

Пояснительная записка

к итоговому проекту на тему:

«Детекция эмоций человека»

Автор: Александр Викторович Дубовик
Группа: DS-72

Оглавление

1. Постановка задачи	3
Описание исходной задачи работы	3
Актуальность задачи	3
Метрики качества	4
2. Анализ данных.....	6
Исходные данные:.....	6
Описание и обоснование алгоритмов и техник, применяемых для решения задачи, их требования к исходным данным.	8
Анализ аналогичных решений	8
3. Методика решения.....	9
Варианты исследований модели CNN, применяемых в работе:	9
Предварительная обработка	10
Обучение модели с учетом весов класса	10
Обучение модели с процессом переобучения данных с использованием метода Synthetic Minority Over-sampling Technique (SMOTE) для решения проблемы дисбаланса классов в наборе данных.....	11
Определение и обучение модели:	12
Функции активации ReLu и ELU.	14
Обнаружение лиц на видео с веб-камеры и определение эмоций на этих лицах с помощью предварительно обученной модели.	16
Интеграция обнаружения лиц и определения эмоций	16
4. Результаты	18
5. Выводы и заключение.....	22
6. Список источников	24

1. Постановка задачи

Описание исходной задачи работы

Исходной задачей работы является реализация искусственной нейронной сети с использованием компьютерного зрения для классификации и детекции объектов. Вторая часть — это определение на видео в режиме реального времени эмоции субъектов. Объекты детекции лица людей, классификация — определение 7 эмоций. Архитектура модели – CNN.

Актуальность задачи

Распознавание (детекция) эмоций и возможность предугадывания тех или иных эмоций человека, в настоящее время очень востребовано. В медицине - для тех людей, у которых есть эмоциональное расстройство и их трудно идентифицировать, разрабатываются целые комплексы распознавания эмоций, особенно для людей с низкой эмоциональной составляющей, что помогает правильно определить лечение. Так же очень близко к медицине данная система может помочь специалистам в области психологии, особенно в детской психологии, где эмоциональная составляющая очень существенна, но не всегда диагностируется, особенно если сессии проходят в удаленном формате. А для специалистов важно оценивать и не упускать любое проявление эмоций.

Как побочный продукт, так как во многих организациях стоят камеры, которые позволяют оплатить что-то с помощью лица, пройти идентификацию, чтобы получить доступ в помещение и т. п. Распознавание эмоций, например в торговых точках или учреждениях, предоставляющих другие услуги физическим лицам, могут с помощью детекции получать эмоциональный отклик клиента, для эффективного взаимодействия.

В рамках безопасности в аэропортах, вокзалах, торговых центрах и прочих местах массового скопления людей - определение эмоционально не устойчивых и возможных нарушителей (как пример возбуждение, бегающие глаза, чувство отвращения, агрессии или презрения). Так же может помочь людям в стрессовой ситуации, например: человек с Альцгеймером потерялся, но пока не заявлен в поиск, но по детекции его поведения можно уточнить не нужна ли ему помощь, тоже самое с детьми.

Коммерческое использование для HR: Собеседование очень сложный процесс и требует от сотрудника, который проводит собеседование навыка определения эмоций кандидата на тот или иной вопрос, а система детекции эмоций может стать отличным помощником в данном вопросе. А учитывая, что в ближайшем будущем, собеседование будет сначала проводить робот или ИИ, роботу точно нужно будет понимать, что думает и как реагирует человек на вопросы.

Метрики качества

F1-мера (F1-score): является средним гармоническим точности и полноты, что позволяет учесть и ошибки первого рода (пропущенные положительные), и ошибки второго рода (ложноположительные).

F1-мера для многоклассовой классификации. Три вида усреднения

Эти метрики являются ключевыми при оценке качества моделей машинного обучения, особенно в задачах бинарной и многоклассовой классификации. Они дают возможность оценить модель с разных сторон, учитывая не только общую точность предсказаний, но и специфичность и полноту обнаружения положительных примеров.

Для вычисления F-меры (и других) метрик в рамках многоклассовой классификации используется подход «один против всех»: каждый класс ровно один раз становится «положительным», а остальные — отрицательным (пример вычисления изображён на матрице).

