

## ИНФОРМАТИКА И ИНФОРМАЦИОННЫЕ ПРОЦЕССЫ/INFORMATICS AND INFORMATION PROCESSES

DOI: <https://doi.org/10.60797/IRJ.2025.156.52>

### РАЗРАБОТКА СИСТЕМЫ РАСПОЗНАВАНИЯ ЭМОЦИЙ ПО ЛИЦЕВЫМ ВЫРАЖЕНИЯМ НА ОСНОВЕ МАШИННОГО ОБУЧЕНИЯ

Научная статья

Шурдилов И.С.<sup>1</sup>, Нуриев М.Г.<sup>2,\*</sup>, Лаптева М.Г.<sup>3</sup>, Толмачева А.В.<sup>4</sup>

<sup>2</sup> ORCID : 0009-0003-0741-1734;

<sup>1, 2, 3, 4</sup> Казанский национальный исследовательский технический университет им. А.Н. Туполева – КАИ, Казань, Российская Федерация

\* Корреспондирующий автор (mgnuriev[at]kai.ru)

#### Аннотация

В статье представлены результаты разработки и тестирования системы машинного обучения для распознавания эмоций по лицевым выражениям. Создан веб-сервис, обеспечивающий три режима работы: анализ эмоций на статических изображениях, видеозаписях и в реальном времени с использованием веб-камеры. Для обучения модели использован размеченный датасет базовых эмоций, включающий несколько классов выражений лица. В ходе работы проведена оценка эффективности системы с применением различных тестовых наборов данных и метрик качества классификации. Анализ экспериментальных результатов подтвердил надежность предложенного подхода. Разработанная система может быть применена в различных областях, включая компьютерное зрение, автоматическое распознавание эмоций, психологические исследования, разработку интерактивных пользовательских интерфейсов, а также в системах дистанционного обучения и анализа поведения пользователей.

**Ключевые слова:** распознавание эмоций, машинное обучение, лицевые выражения, обработка изображений, глубокое обучение.

### DEVELOPMENT OF A SYSTEM FOR RECOGNISING EMOTIONS FROM FACIAL EXPRESSIONS BASED ON MACHINE LEARNING

Research article

Shurdilov I.S.<sup>1</sup>, Nuriev M.G.<sup>2,\*</sup>, Lapteva M.G.<sup>3</sup>, Tolmacheva A.V.<sup>4</sup>

<sup>2</sup> ORCID : 0009-0003-0741-1734;

<sup>1, 2, 3, 4</sup> Kazan National Research Technical University named after A.N. Tupolev – KAI, Kazan, Russian Federation

\* Corresponding author (mgnuriev[at]kai.ru)

#### Abstract

The article presents the results of development and testing of a machine learning system for recognising emotions from facial expressions. A web service has been created that provides three modes of operation: emotion analysis on static images, video recordings and in real time using a webcam. A marked-up dataset of basic emotions, including several classes of facial expressions was used to train the model. The performance of the system was evaluated using different test datasets and classification quality metrics. The analysis of experimental results confirmed the reliability of the proposed approach. The developed system can be applied in various fields, including computer vision, automatic emotion recognition, psychological research, development of interactive user interfaces, as well as in distance learning systems and user behaviour analysis.

**Keywords:** emotion recognition, machine learning, facial expressions, image processing, deep learning.

#### Введение

Распознавание эмоций сложная задача, объединяющая знания из разных областей. Новые технологии и алгоритмы машинного обучения открывают новые возможности для создания более точных систем распознавания эмоций [1], [2].

Важность разработки систем распознавания эмоций на основе машинного обучения становится все более очевидной. С появлением новых форм общения, таких как голосовые ассистенты, чат-боты, виртуальные ассистенты, интерактивные игры и т.п., возникает необходимость в создании систем, способных общаться с людьми более естественным образом. Распознавание эмоций может играть важную роль в улучшении опыта взаимодействия с такими системами.

Системы распознавания эмоций могут значительно влиять на всевозможные сферы: в маркетинге и рекламе помогут в анализе реакций потребителей на продукты и услуги [3], [4]. В игровой индустрии — в создании более аутентичных и естественных персонажей, способных реагировать соответствующим образом на различные ситуации. Например, если персонаж должен проявлять гнев или страх в ответ на какое-то действие, система распознавания эмоций может помочь сделать реакцию убедительнее и реалистичнее. Также системы помогут в создании интерактивных игровых сценариев, подстраивающихся под игрока. Кроме того, они могут быть важным инструментом в научных изысканиях в области психологии, анализа эмоциональных реакций и поведения людей. В медицине помочь будет заключаться в диагностике психических расстройств и улучшении качества обслуживания пациентов, значительном сокращении времени диагностики и терапии, улучшая тем самым качество лечения и повышая удовлетворенность клиентов. В образовании — в создании наиболее действенных методик обучения, а также в повышении эффективности обучения за счёт адаптации учебных программ к потребностям и эмоциональному состоянию учеников и студентов.

Помимо этого, системы распознавания эмоций имеют важное значение в различных областях, связанных с виртуальной и дополненной реальностью. Они могут анализировать эмоциональную реакцию пользователей на виртуальные объекты и сцены, а также создавать более реалистичные симуляции и тренировки. Кроме того, они могут использоваться для анализа эмоциональной составляющей во взаимодействии персонала и клиентов, в процессе набора персонала, с целью создания эффективных команд, а также улучшать удовлетворенность и мотивацию персонала. Развитие систем распознавания эмоций может привести к созданию новых инновационных продуктов и сервисов, что делает тему исследования актуальной для различных областей науки и технологий.

Целью данной работы является разработка системы распознавания эмоций на основе машинного обучения и ее апробация на реальных данных. Для достижения этой цели были поставлены следующие задачи:

- подготовка данных;
- обучение модели;
- реализация системы;
- тестирование и оценка разработанной системы.

Выполнение этих задач позволит создать систему распознавания эмоций на основе машинного обучения, способную производить анализ и выявление человеческих эмоций.

### Подготовка данных

Для решения задачи распознавания эмоций на лицах был выбран набор данных FER2013 [5]. Он представляет собой набор изображений лиц людей с различными эмоциями: Angry (Злость), Disgust (Отвращение), Fear (Страх), Happy (Счастье), Sad (Грусть), Surprise (Удивление), Neutral (Нейтральность). Для каждого изображения указывается эмоция, которую оно передает. В наборе данных FER2013 содержится около 35 тысяч изображений, разбитых на тренировочный, тестовый и валидационный наборы. Тренировочный набор содержит более 28 тысяч изображений, тестовый набор около 3,7 тысяч, а валидационный набор около 3,6 тысяч.

Одним из преимуществ набора данных FER2013 является его доступность — скачать его может любой желающий с сайта Kaggle. Кроме того, набор данных FER2013 является стандартным набором данных для задачи распознавания эмоций на лицах, что упрощает сравнение результатов с другими исследованиями.

Однако, набор данных FER2013 имеет и некоторые недостатки. Например, изображения из набора имеют низкое разрешение ( $48 \times 48$  пикселей), что может распознавание эмоций. Кроме того, как видно на рисунке 1, распределение изображений по классам не является равномерным: некоторые классы содержат гораздо меньше изображений, чем другие. Это может привести к проблемам с обучением модели и снижению ее точности.

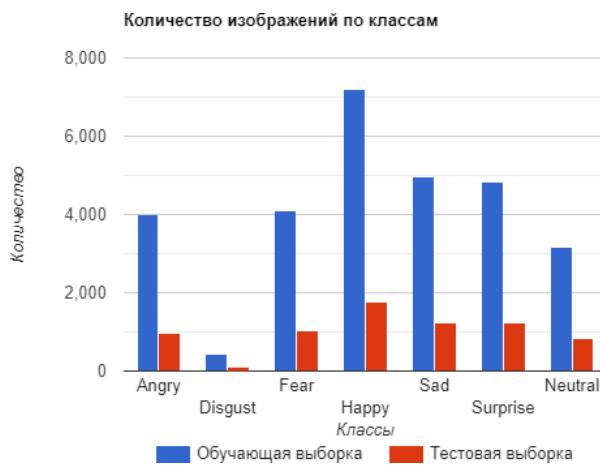


Рисунок 1 - Распределение изображений по классам набора данных FER2013

DOI: <https://doi.org/10.60797/IRJ.2025.156.52.1>

Набор данных FER2013 является популярным и доступным выбором для подобных проектов, обеспечивая достаточное количество изображений для обучения, тестирования и валидации модели. Однако при его использовании необходимо учитывать ряд ограничений, влияющих на качество обучения. В частности, точность предсказания с использованием архитектуры CNN [6] составляет 62,44%, что может оказаться недостаточным для некоторых приложений. В связи с этим целесообразно рассмотреть альтернативные наборы данных или их комбинацию для повышения точности и обобщающей способности модели.

Предобработка данных — не менее важная ступень проекта машинного обучения. Первым шагом в предобработке является загрузка данных (листинг 1) из набора FER2013. Набор данных содержит три файла: train.csv, test.csv и validation.csv, содержащих информацию об изображениях и соответствующих им эмоциях. Эти файлы содержат эмоцию в виде числа от 0 до 6 и набор пикселей, составляющих изображение.

Изображения в наборе данных имеют низкое разрешение  $48 \times 48$  пикселей. Примеры изображений показаны на рис. 2.



Рисунок 2 - Примеры изображений из набора данных:

а – Surprise; б – Sad; в – Neutral; г – Happy; е – Fear; ё – Disgust; ж – Angry

DOI: <https://doi.org/10.60797/IRJ.2025.156.52.2>

Для улучшения качества изображений и упрощения их анализа, были применены следующие преобразования:

1. Конвертация в черно-белый формат для уменьшения размерности данных и ускорения обучения модели.

2. Нормализованы для того, чтобы значения пикселей находились в диапазоне от 0 до 1.

Таким образом, мы ускорим процесс обучения и улучшим качество модели.

