Конференция научно-технического творчества школьников
«Лабиринты науки»

Секция «Информационно-телекоммуникационные технологии»

Тренажер эмоционального интеллекта с вербализацией выражений лиц

Выполнил ученик 10 класса
МОУ СОШ №36
Лукашов Дмитрий Андреевич
Руководитель: учитель информатики
МОУ СОШ №36
Гагарина Алиса Игоревна

Рыбинск, 2023-2024 учебный год

Содержание

Введение 3
1. Аналитический обзор методов и систем классификации эмоций 5
1.1 Анализ программных систем распознавания эмоций 5
1.2. Анализ методов распознавания эмоций 5
2. Проектирование и разработка системы тренажера эмоционального интеллекта 7
3. Разработка и обучение нейросетевых моделей классификации текстов и изображений лиц 8
3.1. Разработка и обучение нейронной сети классификации эмоций по изображению лица 8
3.2. Разработка и обучение нейронной сети классификации эмоциональной окраски текста 9
4. Экспериментальное исследование качества обучения моделей нейронных сетей 10
Заключение 11
Список источников 11
Приложение А Виды классификации базовых эмоций 12
Приложение Б Фрагмент листинга программного кода обучения нейронной сети по изображению 13
Приложение В Результаты эксперимента по классификации эмоций изображению лица 16
Приложение Г Фрагмент листинга программного кода предобработки текстового датасета 17

Введение

Эмоциональный интеллект (EQ) – способность человека понимать эмоции, как свои, так и других людей, и уметь использовать эти знания в своей жизни. Человеку с развитым эмоциональным интеллектом легче добиваться своих целей, развиваться за счет правильной интерпретации своих чувств, легче проводить переговоры, а в случае конфликта идти на компромисс. Если эмоциональный интеллект невысокий, то человеку трудно общаться и выражать свои мысли, так как он не может понять ни свои эмоции, ни своего собеседника, не способен к эмпатии, его в первую очередь волнуют свои интересы, а не цель диалога. Неумение контролировать свои эмоции часто приводит к психическим расстройствам и даже к зависимости от психоактивных веществ.

В настоящее время эмоциональный интеллект оценивается как более важная по сравнению с уровнем IQ способность, которую необходимо развивать, однако методики и программные средства для формирования навыков понимания эмоций в настоящее время недостаточно разработаны [1].

При анализе средств выражения эмоций различных модальностей: мимики лица и текстовых сообщений с помощью методов машинного обучения выявляется теоретическая проблема – сопоставление эмбеддингов вербальной и невербальной информации [2]. Важным аспектом является векторизация, так как большое количество математических моделей работают в векторных пространствах больших размерностей.

Стоит отметить взаимосвязь модальностей через концепцию векторного пространства слов визуального и текстового мешка слов (bag-of-words). Визуальное векторное пространство слов, полученных из изображений, представляется как набор векторов, каждый из которых - определенный объект или класс объектов. Набор векторов слов, которые используются для классификации и распознавания текста имеют и текстовые векторные пространства слов. Эти два вида мешков слов могут быть связаны через общие векторные пространства, позволяя выполнять задачи, такие как распознавание

объектов в изображениях на основе текстовых описаний и распознавание текста на основе изображений.

Гипотеза исследования заключается в следующем – поскольку существует один источник возникновения эмоции, то информация различных модальностей должна быть связана, следовательно, при разработке тренажера эмоционального интеллекта целесообразно использовать обработку вербальной и невербальной информации.

Цель работы: разработать тренажер эмоционального интеллекта на основе классификации тональности текстов и выражений лиц для развития навыков вербальной и невербальной оценки эмоций.

Для достижения цели поставлены и решены следующие **задачи**.

1. Проанализировать методы и программные средства для распознавания эмоций.

2. Разработать программную систему тренажера эмоционального интеллекта.

3. Разработать и обучить нейросетевые модели классификации эмоций по изображению лица и по тексту.

4. Выполнить экспериментальное исследование качества обучения моделей нейронных сетей.

Методы, которые используются в данной работе: машинное обучение, программирование, интеллектуальная обработка изображений и текста.

