|  |  |
| --- | --- |
|  | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ **Информатика и системы управления**

КАФЕДРА **Компьютерные системы и сети (ИУ6)**

НАПРАВЛЕНИЕ ПОДГОТОВКИ **09.04.01 Информатика и вычислительная техника**

МАГИСТЕРСКАЯ ПРОГРАММА **09.04.01/07 Интеллектуальные системы анализа, обработки и интерпретации больших данных**

**Отчет**

|  |  |
| --- | --- |
| **по рубежному контролю №** | 2 |

**Название:** Проектирование системы определения дипфейков

**Дисциплина:** Искусство системного инжиниринга и менеджмента организаций

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Студент | ИУ6-12М |  |  | С.В. Астахов | |
|  | (Группа) |  | (Подпись, дата) | | (И.О. Фамилия) |
|  |  |  |  | |  |
| Преподаватель |  |  |  | | Д.В. Березкин |
|  |  |  | (Подпись, дата) | | (И.О. Фамилия) |

Москва, 2023

**СОДЕРЖАНИЕ**

[Введение 3](#_Toc3)

[1. Методы на основе сверточных и капсульных нейросетей 4](#_Toc4)

[2. Методы на основе временной согласованности 7](#_Toc5)

[2.1. Комбинация сверточной и рекуррентной нейронных сетей 7](#_Toc6)

[2.2. Улучшенные системы анализа временной согласованности 10](#_Toc7)

[3. Методы на основе визуальных артефактов 13](#_Toc8)

[Заключение 15](#_Toc9)

# Введение

В наше время социальные сети и мессенджеры стали неотъемлемой частью жизни многих людей. Однако, вместе с возможностью общения и обмена информацией, появилась и проблема фейковых новостей и фотографий.

Дипфейки - это фотографии или видео, созданные с помощью искусственного интеллекта, которые могут быть использованы для создания фальшивых новостей или дезинформации.

В связи с этим, возникает необходимость в разработке методов определения дипфейков, которые позволят бороться с распространением фальшивой информации. В данном реферате рассмотрены основные методы определения дипфейков.

# **Х. Определение зон интереса на изображении (/подготовка данных)**

Исходя из разработанного сценария обработки изображение, очевидно, что первым этапом на пути обработки изображения является выделение зон интереса. На основе характеристик изображения в этих зонах и будет производиться дальнейшее определение признаков дипфейка.

Для определения зон интереса подпрограмма предварительной обработки данных определяет положение 68 типовых точек лица с помощью библиотеки dlib [1].

Для обнаружение участка исходного изображения, в котором находится лицо, эта библиотека использует так называемую гистограмму направленных градиентов [2]. Для этого исходное изображение сначала переводится в черно-белый формат (рисунок 1).



Рисунок 1 — пример исходного изображения

Затем, для каждого небольшого участка изображения анализируется градиент (направление изменения) яркости в нем (рисунок 2).

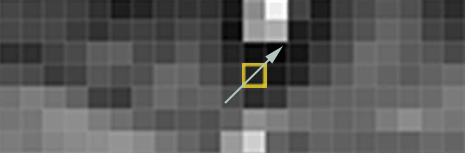


Рисунок 2 — вычисление градиента

Такой подход нивелирует влияние освещения и позволяет упростить вычисления при дальнейшем анализе. Пример изображения, обработанного данным алгоритмом приведен на рисунке 3.

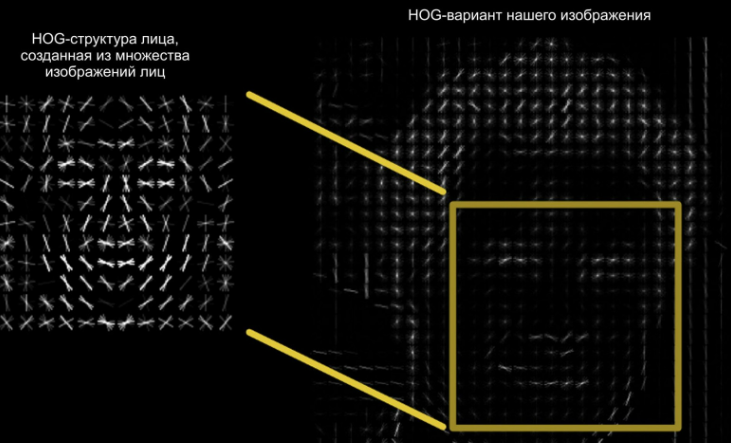


Рисунок 3 — представление изображения в виде набора градиентов

Для определения положения отдельных точек в зоне, где было обнаружено лицо, библиотека использует каскад регрессионных моделей.