Целевая переменная в наборе данных представлена в виде текстовых меток эмоций: Angry, Disgust, Fear, Happy, Sad, Surprise и Neutral. Учитывая, что мы решаем задачу классификации, для обучения модели необходимо преобразовать эти метки в числовую формат. Есть несколько методов кодирования, наиболее популярные среди них – Label encoding [7] и One-Hot [8]. При использовании метода Label encoding создает впечатление, что происходит ранжирование значений. Чтобы избежать этого, было использовано кодирование One-Hot, которое представляет каждую метку в виде вектора с единицей в соответствующей позиции и нулями в остальных позициях. Например, метка "Happy" будет закодирована в вектор [0, 0, 0, 1, 0, 0, 0].

Для обучения и оценки модели было необходимо разделить данные на тренировочный, тестовый и валидационный наборы. Всего в наборе данных 35887 изображений, разбитых на 3 части: обучающая выборка 28709 изображений, тестовая выборка 3589 изображений и валидационная выборка 3589 изображений. Подробное распределение изображений по классам и категориям можно увидеть в таблице 1.

Таблица 1 - Количество изображений каждого класса по наборам

DOI: <https://doi.org/10.60797/IRJ.2025.156.52.3>

| Эмоция   | Обучающая выборка | Тестовая выборка | Валидационная выборка |
|----------|-------------------|------------------|-----------------------|
| Angry    | 3 995             | 467              | 491                   |
| Disgust  | 436               | 56               | 66                    |
| Fear     | 4 097             | 496              | 528                   |
| Happy    | 7 215             | 895              | 879                   |
| Sad      | 4 030             | 653              | 594                   |
| Surprise | 3 171             | 415              | 416                   |
| Neutral  | 4 965             | 607              | 626                   |

Для предобработки изображений и преобразования данных в формат, который может быть использован в модели машинного обучения были использованы:

1. Генератор ImageDataGenerator для аугментации данных для увеличения количества обучающих изображений и большей устойчивости к условиям освещения и позам лиц.

2. Генератор flow\_from\_directory для загрузки изображений и их предобработки в пакеты.

3. Генератор flow для преобразования данных в формат для машинного обучения.

В результате предобработки (см. листинг 1) данных были получены тренировочные, тестовые и валидационные наборы данных, которые могут быть использованы для обучения модели машинного обучения на задаче распознавания эмоций на лицах.

Листинг 1. Код файла model\_creation.py:

```
train_datagen = ImageDataGenerator(rescale=1.0 / 255, rotation_range=10, width_shift_range=0.1,
height_shift_range=0.1, shear_range=0.1, zoom_range=0.1, horizontal_flip=True, fill_mode="nearest")
test_datagen = ImageDataGenerator(rescale=1.0 / 255)
train_generator = train_datagen.flow_from_directory(train_path, target_size=(48, 48), batch_size=64,
shuffle=True,
_mode="grayscale", _mode="categorical")
test_generator=test_datagen.flow_from_directory(test_path, target_size=(48, 48), batch_size=64,
shuffle=True,
```

## Разработка модели машинного обучения

Выбор правильной архитектуры модели является одним из ключевых факторов, влияющих на качество распознавания эмоций. На данный момент существует множество различных архитектур моделей, пригодных для обучения системы распознавания эмоций.

Одна из наиболее распространенных архитектур на основе машинного обучения — это сверточные нейронные сети (Convolutional Neural Networks или CNN). CNN используются для обработки изображений и могут извлекать важные признаки из входных данных, которые затем могут быть использованы для распознавания эмоций.

Кроме того, существуют рекуррентные нейронные сети (Recurrent Neural Networks или RNN [9]). Их используют для обработки последовательностей данных, таких как звуковые сигналы или тексты. RNN позволяют моделировать долгосрочные зависимости между данными, что может быть полезно для распознавания эмоций.

При выборе архитектуры модели необходимо учитывать размер и сложность набора данных, а также количество доступных вычислительных ресурсов для обучения модели. Как правило, для решения задач классификации изображений, таких как распознавание эмоций на лицах, применяются сверточные нейронные сети.

CNN — это класс нейронных сетей, который специально разработан для обработки изображений. Они автоматически извлекают признаки из изображений, что позволяет модели эффективно классифицировать объекты. CNN состоят из нескольких слоев, включая сверточные слои, слои подвыборки (pooling layers) и полносвязные слои (fully connected layers).

CNN имеет доказанную эффективность в решении задач классификации изображений, включая распознавание эмоций на лицах. В данной работе была использована заранее подготовленная архитектура CNN, обученная на большом наборе изображений. Это ускорило процесс обучения и повысило качество модели.

## Подбор параметров обучения

После выбора архитектуры модели, необходимо подобрать оптимальные параметры для обучения. Параметры, включают в себя функцию потерь, скорость обучения, оптимизатор, регуляризацию, размера батча и другие гиперпараметры модели [10], [11].

Функция потерь отражает степень ошибки модели во время обучения и является важным критерием для оценки ее производительности. Скорость обучения определяет, как быстро модель адаптируется к данным и как быстро может быть настроена. Оптимизатор отвечает за процесс оптимизации параметров модели, регуляризация помогает снизить переобучение, а размер батча определяет, сколько примеров будет использоваться в каждой эпохе обучения. К другим гиперпараметрам относят: количество слоев, количество нейронов на каждом слое и т.д.

Для определения оптимальных параметров проводят эксперименты на небольших подмножествах данных, изучают кривые обучения и выбирают параметры, которые демонстрируют наилучший результат на валидационных данных. Этот процесс может занять много времени и требует тщательного анализа результатов экспериментов.

В нашем случае были выявлены следующие параметры (листинг 2): скорость обучения (learning\_rate), количество слоев и регуляризация (regularizers.l2). Например, слишком высокая скорость обучения привела к быстрому переобучению, а слишком низкая затянула процесс обучения. Маленькое количество слоев было совершенно недостаточно для изучения сложных паттернов в данных, а слишком большое привело к переобучению. Регуляризация помогла снизить переобучение, но чрезмерно высокое значение регуляризации может существенно ухудшить результаты.

Важно помнить, что оптимальные параметры могут отличаться для разных моделей и датасетов, поэтому подбор параметров обучения необходимо проводить тщательно.

Листинг 2. Код файла model\_creation.py:

```
model = Sequential()
...
model.add(
    Conv2D(
        128,
        kernel_size=(3, 3), activation="relu",
        padding="same", kernel_regularizer=regularizers.l2(0.01),
    )
)
model.add(
    Conv2D(
        256,
        kernel_size=(3, 3), activation="relu",
        kernel_regularizer=regularizers.l2(0.01),
    )
)
...
model.compile(
    optimizer=Adam(learning_rate=0.0001, decay=1e-6), loss="categorical_crossentropy",
    metrics=["accuracy"],
)
```

Обучение модели машинного обучения включает в себя подготовку данных, определение архитектуры модели, подбор параметров обучения и обучение модели на данных.

Для обучения был использован фреймворк TensorFlow [12]. Модель прошла 60 эпох с размером батча 64. В качестве функции потерь была использована категориальная перекрестная энтропия (categorical crossentropy), а в качестве оптимизатора Adam (Adaptive Moment Estimation) — это метод стохастической оптимизации, который адаптивно изменяет скорость обучения во время процесса. Adam объединяет преимущества двух других популярных методов оптимизации: Adagrad [13] и RMSprop [14].

В процессе обучения модели, оптимизатор Adam [15] принимает на вход градиенты, вычисленные в каждой точке функции потерь, и изменяет веса модели, чтобы уменьшить значение функции потерь. Алгоритм Adam адаптируется к особенностям локальных градиентов функции, используя две ключевые меры: среднее значение градиента и среднеквадратичное отклонение градиента, что позволяет ему корректировать скорость обучения соответствующим образом. Начальный шаг обучения был равен 0,0001.

Во время обучения модели использовались техники регуляризации, такие как Dropout, которые помогли предотвратить переобучение модели (листинг 3).

1. Сверточный слой (Conv2D) с 32 фильтрами, размером ядра (3;3) и функцией активации ReLU, с добавлением нулевого отступа к входным данным. Входной размер изображения (48;48;1).

2. Сверточный слой (Conv2D) с 64 фильтрами, размером ядра (3;3) и функцией активации ReLU, с добавлением нулевого отступа к входным данным.

3. Слой нормализации по батчу (BatchNormalization).

4. Слой объединения с максимальным значением (MaxPooling2D) с размером пула (2,2).

5. Слой регуляризации Dropout с коэффициентом отсечения 0,25.

6. Сверточный слой (Conv2D) с 128 фильтрами, размером ядра (3;3) и функцией активации ReLU, используется коэффициент регуляризации 0,01, с добавлением нулевого отступа к входным данным.

7. Сверточный слой (Conv2D) с 256 фильтрами, размером ядра (3;3) и функцией активации ReLU, используется коэффициент регуляризации 0,1, с добавлением нулевого отступа к входным данным.

8. Слой нормализации по батчу (BatchNormalization).

9. Слой объединения с максимальным значением (MaxPooling2D) с размером пула (2;2).

10. Слой регуляризации Dropout с коэффициентом отсечения 0,25.

11. Сверточный слой (Conv2D) с 512 фильтрами, размером ядра (3;3) и функцией активации ReLU, используется коэффициент регуляризации 0,01, с добавлением нулевого отступа к входным данным.

12. Слой нормализации по батчу (BatchNormalization).

13. Слой объединения с максимальным значением (MaxPooling2D) с размером пула (2;2).

14. Слой регуляризации Dropout с коэффициентом отсечения 0,25.

15. Растигивающий слой (Flatten), который преобразует выходные данные из сверточных слоев в одномерный массив.

16. Полносвязный слой (Dense) с 1024 нейронами и функцией активации ReLU.

17. Слой регуляризации Dropout с коэффициентом отсечения 0,5.

Полносвязный слой (Dense) с 7 нейронами (по количеству классов) и функцией активации softmax.