Теоретическая значимость заключается в исследовании способа развития эмоционального интеллекта на основе вербального и невербального представления эмоций.

Практическая значимость состоит в развитии и повышении уровня эмоционального интеллекта у людей в современном обществе. Оценка эмоций играет важную роль во многих сферах, включая маркетинг, психологию, медицину и других.

1. Аналитический обзор методов и систем классификации эмоций

1.1 Анализ программных систем распознавания эмоций

Мимика лица является главным средством выражения широкого спектра эмоций и передачи информации о состоянии человека окружающим. Однако существуют базовые эмоции, подходы к классификации которых приведены на рисунке А.1 приложения А.

Программа *EmoReact* разработана специально для анализа эмоциональной реакции людей на различные стимулы. Набор данных *EmoReact* включает данные об эмоциях детей в возрасте от четырех до четырнадцати лет. *EmoReact* содержит 1102 аудиовизуальных клипа, аннотированных для 17 различных эмоциональных состояний: шести основных эмоций, нейтральных, и девяти сложных эмоций, включая любопытство, неуверенность, волнение, внимательность, исследование, замешательство, тревогу, смущение и разочарование.

Программное обеспечение *Affectiva* предназначено для анализа и изучения эмоций людей с использованием технологий компьютерного зрения и машинного обучения. Отличительной черной *Affectiva* является способность распознавать и классифицировать эмоции, выражаемые лицом людей на видео [3].

Программа *FaceReader* использует алгоритмы машинного обучения для распознавания семи базовых эмоций: радость, грусть, злость, страх, удивление, презрение и отвращение. Также имеет возможность работать в режиме реального времени. Кроме того, программа может определять интенсивность выражения каждой эмоции на основе микроэкспрессий.

Данные программы не используют вербальный компонент оценки EQ.

1.2. Анализ методов распознавания эмоций

Существуют различные методы для распознавания эмоций по лицу, такие как традиционный анализ мимики, выражения лица, глаз и т.д., а также использование компьютерного зрения и алгоритмов машинного обучения.

Метод анализа мимики лица основан на изучении мимических движений глаз, бровей, рта и других частей лица. Он имеет ограниченную точность и требует определения набора субъективных и сложных правил.

Метод геометрических признаков основан на измерении геометрических параметров лица по ключевым точкам на лице. Метод не учитывает важные для распознавания эмоций текстуры и детали лица.

Методы компьютерного зрения, основанные на текстурных и цветовых признаках, требуют экспертных знаний для ручного выбора признаков.

Методы компьютерного зрения, основанные на глубоких нейронных сетях, позволяют в процессе обучения автоматически выделить необходимые признаки. Наиболее эффективны сверточные нейронные сети (CNN).

Анализ тональности текста (сентимент-анализ) – одна из задач, позволяющая изучить массив сообщений и иных данных и определить их эмоциональную окраску. Задача анализа тональности текста аналогична задаче классификации текста, где категориями текстов могут быть эмоциональные оценки. Для оценки тональности текста используются методы обработки естественного языка, основанные на правилах или на тональных словарях. Эти методы не обладают гибкостью и требуют больших затрат по составлению правил и словарей.

Подход на основе машинного обучения представляет анализ эмоциональной окраски текста как задачу классификации на основе датасета. Процесс создания систем с использованием этого метода будет состоять из следующих шагов: сбор исходных данных для обучения (набор текстов), каждый текст из обучающей выборки нужно представить в виде вектора признаков, для каждого текста указать тип тональности, выбор алгоритма классификации и обучение классификатора. Для данной задачи наиболее эффективны рекуррентные нейронные сети (RNN).

Учитывая современные тенденции, разрабатываемая система тренажера EQ будет основана на нейросетевых моделях.

2. Проектирование и разработка системы тренажера эмоционального интеллекта

Функциональная схема разработанной системы приведена на рисунке 1. Тренажер обучения эмоциональному интеллекту состоит из двух взаимосвязанных частей: классификатор эмоции по изображению лица (*IC*) и по тексту (*TC*). После обучения моделей возможны следующие режимы работы: по изображению эмоции написать текст и по тексту изобразить эмоцию мимикой. Результат обучения оценивается по близости ответов в различных признаковых пространствах.