Прежде всего, использование каскадов решает такую проблему, как влияние освещения и артефактов на точность определения точек. Изначальная дилемма состоит в том, что при определении признаков регрессорами, признаки могут быть искажены. Для борьбы с этим используется каскадная модель, которая итеративно уточняет форму лица и значения признаков в ее рамках.

Второй проблемой является тот факт, что алгоритм должен оценить форму, вектор высокой размерности, который наилучшим образом согласуется с данными изображения и моделью формы. Успешные алгоритмы решают эту проблему, предполагая, что предполагаемая форма должна лежать в линейном подпространстве, которое может быть обнаружено, например, путем нахождения основных компонентов форм [3].

Регрессоры представляют из себя деревья решений, используемых в сочетании с технологией градиентного бустинга на основе квадратичной функции потерь. Эти деревья решений анализируют разность значений интенсивности в парах пикселей на изображении, распределение расстояний между которыми определяется априорной функцией вероятности. Априорное распределение позволяет алгоритму бустинга эффективно исследовать большое количество релевантных признаков. Результатом является каскад регрессоров, которые могут локализовать лицевые метки на изображении лица в положении анфас.

Решение в каждом узле основано на пороговом значении разности значений интенсивности в паре пикселей. Это довольно простой подход, но он гораздо более эффективен, чем пороговое значение с одной интенсивностью, из-за его относительной нечувствительности к изменениям глобального освещения. К сожалению, недостатком использования разностей пикселей является то, что число потенциальных признаков является квадратичным по отношению к количеству пикселей в среднем изображении. Это затрудняет поиск хороших пар без поиска по очень большому их числу. Однако этот ограничивающий фактор может быть в некоторой степени ослаблен с учетом структуры данных изображения. Для этого при выборе расстояния между пикселями используется экспоненциальное распределение.

Стоит так же отметить, что данная модель умеет работать с обучающими примерами, где некоторая часть рассматриваемых меток не определена, задавая для них 0 вес при обучении.

Рассматриваемая модель состоит из большого числа “слабых” регрессоров, которые в свою очередь объединяютсяв несколько “сильных”. Зависимость точности модели от числа “сильных” регрессоров показана на рисунке 1.



Рисунок 4 — зависимость точности модели от числа регрессоров

Модель, используемая в библиотеке dlib обучалась на наборе данных HELEN, состоящем из 2000 размеченных изображений человеческих лиц из социальной сети Flickr [4].

Сравнение ошибок для используемых в бибилиотеке регрессоров в сопоставлении с другими возможными вариантами показано на рисунке 2. Ось абсцисс описывает число уровней каскада, ординат — значение ошибки. Кривая для используемых регрессоров обозначена зеленым цветом.

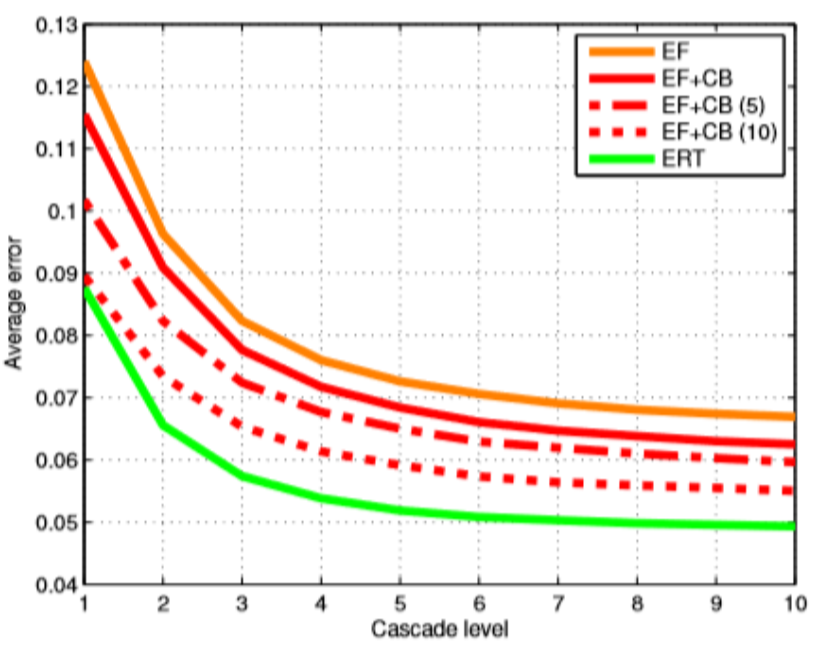


Рисунок 2 — уровни ошибок в зависимости от числа уровней каскада для различных типов регрессоров