Листинг 3. Код файла model\_creation.py:

```
model.add(Conv2D(32,kernel_size=(3, 3),padding="same",activation="relu",input_shape=(48, 48, 1)))
model.add(Conv2D(64, kernel_size=(3, 3), activation="relu", padding="same"))
model.add(BatchNormalization())
model.add(MaxPooling2D(2, 2))
model.add(Dropout(0.25))
model.add(Conv2D(128, kernel_size=(3, 3), activation="relu", kernel_regularizer=regularizers.l2(0.01)))
model.add(Conv2D(256, kernel_size=(3, 3), activation="relu", kernel_regularizer=regularizers.l2(0.01)))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))
model.add(Conv2D(512, kernel_size=(3, 3), activation="relu", padding="same", activation="relu",
padding="same", kernel_regularizer=regularizers.l2(0.01),))
kernel_regularizer=regularizers.l2(0.01))
model.add(Conv2D(512, kernel_size=(3, 3), activation="relu", padding="same",
kernel_regularizer=regularizers.l2(0.01),))
model.add(BatchNormalization())
model.add(MaxPooling2D(pool_size=(2,2)))
model.add(Dropout(0.25))
model.add(Flatten())
model.add(Dense(1024,activation="relu"))
model.add(Dropout(0.5))
model.add(Dense(7,activation="softmax"))
```

Модель, содержащая вышеописанные слои, отображена на рисунке 3.

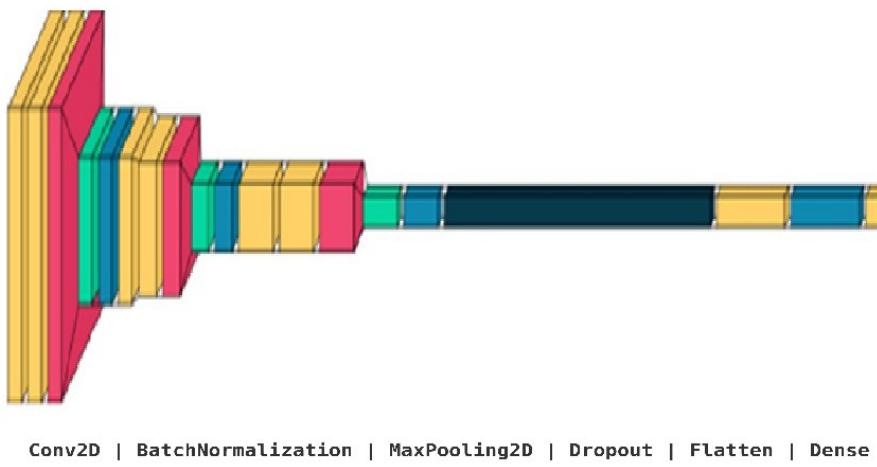


Рисунок 3 - Многоуровневое представление архитектуры модели  
DOI: <https://doi.org/10.60797/IRJ.2025.156.52.4>

После обучения на тренировочном наборе данных модель была протестирована на тестовом наборе для оценки её точности. В результате тестирования точность модели составила 69,4% [15]. Данный показатель можно улучшить за счёт оптимизации гиперпараметров и применения более сложных архитектур нейросетей.

### Тестирование и оценка эффективности системы

Метрики являются важным инструментом для оценки эффективности системы. В данной работе используются следующие метрики:

- Classification Report;
- Confusion Matrix;
- Per Class Accuracy: (точность по каждому классу);
- Training Accuracy;
- Validation Accuracy;
- Training Loss;
- Validation Loss;
- ROC Curve.

Пояснения к метрикам:

Classification Report — отчет о классификации, который показывает точность, полноту, F1-score и поддержку для каждого класса. Он предоставляет информацию о том, как хорошо модель предсказывает каждый класс. Он содержит следующие показатели:

1. Точность (precision) измеряет, сколько объектов было правильно классифицировано в определенную категорию по сравнению с общим числом объектов, которые модель предсказала как принадлежащие этой категории.

2. Полнота (recall) показывает, сколько объектов данной категории было корректно идентифицировано моделью относительно общего числа объектов данной категории.

3. F1-score является гармоническим средним между точностью и полнотой, и оно показывает общую эффективность модели в определении каждого класса.

4. Поддержка (support) представляет собой количество экземпляров в каждом классе в тестовом наборе.

Каждая метрика дает разные показатели о качестве классификации. Например, высокая точность и низкая полнота могут означать, что модель часто ошибается в предсказании некоторых классов и пропускает их. И наоборот, высокая полнота и низкая точность могут означать, что модель предсказывает некоторые классы с высокой уверенностью, но также классифицирует их ошибочно в другие классы. Поэтому важно анализировать все метрики вместе, чтобы понимать, как модель работает на каждом классе.

Confusion Matrix — это таблица, которая показывает количество верно и неверно определённых примеров для каждого класса в задаче многовариантной классификации.

В частности, для каждого класса из списка истинных меток (true labels) мы можем узнать, сколько примеров было верно отнесены к данному классу, а сколько было ошибочно отнесены к другим классам. Это же можно сделать и для каждого класса из списка предсказанных меток (predicted labels), то есть мы можем узнать, сколько примеров алгоритм отнес к каждому классу, и сколько ошибочных прогнозов он сделал.

Таким образом, Confusion Matrix состоит из  $n$  строк и  $n$  столбцов, где  $n$  количество классов в задаче. Каждая строка соответствует одному из классов, а каждый столбец прогнозам модели. Элемент в  $i$ -й строке и  $j$ -ом столбце матрицы ( $i;j$ ) показывает количество примеров, которые на самом деле принадлежат  $i$ -му классу, но были предсказаны моделью как  $j$ -й класс.

Кроме того, по элементам матрицы можно вычислить множество различных метрик, таких как точность (accuracy), четкость (precision), полнота (recall) и F1-мера (F1-score) для каждого класса, а также для всей модели в целом.

Per Class Accuracy (точность по каждому классу) показывает процент правильных прогнозов модели для каждого класса. Эта метрика вычисляется путем деления диагональных элементов матрицы ошибок (Confusion Matrix) на сумму элементов в соответствующих строках. Таким образом, мы можем получить точность для каждого класса,

узнать, как хорошо модель справляется с определением каждой категории. По этой метрике можно оценить, какие классы модель распознает лучше всего, а какие хуже всего. Результаты метрики могут быть представлены в виде графика, где на оси x отображаются классы, а на оси y процент правильных ответов для каждого класса.

Training Accuracy — это метрика, которая показывает, насколько хорошо модель работает с обучающими данными и выражает процент правильных ответов модели во время обучения.

Validation Accuracy — это метрика, которая показывает, насколько хорошо модель работает на данных проверки, то есть на данных, которые модель не видела во время обучения. Иначе говоря, это процент верных ответов для проверочных данных.

Цель состоит в том, чтобы обучить модель, которая показывает высокую точность как на данных обучения, так и на данных проверки. Однако, если модель показывает очень высокую точность на данных обучения, но низкую точность на данных проверки, это может означать, что модель переобучена (overfitting), то есть она выучила особенности данных обучения, но не смогла обобщить новые данные. В этом случае, необходимо пересмотреть процесс обучения модели и отрегулировать ее, чтобы снизить эффект переобучения.

Training Loss — это потери, которые происходят во время анализа обучающих данных, то есть это ошибки, которые происходят между предсказанными значениями и истинными значениями на обучающих данных.

Validation Loss — это потери, которые происходят во время обучения на проверочных данных, то есть это ошибки, происходящие между предсказанными значениями и истинными значениями на проверочных данных.

Чем ниже значение потерь, тем лучше модель обучается. Однако, если модель будет слишком сложной и будет хорошо подгоняться под обучающие данные, то она может переобучиться, что приведет к плохой обобщающей способности на новые данные. Поэтому нужно контролировать значения обеих метрик и выбирать модель с наименьшим значением потерь на проверочных данных, но при этом, сохраняя приемлемую точность на обучающих данных.

ROC Curve (Receiver Operating Characteristic) [16] — это графический инструмент для оценки качества бинарного классификатора. Он позволяет визуализировать отношение между чувствительностью (True Positive Rate TPR) и специфичностью (False Positive Rate FPR) классификатора при различных пороговых значениях.

На графике, FPR отображается по оси X, а TPR по оси Y. Каждая точка на ROC-кривой соответствует определенному пороговому значению. Чем ближе ROC-кривая к верхнему левому углу графика, тем лучше производительность классификатора.

Кроме того, AUC (Area Under the Curve) показывает площадь под кривой ROC, что является мерой качества классификатора. Значение AUC может изменяться от 0 до 1, где 1 означает идеальный классификатор, а 0,5 случайный классификатор.

### Проведение тестирования

После обучения модели и подбора оптимальных параметров, необходимо провести тестирование, чтобы оценить ее реальную эффективность. В данном разделе мы подробнее рассмотрим процесс проведения тестирования и анализа результатов.

Для начала, мы подготовили тестовую выборку, которая была отложена во время обучения модели. Тестовая выборка представляла собой набор данных, который не участвовал в обучении модели и была предназначена для проверки ее работоспособности в реальных условиях.

Затем, мы использовали выбранные метрики для оценки результатов работы модели на тестовых данных. Мы сравнивали полученные результаты с целевыми значениями метрик, чтобы оценить эффективность модели. Если модель давала плохие результаты, мы анализировали ошибки и вносili соответствующие изменения в модель.

Одной из основных метрик, которую мы использовали для оценки эффективности модели, была точность (accuracy). Мы также использовали confusion matrix, classification report и ROC-кривую для более детальной оценки результатов работы модели.

Тестирование является важным этапом разработки модели машинного обучения, так как оно позволяет оценить ее реальную эффективность и точность в конкретных условиях. Результаты тестирования могут использоваться для улучшения модели и ее дальнейшей оптимизации. Кроме того, тестирование позволяет убедиться в корректности работы модели и ее способности обобщать новые данные.

В ходе тестирования мы выявляли слабые места модели, настроили гиперпараметры для улучшения результатов оценки эффективности работы модели. Мы также проверили ее работоспособность на различных наборах данных, чтобы оценить универсальность и применимость в различных условиях.

После завершения обучения модели и проведения тестирования, мы провели анализ полученных результатов. В данном разделе подробно рассмотрим результаты работы модели, включая значения метрик: classification report, confusion matrix и ROC-кривую [17], [18].