Таким образом, в зависимости от этапа вычисления, на котором производится усреднение, можно вычислить micro-average, macro-average и average F-меры (логика вычисления изображена на схеме справа). Микро- и макро-:

$$F1 - score = \frac{2 * Precision * Recall}{Precision + Recall}$$

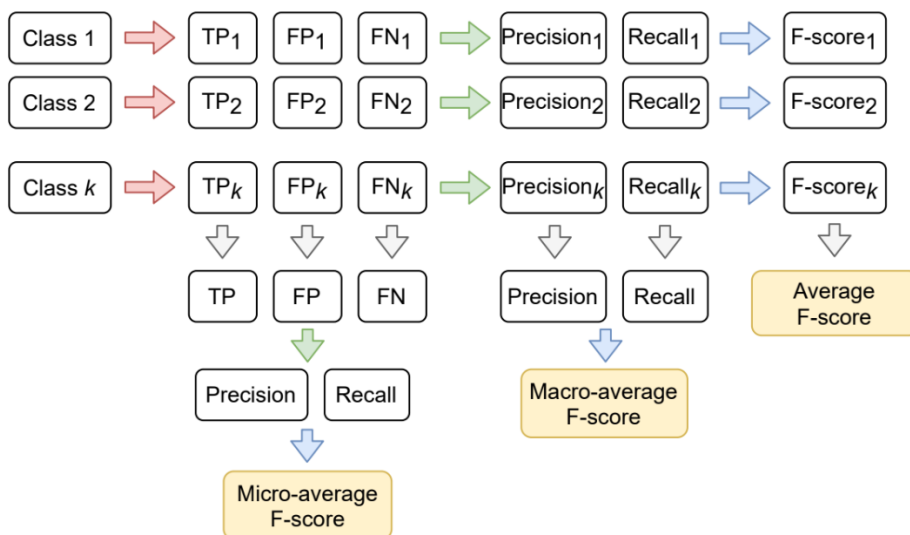
, где для micro-average precision и recall вычислены из усреднённых TP, FP, FN;

для macro-average precision и recall вычислены из усреднённых precision_i, recall_i;

Усреднённая:

$$F = \frac{1}{k} \sum_{i=0}^k F_1 score_i$$

где i — индекс класса, а k — число классов.



Для справки:

Точность (Precision): Процент правильно классифицированных положительных примеров относительно всех примеров, которые были классифицированы как положительные.

Формула для точности:

$$Precision = \frac{TP}{TP + FP}$$

Полнота (Recall): Процент правильно классифицированных положительных примеров относительно всех истинно положительных примеров.

Формула для полноты:

$$Recall = \frac{TP}{TP + FN}$$

, где

TP (True Positives) — количество правильно классифицированных положительных примеров.

TN (True Negatives) — количество правильно классифицированных отрицательных примеров.

FP (False Positives) — количество неправильно классифицированных положительных примеров.

FN (False Negatives) — количество неправильно классифицированных отрицательных примеров.

2. Анализ данных

Исходные данные:

Датасет: «FER2013»

<https://www.kaggle.com/datasets/ashishpatel26/facial-expression-recognitionferchallenge/data>

Количество изображений: 35887 шт.

Количество изображений в обучающей выборке: 28709 шт.

Количество изображений в тестовой выборке: 7178 шт.

Количество классов: 7

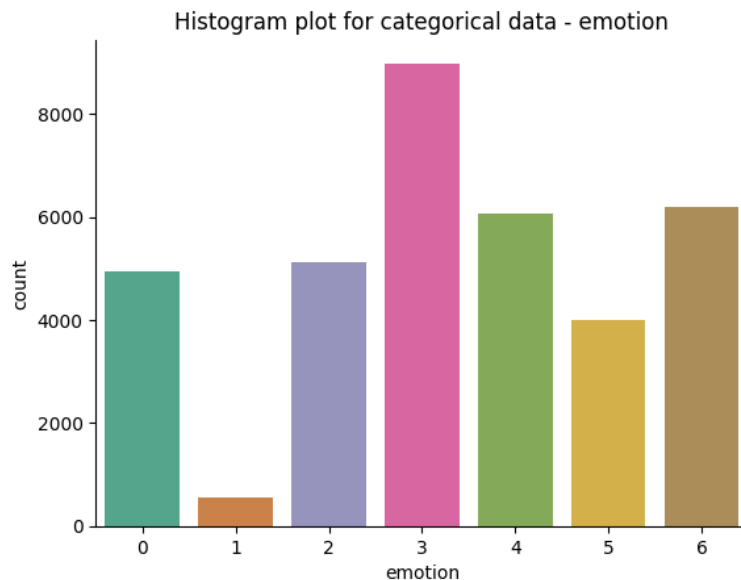
Классы: anger, disgust, fear, happiness, sadness, surprise, neutral

Описание: Данные состоят из изображений лиц размером 48x48 пикселей в оттенках серого. Столбец «**emotion**» содержит числовой код от 0 до 6 включительно для эмоции, присутствующей на изображении. Столбец «**pixels**» содержит строку, заключенную в кавычки для каждого изображения. Содержимое этой строки представляет собой значения пикселей, разделенные пробелами, в порядке возрастания строк.