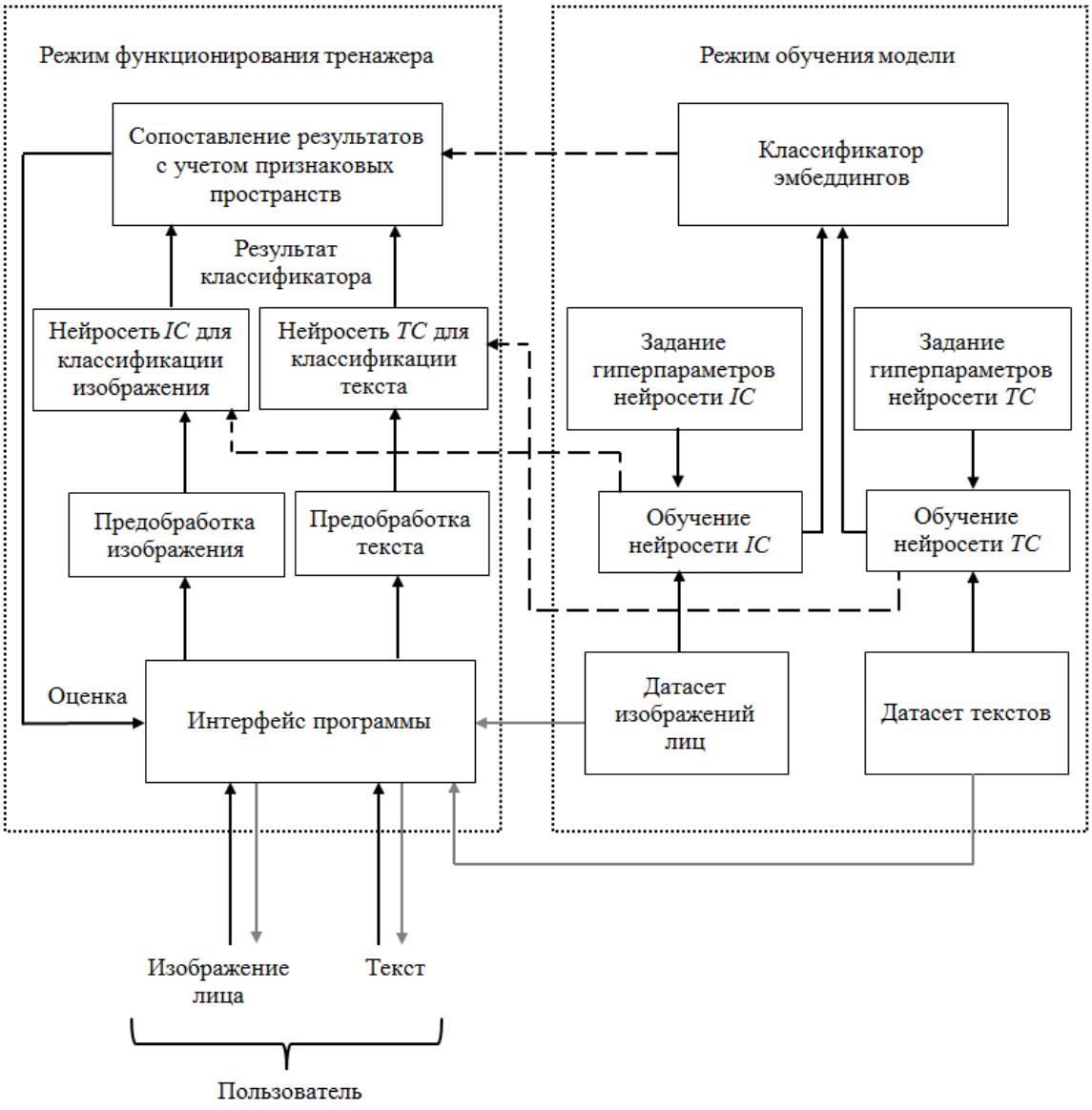


Рисунок 1 – Функциональная схема системы тренажера EQ

3. Разработка и обучение нейросетевых моделей классификации текстов и изображений лиц

3.1. Разработка и обучение нейронной сети классификации эмоций по изображению лица

Данные для обучения взяты с сервиса *Kaggle* [4]. Датасет содержит 35887 файлов. Он разбит на наборы изображений по эмоциям. Выбранные категории: «*angry*», «*disgust*», «*fear*», «*happy*», «*neutral*», «*sad*», «*surprised*».