После 60-й эпохи обучения модель достигла значения функции потерь (loss) 0,8282 и точности (accuracy) 0,7432. Однако значение функции потерь на валидационной выборке (val\_loss) не показало улучшения по сравнению с предыдущей эпохой и составило 1,0312, а точность на валидационной выборке (val\_accuracy) составила 0,6920. Это может свидетельствовать о начале переобучения модели, что требует дополнительного анализа и возможной корректировки параметров обучения.

Полученный classification report показывает метрики precision, recall и f1-score для каждого класса, а также средневзвешенные значения для всех классов.

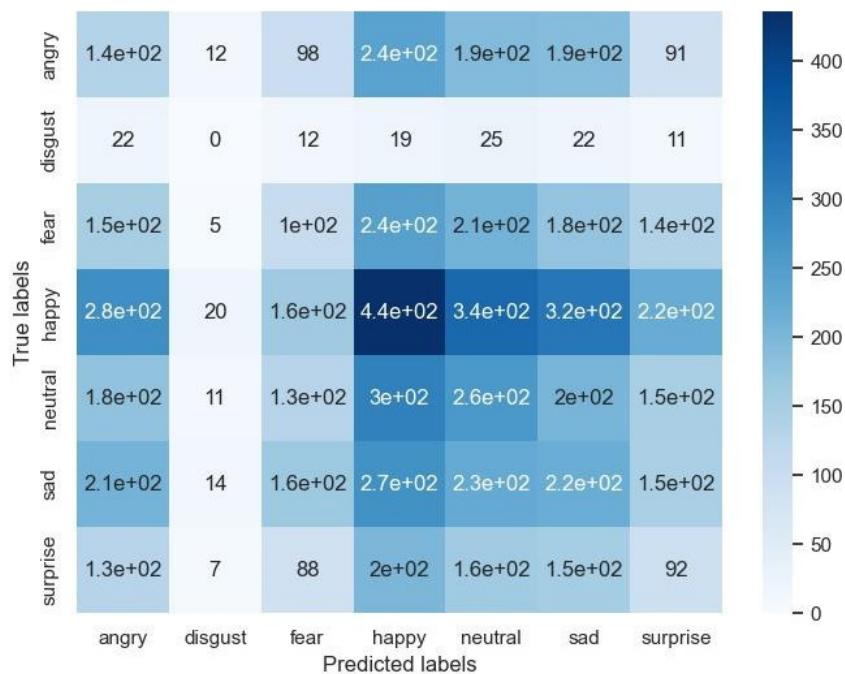
Результаты показывают, что модель имеет низкую чёткость (precision) и отзыв (recall) для всех классов, с наиболее высокими значениями для классов happy, neutral и sad. Средневзвешенное значение f1-score для всех классов составляет 0,17, что говорит о низкой общей эффективности модели.

Матрица ошибок (рис. 4) (confusion matrix) предоставляет информацию о классификации объектов моделью. В данном случае, матрица имеет размерность  $7 \times 7$  и показывает количество объектов каждого класса, которые были правильно и неправильно классифицированы. Наблюдается значительное количество ошибок классификации для всех классов, особенно для классов angry, fear и happy.

Таблица 2 - Текст файла learning.log

DOI: <https://doi.org/10.60797/IRJ.2025.156.52.5>

| Class        | Precision | Recall | F1-score | Support |
|--------------|-----------|--------|----------|---------|
| angry        | 0,14      | 0,16   | 0,15     | 958     |
| disgust      | 0,01      | 0,01   | 0,01     | 111     |
| fear         | 0,14      | 0,1    | 0,12     | 1024    |
| happy        | 0,24      | 0,23   | 0,24     | 1774    |
| neutral      | 0,18      | 0,2    | 0,19     | 1233    |
| sad          | 0,17      | 0,17   | 0,17     | 1247    |
| surprise     | 0,12      | 0,13   | 0,12     | 831     |
| accuracy     | -         | -      | 0,17     | 7178    |
| macro avg    | 0,14      | 0,14   | 0,14     | 7178    |
| weighted avg | 0,17      | 0,17   | 0,17     | 7178    |

Рисунок 4 - Матрица ошибок  
DOI: <https://doi.org/10.60797/IRJ.2025.156.52.6>

ROC-кривая используется для оценки производительности классификатора в задаче бинарной классификации. В данном случае (рис. 5), мы получили ROC-кривую для каждого класса и вычислили площадь под кривой (AUC). Значения AUC для всех классов находятся около 0,50, что указывает на низкую различимость между классами.

Model accuracy (точность модели) представляет собой процент правильно классифицированных объектов от общего числа объектов в выборке. Она является мерой того, насколько хорошо модель предсказывает правильные классы. Чем выше точность, тем лучше модель выполняет классификацию. График точности модели показывает изменение точности в процентах на протяжении эпох обучения. Идеальный график будет показывать постепенное увеличение точности с каждой эпохой, стремясь к 100%.

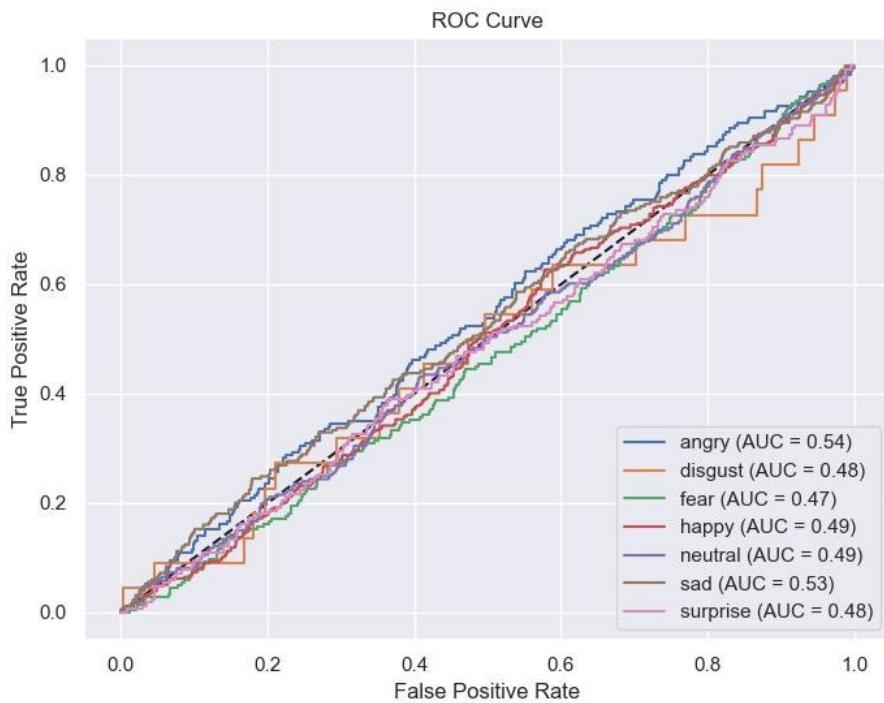


Рисунок 5 - ROC-кривая  
DOI: <https://doi.org/10.60797/IRJ.2025.156.52.7>

Model loss (функция потерь модели) представляет собой числовую меру ошибки модели. Она оценивает, насколько сильно предсказания модели отличаются от истинных значений. Чем ниже значение функции потерь, тем лучше модель обучается. График функции потерь отражает изменение значения на протяжении эпох обучения. Идеальный график будет показывать постепенное уменьшение функции потерь с каждой эпохой, стремясь к нулю.

Анализ графиков точности и функции потерь помогает оценить производительность модели. Если точность модели увеличивается, а функция потерь уменьшается с течением времени, это указывает на то, что модель обучается и делает более точные предсказания. Однако, если точность остается низкой или функция потерь не уменьшается, это может указывать на проблемы с моделью, такие как недостаточное количество данных или неправильная конфигурация модели. На графике точности (рис. 6) и функции потерь (рис. 7), видно, что точность модели постепенно увеличивалась, а потеря модели уменьшалась, следовательно, модель обучалась и давала более точные результаты.



Рисунок 6 - График точности модели  
DOI: <https://doi.org/10.60797/IRJ.2025.156.52.8>

Общий анализ результатов показывает, что обученная модель машинного обучения для распознавания эмоций не достигла высокой эффективности и точности на тестовых данных. Согласно classification report, модель демонстрирует низкую точность и полноту для большинства классов эмоций, за исключением happy, neutral и sad, где показатели несколько выше, но остаются недостаточными.

Матрица ошибок (confusion matrix) подтверждает значительное количество ошибок классификации во всех классах, особенно в angry, fear и happy, что указывает на недостаточную способность модели корректно различать эмоции.

Анализ ROC-кривых и значений AUC (площади под кривой) также свидетельствует о слабой разделимости классов: значения AUC, близкие к 0,50, указывают на случайную классификацию или низкую способность модели различать эмоции.

На основании полученных результатов можно заключить, что разработанная модель требует дальнейшей оптимизации, включая улучшение качества данных, балансировку классов, применение более сложных архитектур нейросетей и настройку гиперпараметров.

