обучении [Электронный ресурс]: URL: <https://www.elibrary.ru/item.asp?id=29457633>
4. Башмаков С.А., Журчев А.А. "Распознавание лиц с помощью компьютерного зрения и глубокого обучения", Д. 5-е издание 2019 г.
5. Избачков Ю., Петров В. "Машинное обучение и распознавание лиц".-Питер 2009.
6. Искусственный интеллект в распознавании лиц: как это работает [Электронный ресурс]: URL: <https://www.rewizor.ru/articles/iskusstvennyj-intellekt-v-raspoznavanii-lic-kak-eto-rabotaet>
7. Как работают алгоритмы распознавания лиц [Электронный ресурс]: URL: <https://aftershock.news/?q=node/747178>
8. Машинное обучение в задачах распознавания лиц [Электронный ресурс]: URL: <https://www.digitalru.net/news/machine-learning-v-zadachakh-raspoznavaniya-lits/>
9. Машинное обучение для распознавания лиц [Электронный ресурс]: URL: <https://habr.com/ru/post/434088/>
10. Обзор алгоритмов распознавания лиц в машинном обучении [Электронный ресурс]: URL: <https://dataroot.ru/blog/algoritmy-raspoznavaniya-lits-v-mashinnom-obuchenii-obzor>
11. Применение нейронных сетей в задачах распознавания лиц [Электронный ресурс]: URL: <https://www.computerra.ru/340754/primenenie-neyronnyh-setej-v-zadachah-raspoznavaniya-lic/>

12. Распознавание лиц на Python с использованием библиотеки OpenCV [Электронный ресурс]: URL: <https://tproger.ru/translations/face-detection-python-opencv/>
13. Распознавание лиц на изображении с помощью Python и OpenCV [Электронный ресурс] :URL: <https://proglib.io/p/raspoznavanie-lica-na-izobrazhenii-s-pomoshchyu-python-i-opencv-2020-06-02>
14. Распознавание лиц на основе глубокого обучения [Электронный ресурс] URL: <https://neurohive.io/ru/tutorial/raspoznavanie-lic-na-osnove-glubokogo-obucheniya/>
15. Распознавание лиц с помощью алгоритмов машинного обучения [Электронный ресурс]: URL: <https://news.rambler.ru/other/39702036-raspoznavanie-lits-s-pomoschyu-algoritmov-mashinnogo-obucheniya/>
16. Распознавание лиц с помощью алгоритмов машинного обучения [Электронный ресурс]: URL: <https://nplus1.ru/news/2019/01/22/face-recognition-ml>
17. Система безопасности для образовательных учреждений PERCO [Электронный ресурс]: сайт компании. URL: <https://www.perco.ru/products/sistema-bezopasnosti-perco-s-20-shkola/>
18. Adrian, D. Facial Recognition Using Machine Learning. 2014. [Электронный ресурс]. Режим доступа: <https://weakdh.org/imperfect-forward-secrecy-ccs15.pdf>
19. Boni, S. Facial Recognition with Machine Learning: An Introduction / S. Boni, J. Bhatt, S. Bhat. 2015. [Электронный ресурс]. Режим доступа: <http://www.ijcaonline.org/research/volume130/number15/boni-2015-ijca-907170.pdf>
20. Chaturvedi, A. A Secure Wireless Deep Learning for Face Recognition / A. Chaturvedi, N. Srivastava, V. Shukla. 2015. [Электронный ресурс]. Режим доступа: <http://www.ijcaonline.org/research/volume126/number5/chaturvedi-2015-ijca906060.pdf>

21. Cozzens, M.J. Facial Recognition Technology: A Primer 2013. [Электронный ресурс]. Режим доступа: <https://books.google.ru/books?id=GbKyAAAAQBAJ&printsec=frontcover&dq=The+Mathematics+of+Encryption:+An+Elementary+Introduction&hl=ru&sa=X&ved=0ahUKEwif8-7t#v=onepage&q=The%20Mathematics%20of%20Encryption%3A%20An%20Elementary%20Introduction&f=false>
22. Face Recognition Using Machine Learning [Электронный ресурс]. Режим доступа: <https://www.math.ualberta.ca/~bowman/m422/m422.pdf>
23. Jenings, T.A. Face Detection and Recognition Using Machine Learning. 2012. [Электронный ресурс]. Режим доступа: [https://www.carroll.edu/library/thesisArchive/Anderson%20J\\_2012final.pdf](https://www.carroll.edu/library/thesisArchive/Anderson%20J_2012final.pdf)
24. Lai, D. Facial Recognition Algorithms Explained. / D. Lai, A. Khader. 2015. [Электронный ресурс]. Режим доступа: [https://www.researchgate.net/publication/280722113\\_Preventing\\_Man-In-The-Middle\\_Attack\\_in\\_Diffie-Hellman\\_Key\\_Exchange\\_Protocol\\_41](https://www.researchgate.net/publication/280722113_Preventing_Man-In-The-Middle_Attack_in_Diffie-Hellman_Key_Exchange_Protocol_41)
25. Lehtinen, S. Diffie-Hellman Key Exchange – A Survey of Face Recognition Techniques. 2011. [Электронный ресурс]. Режим доступа: <https://tampub.uta.fi/bitstream/handle/10024/83062/gradu05484.pdf?sequence=1>
26. Ruohonen, K. Machine Learning for Face Recognition: A Comprehensive Review. 2014. [Электронный ресурс]. Режим доступа: <http://math.tut.fi/~ruohonen/MC.pdf>