Особенности: Лица были автоматически зарегистрированы таким образом, чтобы лицо располагалось более или менее по центру и занимало примерно одинаковое пространство на каждом изображении.

В датасете несбалансированное распределение классов, т. е. классы, которые мы пытаемся предсказать, представлены неравномерно.



```
emotion
happiness (3) 8989
neutral (6) 6198
sadness (4) 6077
fear (2) 5121
anger (0) 4953
surprise (5) 4002
disgust (1) 547
```

Код для анализа данных:

https://github.com/dubovik-alexander/Netology-DS/blob/main/Diploma/final/py_files/data_analysis.ipynb

Описание и обоснование алгоритмов и техник, применяемых для решения задачи, их требования к исходным данным.

Современные модели часто требуют больших ресурсов. В моей работе детекция эмоций подразумевает использование классификатора эмоциональных выражений в реальном времени, т. е. Определение эмоциональных выражений с поддержкой веб-камеры. Методами глубокого обучения, включающих обнаружение объектов, на основе сверточных нейронных сетей (CNN) с предварительно подготовленной архитектурой глубокого обучения. Результатом работы является программа, которая получает входное изображение и на выходе отображает, в качестве выходных данных, на обнаруженном лице надпись с названием эмоции.

Анализ аналогичных решений

Исследование возможности детектирования эмоций — это перспективная область компьютерного зрения, которая развивается во всем мире. Существуют подходы и методы, используемые для решения данной задачи:

DenseNet169 Transfer Learning - В работе решается задача распознавания эмоций человека для применения в психологии, маркетинге и интерактивных технологиях. Авторы предлагают использовать нейронную сеть DenseNet169 с механизмом Transfer Learning, обученную на наборе данных AffectNet, для локализации и классификации эмоций на изображениях лиц. В сочетании с предварительно обученными моделями, в которых содержится информация о различных эмоциональных состояниях, система способна предоставлять точные данные о типе и интенсивности эмоции на лице человека.

AlexNet model - Другой подход заключается в использовании модификации AlexNet для локализации лиц (нахождение их ограничивающих прямоугольников) и последующего анализа выражений лица для распознавания эмоций. Исходный код и модели доступны в рамках исследовательских работ по распознаванию эмоций. AlexNet — это архитектура нейронной сети, которая была разработана для задач классификации изображений и последующего выявления важных признаков в данных. AlexNet, описанная в работах по глубокому обучению, использует слои свертки для извлечения важных признаков из изображений.

3. Методика решения

Варианты исследований модели CNN, применяемых в работе:

Модель CNN, активация ReLu, активация ELU

1. As is (как есть). Т.е. модель обучалась на данных без весов класса (class_weight) CNN_1_asis.ipynb
 - https://github.com/dubovik-alexander/Netology-DS/blob/main/Diploma/final/py_files/CNN_1_asis.ipynb - активация «ReLU»
 - https://github.com/dubovik-alexander/Netology-DS/blob/main/Diploma/final/py_files/CNN_1_asis_elu.ipynb - активация «ELU»
2. Обучение модели с учетом весов класса.
 - https://github.com/dubovik-alexander/Netology-DS/blob/main/Diploma/final/py_files/CNN_2_weight_colab.ipynb - активация «ReLU»
 - https://github.com/dubovik-alexander/Netology-DS/blob/main/Diploma/final/py_files/CNN_2_weight_elu_colab.ipynb - активация «ELU»
3. Обучение модели с процессом переобучения данных с использованием метода Synthetic Minority Over-sampling Technique (SMOTE) для решения проблемы дисбаланса классов в наборе данных.
 - https://github.com/dubovik-alexander/Netology-DS/blob/main/Diploma/final/py_files/CNN_4_weight_oversampling_relu_colab.ipynb - активация «ReLU»
 - https://github.com/dubovik-alexander/Netology-DS/blob/main/Diploma/final/py_files/CNN_4_weight_oversampling_elu_colab.ipynb - активация «ELU»
4. Проводились исследования влияние функции активации ReLu и ELU на точность обучения модели.