Далее необходимо провести предварительную обработку данных: изменение размера изображений, нормализация значений пикселей, устранение шума. После этого требуется разбить данные на обучающую и тестовую выборки. Разделение проведено случайным образом в примерном соотношении 80% и 20% (28709 и 7178 файлов).

Рассмотрим архитектуру нейросети в данной программе [5]. *Conv2D* – сверточный слой. *BatchNormalization* выполняет нормализацию активаций предыдущего слоя. Он улучшает стабильность обучения, ускоряет сходимость. *Activation*. В данной архитектуре используется функция активации *ReLU* (*Rectified Linear Unit*). *MaxPooling2D* – слой пулинга, который уменьшает размерность пространственных признаков путем выбора максимального значения внутри окна пулинга. В данном случае, размер пулинга составляет 2x2. *Dropout* – метод регуляризации. Набор из этих слоев повторяется три раза с разными параметрами. В каждом повторении увеличивается количество фильтров свертки, начиная с 64 и заканчивая 512. Далее добавляется слой *Flatten* для преобразования многомерных данных в одномерный вектор, чтобы передать их в полносвязные слои. Затем следует полносвязный слой (*Dense*). Использует функцию активации *softmax* для предсказания вероятности каждого класса. В данной архитектуре используется оптимизатор *Adam*. Код обучения приведен в Приложении Б.

Результаты распознания эмоций тестового набора (рисунок В.1) приведены на рисунке В.2 (Приложение В).

3.2. Разработка и обучение нейронной сети классификации эмоциональной окраски текста

Вербализация, она отвечает за адекватное выражение собственных эмоций и правильно восприятие чужих эмоций, а также расширение своего словарного запаса для описания эмоций. Одно из определений вербализации это: словесное (вербальное) описание чувств, переживаний и др. Один из способов развития вербальной составляющей интеллекта это исключение из своей речи слов паразитов, заменяя их на более интеллектуальные аналоги, например текст «Ты меня достал» заменить на фразу «Я в тебе разочарован».

Хотя русский язык и является одним из самых распространённых, количество датасетов в сфере анализа тональности значительно меньше, чем на английском. Исследования классификации эмоций в русскоязычных текстах проводятся в больших количествах, но лишь небольшое количество авторов выкладывают свои наборы данных в открытый доступ. Рассмотрим следующие датасеты.

RuSentiment – открытый набор данных с примерами настроений из публикаций в соцсети ВКонтакте. Аннотирование ручное, классов тональности выделено 5 (негативное, позитивное, нейтральное, частые речевые обороты – подкатегория позитивной тональности, неясные).

Twitter Sentiment for 15 European Languages – датасет с примерами настроений, содержащий больше 1,6 млн. Twitter-сообщений на 15 языках, в том числе русском. Аннотирование ручное, классов тональности выделено 3.

Последовательность предобработки текста заключается в информации, ухудшающей качество машинного обучения. Этап подготовки текста включает операции: перевод всех букв в тексте в нижний или верхний регистры; удаление цифр (чисел) или замена на текстовый эквивалент; удаление пунктуации; удаление пробельных символов; дедубликация; токенизация; удаление стоп слов; стемминг; лемматизация; векторизация. Код предобработки текста приведен в Приложении Г.

4. Экспериментальное исследование качества обучения моделей нейронных сетей

Для оценки качества обучения использовался датасет Fer-2013, содержащий 35887 изображений человеческих лиц с определёнными эмоциями (7 штук). Разделяем эти данные на три выборки: тренировочный набор (training), проверочный (valid) и контрольный (test). На тренировочном наборе сеть будет тренироваться, после каждой эпохи тренировки в качестве оценки будут использоваться потери и точность для двух наборов – тренировочного и проверочного. По окончании работы с тренировочным набором поменяем их местами с проверочным и дообучим сеть на проверочном наборе (в качестве оценки используется тренировочный). В конце обучения производится проверка на тестовом наборе. Количество изображений в наборах: тренировочный набор – 28709, проверочный – 3589, тестовый – 3589. Мини пакет размером 64.