На рисунке 3 показаны пары пикселей, отбираемые в качестве признаков при равномерном и экспоненциальном распределениях расстояний между ними соответственно. Из рисунка интуитивно понятно, что априорный выбор экспоненциальной функции повышает эффективность и точность модели, стимулируя ее выбирать более близкие точки для определения деталей лица.

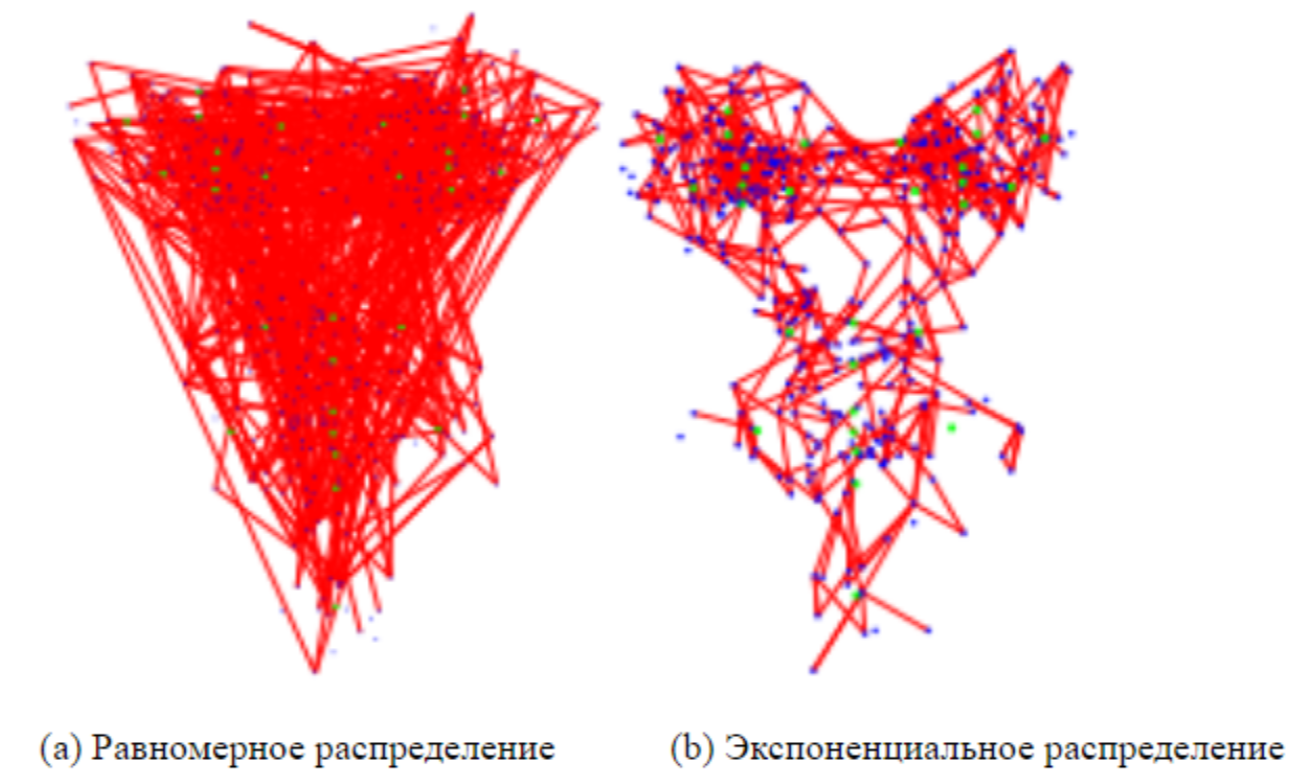


Рисунок 5 — выбор признаков при различных функциях распределения

Преимущество использования многоуровневой модели показано в таблице 1. Как видно, число уровней каскада влияет на значение ошибки куда сильнее, чем число слабых регрессоров на одном уровне каскада.