Предварительная обработка

Для предварительной обработки и обучения сверточной нейронной сети (CNN) для распознавания эмоций на основе изображений. Данные должны быть подготовлены, следующим образом:

- a) Кодирование меток и one-hot кодирование: используется `LabelEncoder` из `scikit-learn` для преобразования категориальных меток (эмоций) в числовую форму. Затем применяется функция `to_categorical` для преобразования этих числовых меток в векторы, закодированные в формате one-hot, для задач классификации, где выходной слой использует активацию `softmax`¹.
- b) Создание словаря отображения: создаётся словарь (`le_name_mapping`), который отображает оригинальные метки классов обратно к их закодированным значениям. Для интерпретации прогнозов модели позже.
- c) Преобразование данных изображений в трехмерный массив NumPy: Столбец `pixels` `DataFrame`, предполагаемый содержать строковые представления значений пикселей, преобразуется в трехмерный массив NumPy (`img_array`). Каждая строка представляет собой изображение, а пиксели разделяются на отдельные значения, преобразуются в числа с плавающей точкой и формируются в массив размером 48x48x1.
- d) Стекинг изображений в один трехмерный массив: Используя `np.stack`, 2D-массивы, представляющие отдельные изображения, стекаются вдоль новой оси, чтобы образовать единственный трехмерный массив. Это общий шаг предварительной обработки для подготовки наборов данных изображений к машинному обучению.
- e) Разделение данных на обучающую и валидационную выборки: Набор данных разбивается на обучающую и валидационную выборки с использованием `train_test_split`. Параметр `stratify` гарантирует сохранение распределения классов в обоих подмножествах.
- f) Нормализация данных изображений: перед тем как передавать изображения в CNN, они нормализуются путем деления на 255. Это масштабирует значения пикселей в диапазон `[0, 1]`, что помогает модели быстрее сходиться во время обучения.

Обучение модели с учетом весов класса

Вычисление весов классов: вычисляются веса классов с использованием балансирующего подхода, где классы с меньшим количеством примеров получают больший вес. Это помогает улучшить производительность модели на несбалансированных наборах данных. Которые потом используем в обучении модели (один из экспериментов).

¹Функция `Softmax` используется для преобразования вектора значений в вероятностное распределение, которое суммируется до 1. Она особенно полезна в многоклассовой классификации, где необходимо определить вероятности для каждого класса.

Для вычисления используем исходный датасет. Далее выполняются следующие шаги:

- g) Данные разбиваются на обучающую и тестовую выборки.
- h) Метки классов преобразуются в категориальный формат.
- i) Восстанавливается исходный целевой массив: после преобразования меток классов обратно восстанавливают их исходный целевой массив для вычисления весов классов.
- j) Вычисление веса классов.
- k) Конвертация веса в словарь: веса классов конвертируются в словарь для удобства использования в процессе обучения модели.

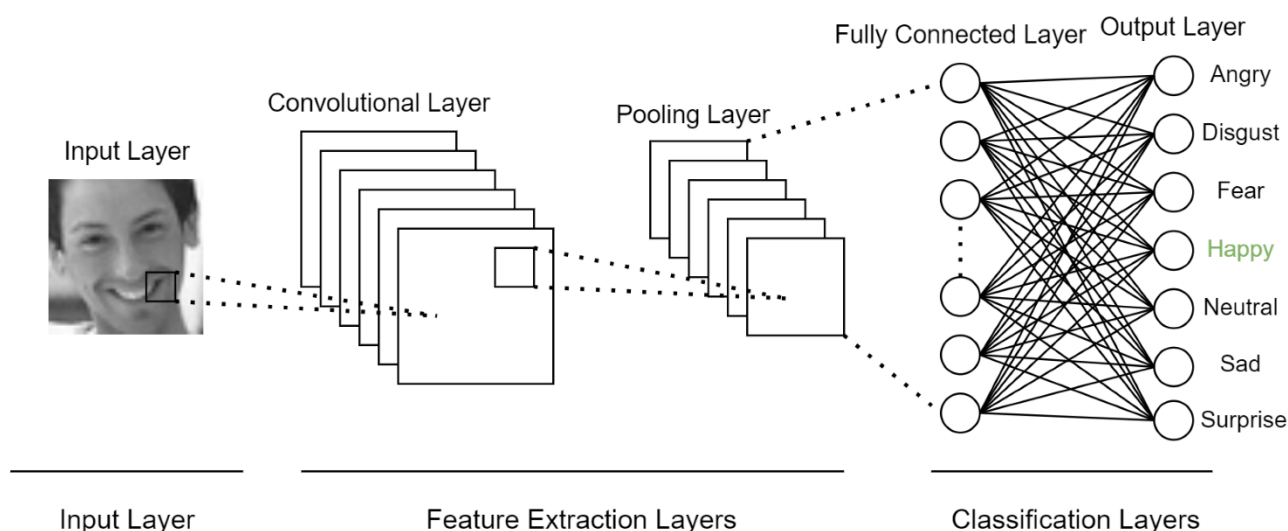
Обучение модели с процессом переобучения данных с использованием метода Synthetic Minority Over-sampling Technique (SMOTE) для решения проблемы дисбаланса классов в наборе данных.