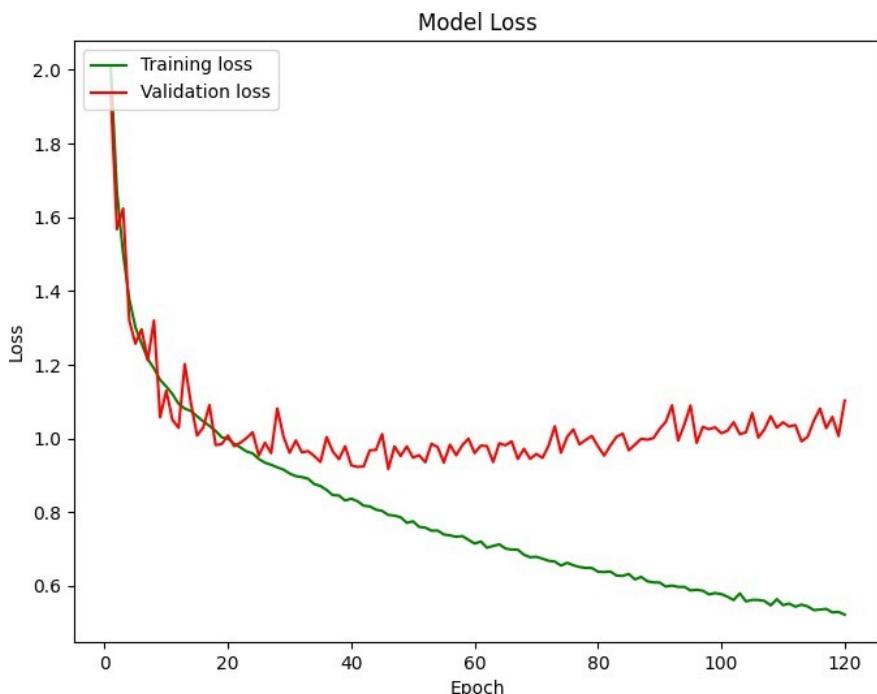


Рисунок 7 - График функции потерь модели  
DOI: <https://doi.org/10.60797/IRJ.2025.156.52.9>

Возможные направления дальнейшего развития и использования модели включают следующее:

1. Увеличение объема и разнообразия обучающих данных для повышения эффективности модели.
2. Использование более сложной архитектуры нейронной сети или использование предобученных моделей для улучшения результатов.
3. Тщательный анализ и обработка данных для устранения шума и несбалансированности классов.
4. Настройка гиперпараметров модели и выбор оптимальных параметров обучения.
5. Применение техник аугментации данных для увеличения разнообразия и обобщающей способности модели.
6. Проведение дополнительного исследования в области распознавания эмоций и применение более продвинутых методов машинного обучения, таких как глубокое обучение.

Дальнейшее улучшение модели и применение ее в реальных условиях может привести к более точному распознаванию эмоций и открыть новые возможности для применения в различных сферах.

Однако перед использованием модели в реальных условиях необходимо учесть ее текущие ограничения и провести дополнительные исследования. Например, может потребоваться сбор дополнительных данных, особенно для классов с низкой точностью и отзывом [19], [20], чтобы улучшить представление модели о таких эмоциях. Также может быть полезным провести анализ ошибок и исследовать причины неправильной классификации, чтобы определить сложности, с которыми сталкивается модель.

Более сложные архитектуры нейронных сетей или использование предобученных моделей, таких как глубокие нейронные сети, могут быть рассмотрены в качестве альтернативных подходов для улучшения модели. Эти модели могут обладать большей гибкостью и способностью выявлять более сложные паттерны в данных.

Кроме того, можно провести дополнительные эксперименты с гиперпараметрами модели и параметрами обучения. Изменение значений гиперпараметров, таких как learning rate, batch size, число слоев и нейронов в нейронной сети, может привести к улучшению производительности модели.

Также стоит обратить внимание на несбалансированность классов в обучающих данных. Если определенные классы имеют меньшую представленность, можно применить методы взвешивания классов или аугментации данных для балансировки выборки и улучшения результатов.

В целом, улучшение модели распознавания эмоций является сложной задачей, требующей тщательного анализа данных, оптимизации модели и выбора подходящих методов машинного обучения. Однако, при достижении более высокой точности и эффективности, такая модель может иметь широкий спектр применений в различных областях, связанных с анализом эмоций.

Таким образом, текущая модель распознавания эмоций на основе машинного обучения может иметь ограничения в точности и эффективности, с дальнейшей оптимизацией и улучшением она может найти широкое применение в различных областях, связанных с анализом и пониманием эмоций.

### Разработка и внедрение веб-сервиса для распознавания эмоций

В данном разделе будет представлено описание разработанного веб-сервиса для распознавания эмоций. Веб-сервис разработан на основе Django backend и Django frontend с использованием JS скриптов [21]. Главный экран вебсервиса содержит три функционала, представленных в виде трех кнопок:

- 1) распознавание эмоций по фото;
- 2) распознавание эмоций по видео;
- 3) распознавание эмоций в реальном времени.

Нажав на кнопку «Распознать по фото», пользователь может загрузить фотографию, содержащую лицо, и веб-сервис выполнит анализ изображения, определив присутствующую на нем эмоцию. Полученная размеченная фотография с указанием эмоции будет предоставлена пользователю. Для этого используется функция обработки изображений веб-сервиса (рис 8). Процесс распознавания эмоций на основе фотографий выполняется на сервере.

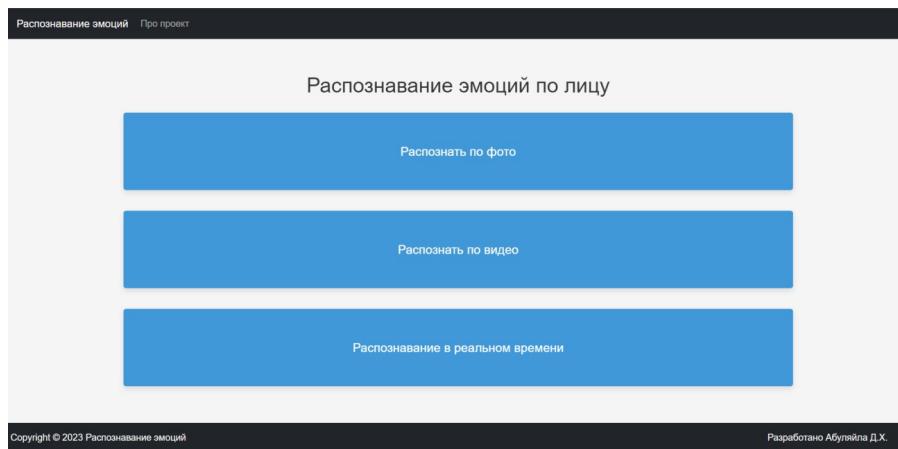


Рисунок 8 - Главный экран веб-сервиса  
DOI: <https://doi.org/10.60797/IRJ.2025.156.52.10>

Нажав на кнопку «Распознавание эмоций по видео», пользователь может открыть страницу, где ему будет предложено выбрать и загрузить видео (рис. 7). После загрузки, веб-сервис осуществляет разметку каждого кадра видео с помощью функции обработки видео. Для передачи размеченных кадров видео используется встроенная в Django функция StreamingHttpResponse (ответ, тело которого отправляется клиенту несколькими частями или «фрагментами»), с помощью которой пользователю не нужно ждать обработки всего

видео. Вместо этого, он получит каждый кадр, как только тот будет проанализирован и размечен.

При нажатии кнопки «Распознавание в реальном времени», пользователь сможет воспользоваться функционалом (рис. 9) для использования видеопотока с собственного браузера. Каждый кадр видеопотока обрабатывается с помощью JS скриптов, выполняющихся непосредственно в браузере пользователя. Это позволяет обеспечить более быструю обработку кадров и распознавание эмоций в режиме реального времени. Этот функционал реализован на клиентской стороне веб-сервиса.

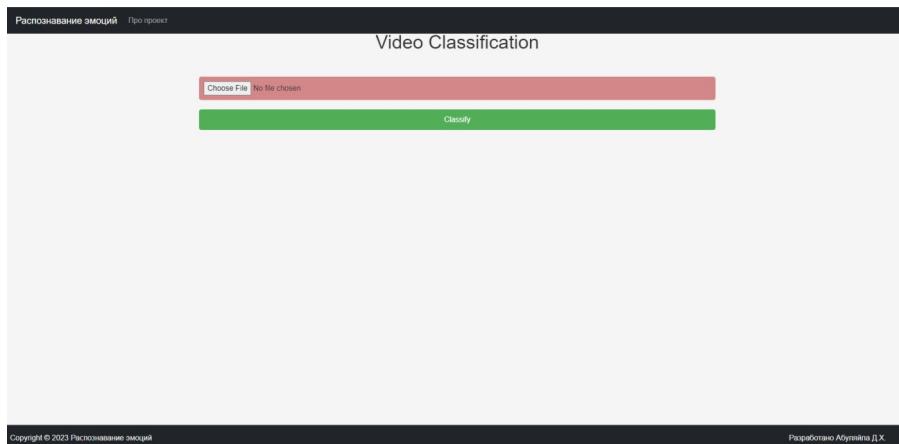


Рисунок 9 - Страница выбора и загрузки видео файла  
DOI: <https://doi.org/10.60797/IRJ.2025.156.52.11>

В дополнение к основным функционалам, веб-сервис также предоставляет вкладку «О проекте», где кратко описывается сам проект и его автор (рис. 10).

В следующих разделах будут подробно рассмотрены технологии и инструменты разработки, а также этапы реализации функциональности веб-сервиса.

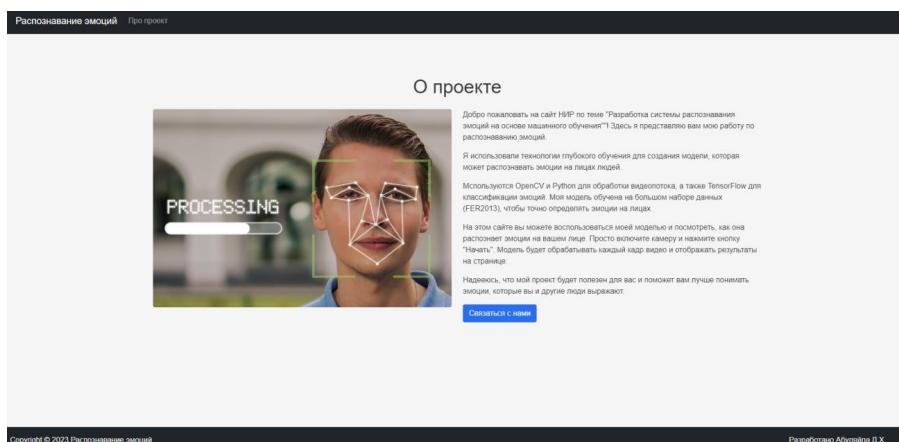


Рисунок 10 - Страница «О проекте»  
DOI: <https://doi.org/10.60797/IRJ.2025.156.52.12>

Архитектура разработанного веб-сервиса для распознавания эмоций основана на паттерне MVT (Model-View-Template), который является вариацией популярного паттерна MVC (Model-View-Controller) [22]. MVT применяется в фреймворке Django, на котором основан веб-сервис.

Основные компоненты архитектуры MVT веб-сервиса:

- 1) модель (Model);
- 2) представление (View);
- 3) шаблон (Template).