Входные данные нейронной сети – нормализованные изображения лиц 48x48 пикселей. Выходные данные нейронной сети – массив, состоящий из вероятностей для каждой эмоции.

Примеры графиков зависимости функций точности и потерь для обучающей и валидационной выборок от эпох обучения представлены на рисунке 2.

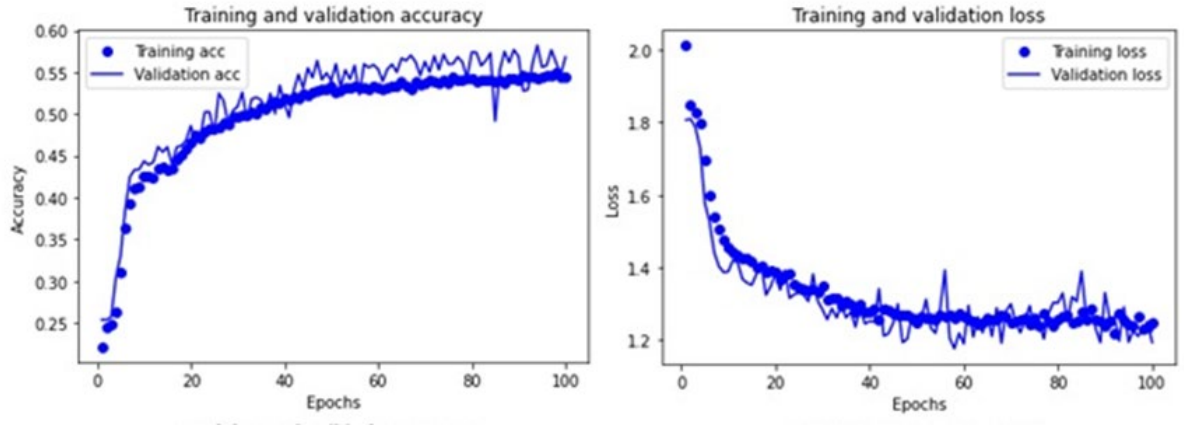


Рисунок 2 – Оценка качества обучения по функциям точности и потерь

Заключение

Разработанная программа сможет быть применена для развития эмоционального интеллекта в различных сферах, включая социальные исследования, медицину, разработку интерфейсов и даже виртуальную реальность. Положительные эффекты состоят в развитии эмпатии, повышении эффективности коммуникации и создании более персонализированных и адаптивных систем.

Сближение результатов классификации по вербальному и визуальному эмбеддингам позволит судить о качестве обучения на тренажере. Новые метрики будут способствовать более точному анализу и принятию соответствующих решений, основанных на эмоциональных данных.

Список источников

1. Пазельская А., Соловьев А. Метод определения эмоций в текстах на русском языке. // Компьютерная лингвистика и интеллектуальные технологии: «Диалог-2011». Сб. научных статей / Вып. 11 (18).- М.: Изд-во РГГУ, 2011.– С.510-523.
2. Tarasov D.S. Deep Recurrent Neural Networks for Multiple Language Aspect-Based Sentiment Analysis // Computational Linguistics and Intellectual Technologies: Proceedings of Annual International Conference «Dialogue-2015», Issue 14(21), V.2, pp. 65-74 (2015).
3. Affectiva [Электронный ресурс] / Режим доступа: URL: https://www.affectiva.com/ (дата обращения: 15.11.2023 г.)
4. Kaggle / [Электронный ресурс] / Режим доступа: - URL: https://www.kaggle.com/ (дата обращения: 15.11.2023 г.).
5. End-to-End Multimodal Emotion Recognition using Deep Neural Networks / Panagiotis Tzirakis, George Trigeorgis, Mihalis A. Nicolaou, Björn Schuller, and Stefanos Zafeiriou // JOURNAL OF LATEX CLASS FILES, vol. 14, no. 8, August, 2015 [электронный ресурс] URL: https://arxiv.org/pdf/1704.08619.pdf