Таблица 1 — зависимость ошибки модели от числа уровней каскада и числа слабых регрессоров

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | Число уровней | Число слабых регр. | Число уровней | Число слабых регр. | Число уровней | Число слабых регр |
| Конф. модели | 1 | 500 | 1 | 5000 | 10 | 500 |
| Значение ошибки | 0.085 | | 0.074 | | 0.049 | |

Зависимость ошибки модели при 10 уровнях каскада для различного числа примеров показана в таблице 2.

Таблица 2 — зависимость ошибки от числа примеров

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Число примеров | 100 | 200 | 500 | 1000 | 2000 |
| Значение ошибки | 0.090 | 0.074 | 0.059 | 0.054 | 0.049 |

Зависимость значения ошибки от числа примеров для различного числа уровней каскада продемонстрирована на рисунке 4.

Пример реализации функции определения формы лица приведен в листинге 1.

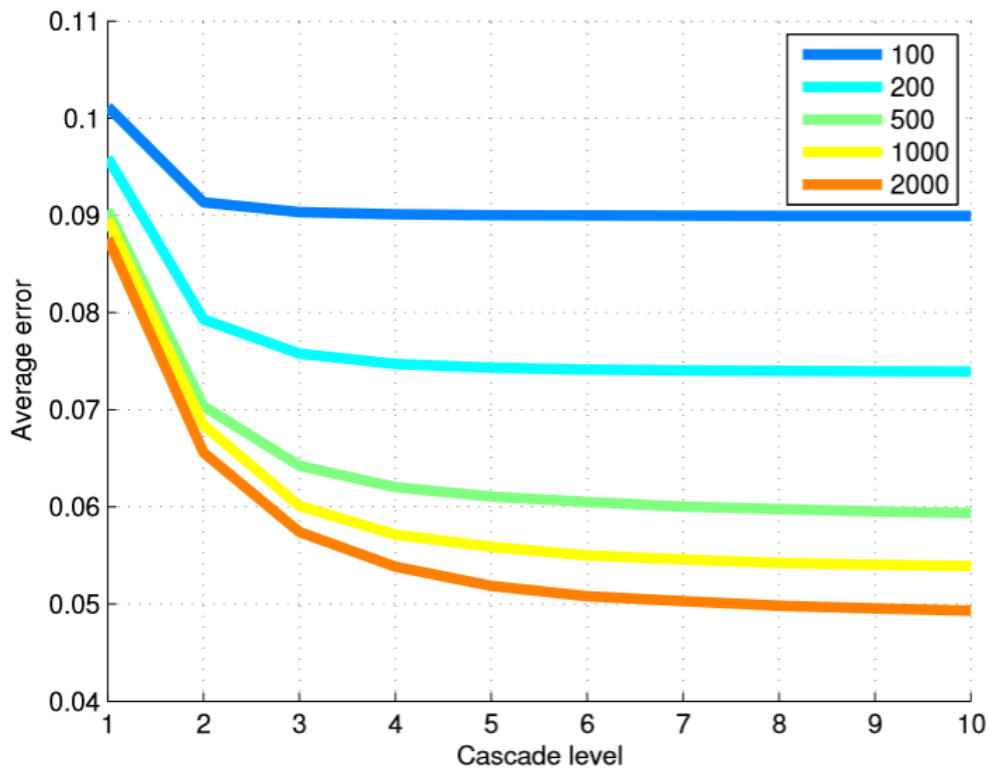


Рисунок 6 — зависимость значения ошибки от числа примеров для различного числа уровней каскада

Листинг 1 — функция определения формы лица

|  |
| --- |
| # импорт библиотек  import collections  import cv2 as cv  from imutils import face\_utils  import numpy as np  import imutils  import dlib  import numpy as np  # обучающий набор данных  PREDICTOR\_DATASET = './shape\_predictor\_68\_face\_landmarks.dat'  # определитель форм лица  PREDICTOR = dlib.shape\_predictor(PREDICTOR\_DATASET)  # функция определения формы лица  def detect\_faces(image):    # определитель зон, содержащих изображения лиц  detector = dlib.get\_frontal\_face\_detector()  predictor = PREDICTOR    # поиск лиц на ч/б изображении  gray = cv.cvtColor(image, cv.COLOR\_BGR2GRAY)  rects = detector(gray, 1)    # массив для координат точек  shapes = []  # для каждого изображения лица  for (i, rect) in enumerate(rects):    # определить 68 точек  shape = predictor(gray, rect)    # конвертировать в массив NumPy.array  shape = face\_utils.shape\_to\_np(shape)  # добавить форму к существующим  shapes.append(shape)  # вернуть точки для всех лиц  return shapes |

Данная функция является частью модуля “faceparts”, разработанного в рамках данной работы и используется затем для предобработки обучающего набора изображений. Программный код функции, порождающей 1 строку обучающего набора приведен в листинге 2.