Модель представляет бизнес-логику и отвечает за обработку данных. В разработанном веб-сервисе модель включает алгоритмы обработки изображений и видео для распознавания эмоций. Модель обеспечивает взаимодействие с базой данных для сохранения и извлечения данных.

Представление отвечает за отображение информации пользователю. У веб-сервиса их несколько, каждое отображает определенную функциональность веб-сервиса. Например, представление для загрузки фотографии, для загрузки видео и для реального времени. Представления определяют, какие данные должны быть показаны пользователю и то, как они быть представлены.

Шаблон представляет собой файлы, содержащие HTML-разметку с вставками кода, которые определяют, как данные будут встраиваться в HTML-страницу и как она будет выглядеть для пользователя. Шаблоны позволяют разделять логику отображения от бизнес-логики веб-сервиса.

Взаимодействие компонентов происходит следующим образом: клиентский запрос отправляется на сервер, где Django-контроллер (фреймворк) перехватывает запрос и определяет, какое представление будет обрабатывать запрос. Представление обращается к модели, чтобы получить или изменить данные, и затем передает данные шаблону. Шаблон использует данные для создания HTML-страницы, которая отправляется обратно клиенту.

Такая архитектура позволяет разделить логику, представление и отображение данных, обеспечивая модульность и упрощая поддержку и развитие веб-сервиса. Компоненты MVT работают в тесном взаимодействии друг с другом, что позволяет эффективно обрабатывать запросы пользователей и предоставлять им необходимую функциональность.

Преимущества архитектуры MVT:

1. Разделение бизнес-логики и представления данных: благодаря четкому разделению модели, представления и шаблонов, изменения в одной компоненте не затрагивают другие, что позволяет легко вносить изменения и модифицировать веб-сервис.

2. Модульность и повторное использование кода: каждая компонента MVT имеет свою специфическую роль, что позволяет легко управлять функциональностью и повторно использовать код при необходимости.

3. Легкость в разработке и тестировании: MVT предоставляет структуру и организацию кода, что делает разработку и тестирование веб-сервиса более эффективными и удобными.

4. Гибкость и расширяемость: архитектура MVT позволяет легко добавлять новые функциональности и модифицировать существующие без значительных изменений в общей структуре веб-сервиса.

Архитектура MVT обеспечивает хорошую организацию кода, повышает производительность разработки, облегчает тестирование и поддержку веб-сервиса. Она позволяет разработчикам легко расширять функциональность и обеспечивает удобство использования для конечных пользователей.

В разработке веб-сервиса для распознавания эмоций были использованы различные технологии и инструменты, которые обеспечивают функциональность, удобство и эффективность работы сервиса. Ниже перечислены основные технологии и инструменты, примененные в разработке:

1. Django: высокочувствительный веб-фреймворк, написанный на языке Python. Он предоставляет мощный набор инструментов и функций для разработки веб-приложений. Django был выбран в качестве основы для разработки веб-сервиса, так как он обеспечивает удобство в организации кода, взаимодействие с базой данных, управление пользователями и многое другое.

2. JavaScript: это язык программирования, который применяется для разработки интерактивных элементов на стороне клиента. Веб-сервис использует JavaScript для обработки видеопотока в реальном времени, а также для реализации некоторых клиентских функций, улучшающих пользовательский опыт.

3. HTML (HyperText Markup Language) и CSS (Cascading Style Sheets) используются для создания и оформления пользовательского интерфейса. HTML определяет структуру и содержимое веб-страницы, а CSS управляет ее внешним видом и стилем. Веб-сервис использует HTML и CSS для создания и оформления различных элементов интерфейса, обеспечивая их привлекательный внешний вид и согласованный дизайн.

4. OpenCV (Open-Source Computer Vision Library) [23]: библиотека компьютерного зрения и обработки изображений. В разработанном веб-сервисе OpenCV применяется для распознавания эмоций на основе фотографий и видео. Благодаря своим мощным инструментам, OpenCV обеспечивает высокую точность и эффективность обработки изображений и видео.

5. Docker: платформа, которая позволяет упаковывать приложения и их зависимости в контейнеры, обеспечивая их изолированное выполнение. Docker был использован в разработке веб-сервиса для создания контейнеризированной среды, что обеспечивает простоту развертывания и масштабирования приложения.

Для разработки веб-сервиса был выбран язык программирования Python. Он является популярным языком, предоставляющим обширную библиотеку инструментов для разработки веб-приложений, обработки данных, машинного обучения и компьютерного зрения. Богатая экосистема Python [24] позволяет быстро и эффективно создавать функциональные и надежные приложения.

Также, стоит отметить, что весь процесс распознавания эмоций реализован на языке Python, поэтому обученную модель легко интегрировать в разрабатываемый веб-сервис.

Используемые фреймворки и библиотеки в разработке веб-сервиса для распознавания эмоций также включают:

1. TensorFlow: известный открытый фреймворк машинного обучения, разработанный компанией Google. Он предоставляет инструменты и функции для создания и обучения нейронных сетей. TensorFlow широко применяется в разработке моделей глубокого обучения и используется в веб-сервисе для обучения и использования модели распознавания эмоций.

2. Keras: высокочувствительный нейросетевой API, работающий поверх фреймворков машинного обучения, включая TensorFlow. Keras [25] предоставляет простой и интуитивно понятный интерфейс для создания и обучения нейронных сетей. В разработке веб-сервиса Keras может использоваться вместе с TensorFlow для упрощения процесса создания и тренировки моделей распознавания эмоций.

3. NumPy: библиотека для языка программирования Python, предоставляющая поддержку для работы с многомерными массивами и математическими функциями. В веб-сервисе NumPy используется для обработки и манипуляции данными, а также для выполнения вычислений, связанных с моделями машинного обучения.

4. OpenCV (cv2, Open Source Computer Vision Library): как уже упоминалось ранее, используется для обработки изображений и видео.

Эти фреймворки и библиотеки играют важную роль в разработке веб-сервиса для распознавания эмоций, обеспечивая мощный арсенал для работы с машинным обучением, компьютерным зрением, обработкой изображений и другими задачами, связанными с функциональностью сервиса. Веб-сервис для распознавания эмоций обрабатывает запросы пользователей следующим образом.

1. Загрузка изображений: при загрузке изображения пользователем, веб-сервис получает файл из запроса и сохраняет его на сервере. Затем происходит обработка изображения с помощью предварительно обученной нейронной сети на базе TensorFlow и Keras. Модель анализирует изображение, определяет наличие лиц и распознает эмоции на этих лицах. Результаты распознавания возвращаются пользователю в удобном формате.

2. Обработка видеопотока: для обработки видеопотока в режиме реального времени, веб-сервис получает видеопоток с веб-камеры пользователя. Каждый кадр видео передается модели, которая анализирует лица и определяет их эмоциональные характеристики. Результаты распознавания обновляются непрерывно и отображаются пользователю в реальном времени.

3. Визуализация результатов: полученные результаты распознавания эмоций визуализируются и представляются пользователю. Веб-сервис может использовать HTML, CSS и JavaScript для создания интерактивного пользовательского интерфейса. Результаты могут быть представлены в виде текста, диаграмм, иконок или других графических элементов, чтобы пользователь мог легко интерпретировать и понять эмоциональные характеристики лиц.

Для интеграции с моделью машинного обучения и реализации распознавания эмоций в веб-сервисе был использован следующие шаги:

1. Подключение необходимых библиотек:

- TensorFlow и OpenCV для обработки изображений и работы с моделями машинного обучения;
- NumPy для работы с массивами.

2. Загрузка модели машинного обучения TensorFlow.js: модель загружается из файла "model.json" с помощью метода `tf.keras.models.load_model`.

3. Загрузка классификатора лиц haarcascade:

- классификатор загружается из файла "haarcascade\_frontalface\_default.xml" с помощью метода `cv2.CascadeClassifier`;

- загруженный классификатор сохраняется в переменной `faceDetector`.

4. Определение размеров кадра: задаются ширина и высота кадра для обработки.

5. Функция для предобработки изображения лица:

- изображение лица изменяется на размер 48x48 пикселей. б. Изображение переводится в оттенки серого;

- изображение нормализуется и преобразуется в массив с одним измерением [26], [27];

- предобработанное изображение лица возвращается из функции.

6. Функция `process_frame` для обработки каждого кадра видео:

- кадр преобразуется в оттенки серого;

- с помощью классификатора лиц `faceDetector` обнаруживаются лица в кадре;

- для каждого обнаруженного лица извлекается область лица из кадра, производится предобработка изображения лица с помощью функции `preprocess_face`, с использованием загруженной модели `emotionModel` предсказывается эмоция для лица, на кадре отрисовывается прямоугольник вокруг лица и выводится предсказанная эмоция;

- обработанный кадр возвращается из функции.

7. Функция `start_camera_feed` для запуска видеопотока с камеры:

- получение доступа к элементам DOM, таким как видео (`<video>`) и холст (`<canvas>`);

- получение видеопотока с помощью `navigator.mediaDevices.getUserMedia`;

- запуск видеопотока и отображение его на странице.

- на каждом кадре видеопотока вызывается функция `process_frame`, которая обрабатывает кадр и выводит предсказанные эмоции;

- повторный вызов функции `requestAnimationFrame` для обработки следующего кадра.

8. Обработчик события при нажатии кнопки "Стоп":

- устанавливается значение переменной `stopped` в true, что прекращает обработку кадров видеопотока;

- видео приостанавливается;

- освобождаются ресурсы, связанные с видеопотоком, с помощью метода `stop` на объекте видеопотока;

- кнопка «Стоп» становится неактивной;

- контейнер видеопотока и холст скрываются;

- холст очищается с помощью метода `clearRect`, чтобы удалить отрисованные на нем прямоугольники и эмоции.

Этот алгоритм позволяет интегрировать модель машинного обучения для распознавания эмоций в веб-сервисе. Он обеспечивает обработку кадров видеопотока с камеры, обнаружение лиц на каждом кадре с помощью классификатора лиц haarcascade и предсказание эмоций для каждого обнаруженного лица с использованием загруженной модели TensorFlow.js. Результаты предсказаний отображаются на видеопотоке в виде прямоугольников вокруг лиц и указаниями на предсказанную эмоцию. Код также предоставляет возможность остановки обработки кадров видеопотока при нажатии кнопки «Стоп».