Приложение А

Виды классификации базовых эмоций

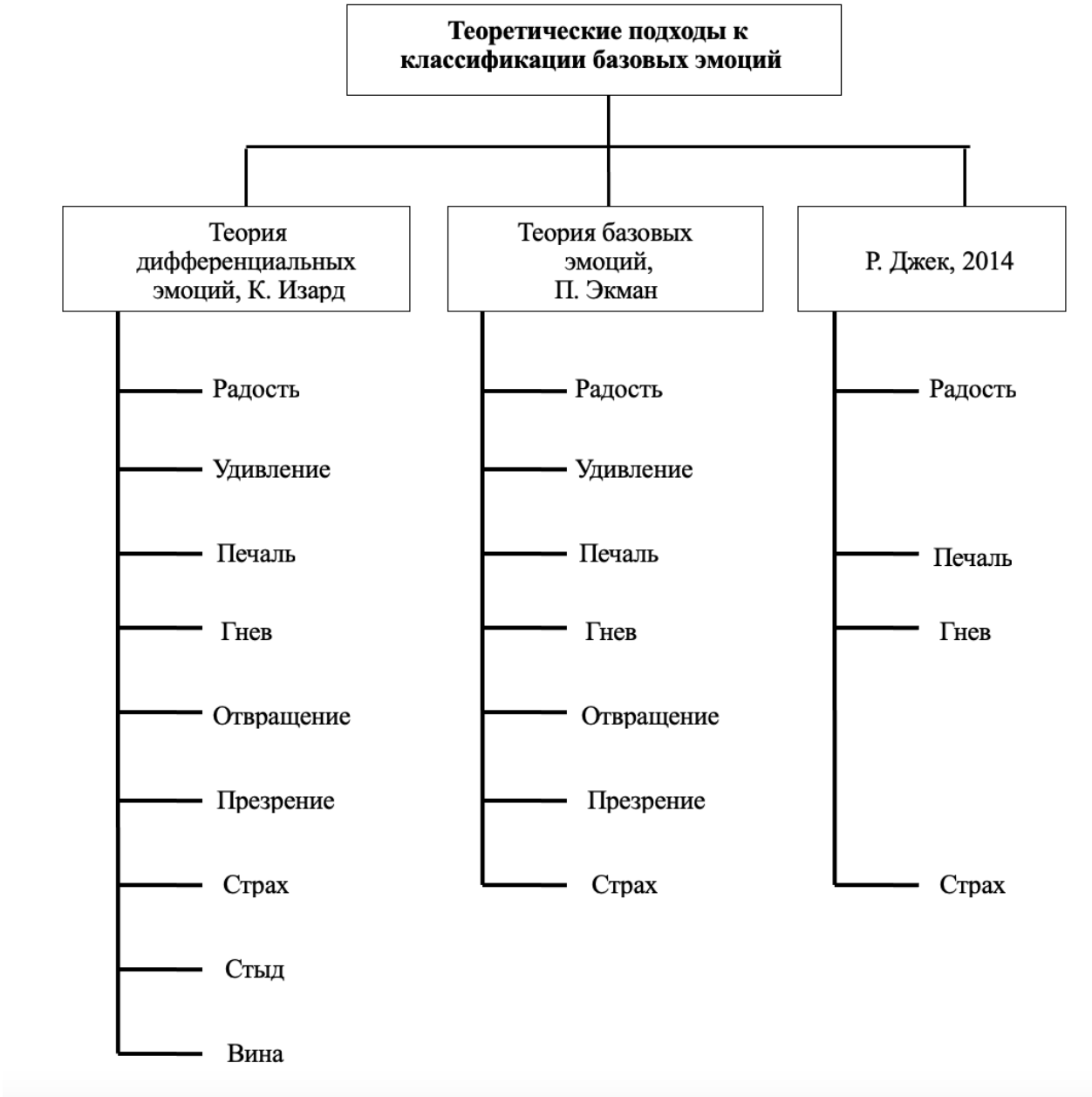


Рисунок А.1 – Теоретические подходы к классификации базовых эмоций

Приложение Б

Фрагмент листинга программного кода обучения нейронной сети по изображению

```
if __name__ == '__main__':
    num_labels = 7
    batch_size = 64
    epochs = 150
    width, height = 48, 48


    x = np.load('./fdataX.npy')
    y = np.load('./flabels.npy')
    x -= np.mean(x, axis=0)
    x /= np.std(x, axis=0)
    # разделяем на обучающую, тестовую и валидирующую выборки
    X_train, X_test, y_train, y_test = train_test_split(x, y, test_size=0.1, random_state=42)
    X_train, X_valid, y_train, y_valid = train_test_split(X_train, y_train, test_size=1/9, random_state=41)
    # сохраняем тестовые сэмплы
    np.save('modXtest', X_test)
    np.save('modytest', y_test)
    # модель CNN
    model = Sequential()
    model.add(Conv2D(64, kernel_size=(3, 3), activation='relu',  padding = 'same', input_shape=(width, height, 1)))
    model.add(MaxPooling2D(pool_size=(2, 2)))
    #model.add(Dropout(0.5))
```

```
model.add(Conv2D(128, kernel_size=(3, 3), activation='relu', padding='same'))
model.add(MaxPooling2D(pool_size=(2, 2)))
#model.add(Dropout(0.5))
model.add(Conv2D(128, kernel_size=(3, 3), activation='relu', padding='same'))
model.add(MaxPooling2D(pool_size=(2, 2)))