Листинг 2 — функция предобработки данных

|  |
| --- |
| # импорт библиотек  import pandas as pd  import faceparts as fp  import cv2 as cv  import numpy as np  import cv2 as cv  # функция получения строки датафрейма  # параметры: имя файла, флаг фейка, коориднаты 68 точек, кадр, масштаб  # ROI  def generate\_dataframe(filename, is\_fake, shape, frame, frame\_scale\_percent):  # перевод в градации серого  gray\_frame = cv.cvtColor(frame.copy(), cv.COLOR\_BGR2GRAY)  data = {}  # расчет радиуса лица  r = pow(  pow(shape[fp.LEFT\_EAR\_POINT][0] - shape[fp.RIGHT\_EAR\_POINT][0], 2) +  pow(shape[fp.LEFT\_EAR\_POINT][1] - shape[fp.RIGHT\_EAR\_POINT][1], 2),  0.5  )  # расчет размера ROI  rect\_size = int((r \* frame\_scale\_percent) / 100.0)    # заполнение вспомогательных полей датафрейма  data['filename'] = [filename]  data['fake'] = [is\_fake]  # для каждой зоны на лице  for zone in fp.FACIAL\_LANDMARKS\_IDXS.keys():  # для каждой точки в зоне  for point\_id in range(fp.FACIAL\_LANDMARKS\_IDXS[zone][0], fp.FACIAL\_LANDMARKS\_IDXS[zone][1]):  # получение координат точек и границ ROI  point = shape[point\_id]  point\_x = point[0]  point\_y = point[1]  x1, y1, x2, y2 = \  int(point\_x - int(rect\_size/2)), \  int(point\_y - int(rect\_size/2)), \  int(point\_x + int(rect\_size/2)), \  int(point\_y + int(rect\_size/2))    # запись бинарных данных  data['pt\_' + str(point\_id) + '\_' + zone +'\_raw'] = [gray\_frame[y1:y2, x1:x2].ravel()]  # информация обо всей зоне лица  point\_x = int((shape[fp.LEFT\_EAR\_POINT][0] + shape[fp.RIGHT\_EAR\_POINT][0])/2)  point\_y = int((shape[fp.LEFT\_EAR\_POINT][1] + shape[fp.RIGHT\_EAR\_POINT][1])/2)  x1, y1, x2, y2 = \  int(point\_x - int(r)), \  int(point\_y - int(r)), \  int(point\_x + int(r)), \  int(point\_y + int(r))  data['overall\_face\_raw'] = [gray\_frame[y1:y2, x1:x2].ravel()]  # перевод в формат pandas.Dataframe  df = pd.DataFrame(data=data)  return (df.copy()) |

# **Заключение**

Проблема фейковых новостей, видео и фотографий становится все более актуальной в нашем информационном обществе. Дипфейки могут нанести серьезный ущерб как отдельным людям, так и обществу в целом.

Однако, существуют различные методы определения дипфейков, которые позволяют бороться с распространением фальшивой информации.

В данном реферате были рассмотрены основные методы определения дипфейков, такие как анализ пикселей, анализ движения, анализ голоса и другие.

Каждый из этих методов имеет свои преимущества и недостатки, и их эффективность зависит от конкретной ситуации. В целом, разработка и совершенствование методов определения дипфейков является важной задачей для обеспечения достоверности информации в нашем информационном обществе.

**СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ**

1. <http://dlib.net/train_shape_predictor.py.html>
2. https://habr.com/ru/articles/306568/
3. <https://www.cv-foundation.org/openaccess/content_cvpr_2014/papers/Kazemi_One_Millisecond_Face_2014_CVPR_paper.pdf>
4. http://www.ifp.illinois.edu/~vuongle2/helen/