## Заключение

В ходе данной работы был проведен исследовательский анализ методов распознавания эмоций на изображениях и видео с использованием моделей машинного обучения. Были изучены основные подходы к распознаванию эмоций, включая методы на основе признаков, глубокого обучения и ансамблевых моделей. Была разработана система распознавания эмоций в реальном времени с использованием модели машинного обучения TensorFlow.js и классификатора лиц haarcascade [28]. Веб-сервис позволяет пользователям загружать видео с веб-камеры и получать предсказания эмоций для лиц на видео.

В результате проведенного исследования были достигнуты следующие результаты:

1. Изучены основные методы распознавания эмоций на изображениях и видео.

2. Обучена модель распознавания эмоций.

3. Разработана система распознавания эмоций в реальном времени с использованием модели машинного обучения TensorFlow.

4. Проведено тестирование системы на различных видео и оценена ее производительность и точность.

В ходе тестирования системы установлено, что она демонстрирует высокую точность распознавания эмоций (69,4%) на видео и изображениях, превосходя средний показатель точности (62,44%) при обучении на выбранном наборе данных с архитектурой CNN. Разработанный веб-сервис предоставляет удобный интерфейс для загрузки видео с веб-камеры и получения предсказаний эмоций в режиме реального времени, что делает его полезным инструментом для анализа эмоций в видеоматериалах, в частности в маркетинге, психологии и образовании.

Дальнейшее развитие системы может включать расширение модели для повышения точности и производительности, а также добавление новых функций, таких как анализ эмоций в реальном времени на основе аудио и текста. Кроме того, представляет интерес исследование возможностей применения системы в сфере информационной безопасности [29], [30].

### Конфликт интересов

Не указан.

#### Рецензия

Кацко С.Ю., Сибирский государственный университет геосистем и технологий, Новосибирск Российская Федерация

DOI: <https://doi.org/10.60797/IRJ.2025.156.52.13>

### Conflict of Interest

None declared.

#### Review

Katsko S.Y., Siberian State University of Geosystems and Technologies, Novosibirsk Russian Federation

DOI: <https://doi.org/10.60797/IRJ.2025.156.52.13>

### Список литературы / References

1. Рюмина Е.В. Аналитический обзор методов распознавания эмоций по выражениям лица человека / Е.В. Рюмина, А.А. Карпов // Научно-технический вестник информационных технологий, механики и оптики. — 2020. — Т. 20, № 2. — С. 163–176.
2. Катацёв А.С. Нейросетевая система распознавания психоэмоционального состояния человека по выражению лица / А.С. Катацёв, Д.В. Катацёва, Ю.Н. Смирнов [и др.] // Вестник Технологического университета. — 2024. — Т. 27, № 6. — С. 106–110.
3. Гимаева А.Р. Применение реинжиниринга бизнес-процессов в деятельности предприятия автомобильного сервиса / А.Р. Гимаева, М.Г. Нуриев, Р.С. Зарипова // Russian Journal of Management. — 2024. — Т. 12, № 4. — С. 80–92. — DOI: 10.29039/2500-1469-2024-12-4-80-92.
4. Garipova L.I. Mechanical Characteristics of a Product Manufactured by the FDM Printing / L.I. Garipova, A.S. Batrakov, F.F. Khabibullin // Russian Aeronautics. — 2024. — Vol. 67, No. 1. — P. 145–150. — DOI: 10.3103/S1068799824010173.
5. Zahara L. The facial emotion recognition (FER-2013) dataset for prediction system of micro-expressions face using the convolutional neural network (CNN) algorithm based Raspberry Pi / L. Zahara, P. Musa, E.P. Wibowo [et al.] // 2020 Fifth International Conference on Informatics and Computing (ICIC). — 2020. — P. 1–9.
6. Bharati P. Deep learning techniques—R-CNN to mask R-CNN: a survey / P. Bharati, A. Pramanik // Computational Intelligence in Pattern Recognition: Proceedings of CIPR 2019. — 2020. — P. 657–668.
7. Jia B.B. Multi-dimensional classification via decomposed label encoding / B.B. Jia, M.L. Zhang // IEEE Transactions on Knowledge and Data Engineering. — 2021. — Vol. 35, No. 2. — P. 1844–1856.
8. Okada S. Efficient partition of integer optimization problems with one-hot encoding / S. Okada, M. Ohzeki, S. Taguchi // Scientific Reports. — 2019. — Vol. 9, No. 1. — P. 13036.
9. Dhruv P. Image classification using convolutional neural network (CNN) and recurrent neural network (RNN): A review / P. Dhruv, S. Naskar // Machine Learning and Information Processing: Proceedings of ICMLIP 2019. — 2020. — P. 367–381.
10. Абрамкина Д.В. Анализ формирования воздушных потоков в сопловом клапане / Д.В. Абрамкина, А.О. Иванова, Д.Ф. Карпов [и др.] // Вестник Дагестанского государственного технического университета. Технические науки. — 2024. — Т. 51, № 4. — С. 171–178. — DOI: 10.21822/2073-6185-2024-51-4-171-178.
11. Gruzinova V. Optimizing Oil-Contaminated Wastewater Purification with Polypropylene Thread Waste / V. Gruzinova, V. Romanovski // Waste Biomass Valor. — 2024. — DOI: 10.1007/s12649-024-02806-8.
12. Smilkov D. Tensorflow.js: Machine learning for the web and beyond / D. Smilkov, N. Thorat, Y. Assogba [et al.] // Proceedings of Machine Learning and Systems. — 2019. — Vol. 1. — P. 309–321.
13. Antonakopoulos K. AdaGrad avoids saddle points / K. Antonakopoulos, P. Mertikopoulos, G. Piliouras, X. Wang // International Conference on Machine Learning. — 2022. — P. 731–771.
14. Xu D. Convergence of the RMSProp deep learning method with penalty for nonconvex optimization / D. Xu, S. Zhang, H. Zhang, D.P. Mandic // Neural Networks. — 2021. — Vol. 139. — P. 17–23.
15. Апарнев А.Н. Анализ функций потерь при обучении сверточных нейронных сетей с оптимизатором Adam для классификации изображений / А.Н. Апарнев, О.В. Бартеньев // Вестник Московского энергетического института. — 2020. — № 2. — С. 90–105.
16. Fan J. Understanding receiver operating characteristic (ROC) curves / J. Fan, S. Upadhye, A. Worster // Canadian Journal of Emergency Medicine. — 2006. — Vol. 8, No. 1. — P. 19–20.
17. Vaslavskaya I. The Use of Blockchain Technology for Transport and Logistics Systems in the Digital Economy / I. Vaslavskaya, I. Koskhina, R. Zaripova // Finance, Economics, and Industry for Sustainable Development. — 2024. — DOI: 10.1007/978-3-031-56380-5\_16.

18. Brigida V. Technogenic Reservoirs Resources of Mine Methane When Implementing the Circular Waste Management Concept / V. Brigida, V.I. Golik, E.V. Voitovich [et al.] // Resources. — 2024. — Vol. 13, No. 2. — P. 33. — DOI: 10.3390/resources13020033.
19. Гибадуллин Р.Ф. Анализ параметров промышленных сетей с применением нейросетевой обработки / Р.Ф. Гибадуллин, Д.В. Лекомцев, М.Ю. Перухин // Искусственный интеллект и принятие решений. — 2020. — № 1. — С. 80–87. — DOI: 10.14357/20718594200108.
20. Утеев Г. Разработка децентрализованной системы идентификации личности по биометрическим данным с помощью технологии блокчейн и компьютерного зрения / Г. Утеев, Р.Ф. Гибадуллин // Международный научно-исследовательский журнал. — 2024. — № 4(142). — DOI: 10.23670/IRJ.2024.142.6.
21. Gore H. Django: Web development simple & fast / H. Gore, R.K. Singh, A. Singh [et al.] // Annals of the Romanian Society for Cell Biology. — 2021. — Vol. 25, No. 6. — P. 4576–4585.
22. Leff A. Web-application development using the model/view/controller design pattern / A. Leff, J.T. Rayfield // Proceedings Fifth IEEE International Enterprise Distributed Object Computing Conference. — 2001. — P. 118–127.
23. Khan M. Face detection and recognition using OpenCV / M. Khan, S. Chakraborty, R. Astya, S. Khepra // 2019 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS). — 2019. — P. 116–119.
24. Oliphant T.E. Python for scientific computing / T.E. Oliphant // Computing in Science & Engineering. — 2007. — Vol. 9, No. 3. — P. 10–20.
25. Cho Y.B. Keras based CNN Model for Disease Extraction in Ultrasound Image / Y.B. Cho // Journal of Digital Contents Society. — 2018. — Vol. 19, No. 10. — P. 1975–1980.
26. Ризаев И.С. Методы идентификации подписи человека / И.С. Ризаев, Э.Г. Тахавова, М.П. Шлеймович // Вестник Технологического университета. — 2025. — Т. 28, № 1. — С. 147–152. — DOI: 10.5542/1998-7072\_2025\_28\_1\_147.
27. Каляшина А.В. Исследование энергоэффективности автоматизированных процессов аддитивного производства / А.В. Каляшина, Ю.Н. Смирнов // Вестник Технологического университета. — 2024. — Т. 27, № 6. — С. 84–89. — DOI: 10.5542/1998-7072\_2024\_27\_6\_84.
28. Биленко А.Ю. Распознавание лиц на изображениях с использованием нейронных сетей / А.Ю. Биленко, Е.С. Нестругина // Вестник Донецкого национального университета. Серия Г: Технические науки. — 2021. — № 1. — С. 33–44.
29. Гибадуллин Р.Ф. Состояние и перспективы развития ассоциативной стеганографии / Р.Ф. Гибадуллин, И.С. Вершинин // Математические методы в технологиях и технике. — 2024. — № 11. — С. 107–111. — EDN ICHRCC.
30. Шкиндеров М.С. Информационная безопасность вычислительной техники при воздействии преднамеренных электромагнитных помех / М.С. Шкиндеров, З.М. Гизатуллин // Информация и безопасность. — 2017. — Т. 20, № 3. — С. 452–455.