#model.add(Dropout(0.5))
#model.add(Conv2D(64, kernel_size=(3, 3), activation='relu', padding='same'))
#model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Flatten())
model.add(Dropout(0.5))
model.add(Dense(256, activation='relu'))
model.add(Dense(num_labels, activation='softmax'))
model.summary()
# компилируем модель
model.compile(loss=categorical_crossentropy,
            optimizer=RMSprop(learning_rate = 0.001),
            metrics=['accuracy'])
# обучаем модель
history = model.fit(np.array(X_train), np.array(y_train),
          batch_size=batch_size,
          epochs=epochs,
          verbose=1,
          validation_data=(np.array(X_valid), np.array(y_valid)),
          shuffle=True)
# Завершили обучение, тренировка на valid наборах
```

```
history = model.fit(np.array(X_valid), np.array(y_valid),
        batch_size=batch_size,
        epochs=30,
        verbose=1,
        validation_data=(np.array(X_train), np.array(y_train)),
        shuffle=True)
# Проверяем точность модели на тестовой выборке
test_loss, test_acc = model.evaluate(np.array(X_test), np.array(y_test))
print('test_acc:', test_acc)
# сохраняем для дальнейшего использования
fer_json = model.to_json()

with open("fer.json", "w") as json_file:
    json_file.write(fer_json)

model.save_weights("fer.h5")
print("Saved model to disk")
```

Приложение В

Результаты эксперимента по классификации эмоций изображению лица



Рисунок В.1 – Тестовые изображения

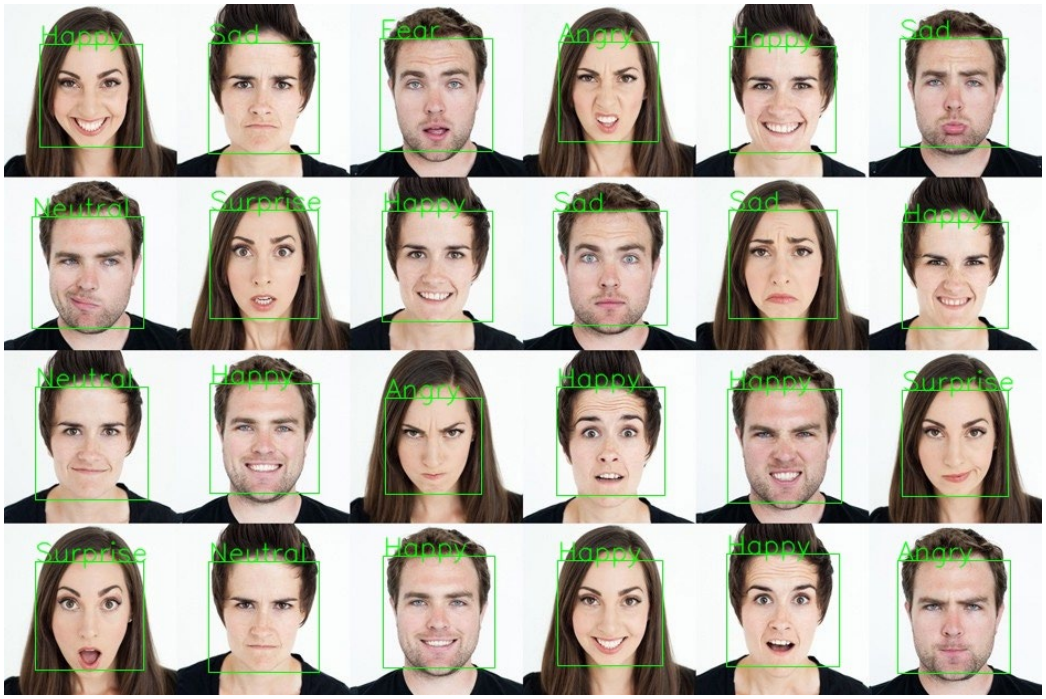


Рисунок В.2 – Результат классификации эмоции на нейронной сети по изображению лица

Приложение Г

Фрагмент листинга программного кода предобработки текстового датасета

```
import nltk
nltk.download('stopwords')
nltk.download('punkt')
from nltk.corpus import stopwords
from nltk.tokenize import word_tokenize
import re
import tqdm


def preprocess_text(texts):
    stop_words = set(stopwords.words('english'))
    regex = re.compile('[^a-z A-Z]')
    preprocess_texts = []
    for i in  tqdm.tqdm(range(len(texts))):
        text = texts[i].lower()
        text = regex.sub(' ', text)
        word_tokens = word_tokenize(text)
        filtered_sentence = [w for w in word_tokens if not w in stop_words]
        preprocess_texts.append( ' '.join(filtered_sentence))
    return preprocess_texts


from nltk.stem.lancaster import LancasterStemmer


def stemming_texts(texts):
  st = LancasterStemmer()
```

```
  stem_text = []
  for text in tqdm.tqdm(texts):
    word_tokens = word_tokenize(text)
    stem_text.append(' '.join([st.stem(word) for word in word_tokens]))
  return stem_text

from sklearn.feature_extraction.text import CountVectorizer

vectorizer = CountVectorizer()
vectorizer.fit()
vectorizer.get_feature_names_out()

def bow(vectorizer, train, test):
  train_bow = vectorizer.fit_transform(train)
  test_bow = vectorizer.transform(test)
  return train_bow, test_bow

X_train_bow, X_test_bow = bow(vectorizer,
                              text_train.data,
                              text_test.data)

X_train_bow_preprocess, X_test_bow_preprocess = bow(vectorizer,
                                                    text_train.preprocess_data,
                                                    text_test.preprocess_data)
```

```
X_train_bow_stem, X_test_bow_stem = bow(vectorizer,
                                        text_train.data_stemming,
                                        text_test.data_stemming)

from sklearn.feature_extraction.text import TfidfVectorizer

vectorizer_tf_idf = TfidfVectorizer()

X_train_tfidf, X_test_tfidf = bow(vectorizer_tf_idf,
                                  text_train.data,
                                  text_test.data)

X_train_tfidf_preprocess, X_test_tfidf_preprocess = bow(vectorizer_tf_idf,
                                                        text_train.preprocess_data,
                                                        text_test.preprocess_data)

X_train_tfidf_stem, X_test_tfidf_stem = bow(vectorizer_tf_idf,
                                            text_train.data_stemming,
                                            text_test.data_stemming)

vectorizer_ngram = TfidfVectorizer(ngram_range=(1,2))

X_train_ngram_stem, X_test_ngram_stem = bow(vectorizer_ngram,
                                            text_train.data_stemming,
                                            text_test.data_stemming)
```