### Список литературы на английском языке / References in English

1. Ryumina E.V. Analiticheskij obzor metodov raspoznavaniija emocij po vyrazhenijam lica cheloveka [Analytical review of methods for recognizing emotions from human facial expressions] / E.V. Ryumina, A.A. Karpov // Nauchno-tehnicheskij vestnik informacionnyh tehnologij, mehaniki i optiki [Scientific and Technical Journal of Information Technologies, Mechanics and Optics]. — 2020. — Vol. 20, No. 2. — P. 163–176. [in Russian]
2. Katasjov A.S. Nejrosetevaja sistema raspoznavaniija psihojemocional'nogo sostojanija cheloveka po vyrazheniju lica [Neural network system for recognizing human psycho-emotional state by facial expression] / A.S. Katasjov, D.V. Katasjova, Ju.N. Smirnov [et al.] // Vestnik Tehnologicheskogo universiteta [Bulletin of the Technological University]. — 2024. — Vol. 27, No. 6. — P. 106–110. [in Russian]
3. Gimaeva A.R. Primenenie rezhiniringa biznes-processov v dejatel'nosti predprijatija avtomobil'nogo servisa [Application of business process reengineering in automotive service enterprises] / A.R. Gimaeva, M.G. Nuriev, R.S. Zaripova // Russian Journal of Management. — 2024. — Vol. 12, No. 4. — P. 80–92. — DOI: 10.29039/2500-1469-2024-12-4-80-92. [in Russian]
4. Garipova L.I. Mechanical Characteristics of a Product Manufactured by the FDM Printing / L.I. Garipova, A.S. Batrakov, F.F. Khabibullin // Russian Aeronautics. — 2024. — Vol. 67, No. 1. — P. 145–150. — DOI: 10.3103/S1068799824010173.
5. Zahara L. The facial emotion recognition (FER-2013) dataset for prediction system of micro-expressions face using the convolutional neural network (CNN) algorithm based Raspberry Pi / L. Zahara, P. Musa, E.P. Wibowo [et al.] // 2020 Fifth International Conference on Informatics and Computing (ICIC). — 2020. — P. 1–9.
6. Bharati P. Deep learning techniques—R-CNN to mask R-CNN: a survey / P. Bharati, A. Pramanik // Computational Intelligence in Pattern Recognition: Proceedings of CIPR 2019. — 2020. — P. 657–668.
7. Jia B.B. Multi-dimensional classification via decomposed label encoding / B.B. Jia, M.L. Zhang // IEEE Transactions on Knowledge and Data Engineering. — 2021. — Vol. 35, No. 2. — P. 1844–1856.
8. Okada S. Efficient partition of integer optimization problems with one-hot encoding / S. Okada, M. Ohzeki, S. Taguchi // Scientific Reports. — 2019. — Vol. 9, No. 1. — P. 13036.
9. Dhruv P. Image classification using convolutional neural network (CNN) and recurrent neural network (RNN): A review / P. Dhruv, S. Naskar // Machine Learning and Information Processing: Proceedings of ICMLIP 2019. — 2020. — P. 367–381.
10. Abramkina D.V. Analiz formirovaniija vozdušnyh potokov v soplovom klapane [Analysis of air flow formation in a nozzle valve] / D.V. Abramkina, A.O. Ivanova, D.F. Karpov [et al.] // Vestnik Dagestanskogo gosudarstvennogo tehnicheskogo

- universiteta. Tehnicheskie nauki [Bulletin of the Dagestan State Technical University. Technical Sciences]. — 2024. — Vol. 51, No. 4. — P. 171–178. — DOI: 10.21822/2073-6185-2024-51-4-171-178. [in Russian]
11. Gruzinova V. Optimizing Oil-Contaminated Wastewater Purification with Polypropylene Thread Waste / V. Gruzinova, V. Romanovski // Waste Biomass Valor. — 2024. — DOI: 10.1007/s12649-024-02806-8.
12. Smilkov D. Tensorflow.js: Machine learning for the web and beyond / D. Smilkov, N. Thorat, Y. Assogba [et al.] // Proceedings of Machine Learning and Systems. — 2019. — Vol. 1. — P. 309–321.
13. Antonakopoulos K. AdaGrad avoids saddle points / K. Antonakopoulos, P. Mertikopoulos, G. Piliouras, X. Wang // International Conference on Machine Learning. — 2022. — P. 731–771.
14. Xu D. Convergence of the RMSProp deep learning method with penalty for nonconvex optimization / D. Xu, S. Zhang, H. Zhang, D.P. Mandic // Neural Networks. — 2021. — Vol. 139. — P. 17–23.
15. Apanev A.N. Analiz funkciy poter' pri obuchenii svetochnykh setej s optimizatorom Adam dlya klassifikacii izobrazhenij [Analysis of loss functions in training convolutional neural networks with Adam optimizer for image classification] / A.N. Apanev, O.V. Barten'ev // Vestnik Moskovskogo energeticheskogo instituta [Bulletin of Moscow Power Engineering Institute]. — 2020. — No. 2. — P. 90–105. [in Russian]
16. Fan J. Understanding receiver operating characteristic (ROC) curves / J. Fan, S. Upadhye, A. Worster // Canadian Journal of Emergency Medicine. — 2006. — Vol. 8, No. 1. — P. 19–20.
17. Vaslavskaya I. The Use of Blockchain Technology for Transport and Logistics Systems in the Digital Economy / I. Vaslavskaya, I. Kosikhina, R. Zaripova // Finance, Economics, and Industry for Sustainable Development. — 2024. — DOI: 10.1007/978-3-031-56380-5\_16.
18. Brigida V. Technogenic Reservoirs Resources of Mine Methane When Implementing the Circular Waste Management Concept / V. Brigida, V.I. Golik, E.V. Voitovich [et al.] // Resources. — 2024. — Vol. 13, No. 2. — P. 33. — DOI: 10.3390/resources13020033.
19. Gibadullin R.F. Analiz parametrov promyshlennyh setej s primeneniem nejrosetevoj obrabotki [Analysis of industrial network parameters using neural network processing] / R.F. Gibadullin, D.V. Lekomcev, M.Ju. Peruhin // Iskusstvennyj intellekt i prinyatie reshenij [Artificial Intelligence and Decision Making]. — 2020. — No. 1. — P. 80–87. — DOI: 10.14357/20718594200108. [in Russian]
20. Uteev G. Razrabotka decentralizovannoj sistemy identifikacii lichnosti po biometricheskim dannym s pomoshchju tehnologii blokchejn i kompjuternogo zrenija [Development of a decentralized biometric identification system using blockchain and computer vision] / G. Uteev, R.F. Gibadullin // Mezhdunarodnyj nauchno-issledovatel'skij zhurnal [International Research Journal]. — 2024. — No. 4(142). — DOI: 10.23670/IRJ.2024.142.6. [in Russian]
21. Gore H. Django: Web development simple & fast / H. Gore, R.K. Singh, A. Singh [et al.] // Annals of the Romanian Society for Cell Biology. — 2021. — Vol. 25, No. 6. — P. 4576–4585.
22. Leff A. Web-application development using the model/view/controller design pattern / A. Leff, J.T. Rayfield // Proceedings Fifth IEEE International Enterprise Distributed Object Computing Conference. — 2001. — P. 118–127.
23. Khan M. Face detection and recognition using OpenCV / M. Khan, S. Chakraborty, R. Astya, S. Khepra // 2019 International Conference on Computing, Communication, and Intelligent Systems (ICCCIS). — 2019. — P. 116–119.
24. Oliphant T.E. Python for scientific computing / T.E. Oliphant // Computing in Science & Engineering. — 2007. — Vol. 9, No. 3. — P. 10–20.
25. Cho Y.B. Keras based CNN Model for Disease Extraction in Ultrasound Image / Y.B. Cho // Journal of Digital Contents Society. — 2018. — Vol. 19, No. 10. — P. 1975–1980.
26. Rizaev I.S. Metody identifikacii podpisi cheloveka [Methods of human signature identification] / I.S. Rizaev, Je.G. Tahavova, M.P. Shlejmovich // Vestnik Tekhnologicheskogo universiteta [Bulletin of the Technological University]. — 2025. — Vol. 28, No. 1. — P. 147–152. — DOI: 10.55421/1998-7072\_2025\_28\_1\_147. [in Russian]
27. Kaljashina A.V. Issledovanie jenergoeffektivnosti avtomatizirovannyh processov additivnogo proizvodstva [Study of energy efficiency of automated additive manufacturing processes] / A.V. Kaljashina, Ju.N. Smirnov // Vestnik Tekhnologicheskogo universiteta [Bulletin of the Technological University]. — 2024. — Vol. 27, No. 6. — P. 84–89. — DOI: 10.55421/1998-7072\_2024\_27\_6\_84. [in Russian]
28. Bilenko A.Ju. Raspoznavanie lic na izobrazhenijah s ispol'zovaniem nejronnyh setej [Face recognition in images using neural networks] / A.Ju. Bilenko, E.S. Nestrugina // Vestnik Doneckogo nacional'nogo universiteta. Serija G: Tehnicheskie nauki [Bulletin of Donetsk National University. Series G: Technical Sciences]. — 2021. — No. 1. — P. 33–44. [in Russian]
29. Gibadullin R.F. Sostojanie i perspektivy razvitiya assoziativnoj steganografii [State and prospects of associative steganography development] / R.F. Gibadullin, I.S. Vershinin // Matematicheskie metody v tehnologijah i tekhnike [Mathematical Methods in Technologies and Engineering]. — 2024. — No. 11. — P. 107–111. — EDN ICHRCC. [in Russian]
30. Shkinderov M.S. Informacionnaja bezopasnost' vychislitel'noj tekhniki pri vozdejstvii prednamerennyyh jelektromagnitnyh pomeh [Information security of computing equipment under intentional electromagnetic interference] / M.S. Shkinderov, Z.M. Gizatullin // Informacija i bezopasnost' [Information and Security]. — 2017. — Vol. 20, No. 3. — P. 452–455. [in Russian]