- a) Преобразование изображений в векторы: сначала все изображения в `X_train` "плоско" преобразуются в одномерные векторы длины $48 \times 48 = 2304$. Это делается для упрощения структуры данных перед применением техник SMOTE.
- b) Проверка совпадения количества образцов: проверяется, что количество образцов в `X_train_flat` совпадает с количеством образцов в `y_train`. Если это не так, то генерируется исключение `ValueError`, указывающее на несоответствие.
- c) Применение SMOTE для увеличения количества образцов: Метод SMOTE используется для увеличения количества образцов в миноритарных классах путем синтеза новых данных. Параметры метода задаются следующим образом:
 - `sampling_strategy='auto'`: Автоматическое определение стратегии выборки, которая может быть полезна для уравнивания классов.
 - `random_state=42`: Установка начального состояния генератора случайных чисел для воспроизводимости результатов.
 - `k_neighbors=6`: Количество соседей для выбора при создании нового образца.
 - `n_jobs=1`: Количество параллельных процессов для выполнения SMOTE.
- d) Изменение формы данных после SMOTE: после применения SMOTE форма данных `X_resampled` изменяется обратно к форме, соответствующей исходным изображениям ($48 \times 48 \times 1$), что необходимо для последующего использования в нейронной сети.
- e) Приведение типов данных для `y_resampled`: Целевые значения `y_resampled` приводятся к типу `np.float32`, для обеспечения единообразия типов данных.

В результате этих операций получают обновленные данные $X_{\text{resampled}}$ и $y_{\text{resampled}}$, которые были использованы для обучения нейронной сети, увеличенные за счет метода SMOTE и приведенные к нужному виду.

Определение и обучение модели:

Модель CNN структурирована таким образом, чтобы эффективно обрабатывать данные изображений, извлекать иерархические признаки и классифицировать изображения по одной из семи категорий. Использование сверточных слоев для извлечения признаков, слоев максимального пула (max-pooling) для уменьшения размерности и плотных слоев (Dense) для классификации, а также методов регуляризации, таких как исключение (Dropout) и пакетная нормализация (batch normalization), делает эту архитектуру надежной для задач на основе изображений.



Разбивка ключевых компонентов и их ролей в модели:

а) Входной слой

- Conv2D: инициализирует модель с помощью сверточного слоя, который использует 32 фильтра размером 3x3, с активацией «ReLU» / «ELU», заполнением «same» и инициализатором «He normal». Этот слой отвечает за извлечение исходных функций из входных изображений.
- BatchNormalization: нормализует активации предыдущего слоя в пакетном измерении, повышая скорость и стабильность обучения.
- Conv2D: добавляет еще один сверточный слой с той же конфигурацией, что и первый, что дополнительно совершенствует извлечение признаков.

- BatchNormalization: снова применяет нормализацию после второго сверточного слоя.

f) Промежуточные слои

- MaxPooling2D: уменьшает пространственные размеры (высоту и ширину) карт объектов, беря максимальное значение в окне размером 2x2, эффективно уменьшая входные данные. Это помогает снизить вычислительную сложность и контролировать переобучение.
- Dropout: случайным образом устанавливает для части (0,1) входных единиц значение 0 при каждом обновлении во время обучения, что помогает предотвратить переобучение.
- Шаблон Conv2D -> BatchNormalization -> Conv2D -> BatchNormalization -> MaxPooling2D -> Dropout повторяется три раза, каждый раз удваивая количество фильтров (с 32 до 64, до 128, до 256 до 512) для постепенного захвата более сложных функций.

g) Выходные слои

- После серии сверточных слоев, слоев объединения и исключения модель выравнивает выходные данные и добавляет плотные слои для выполнения классификации. Последний слой Dense использует функцию активации softmax, что указывает на то, что эта модель предназначена для задачи многоклассовой классификации с 7 классами.

h) Компиляция модели: Модель компилируется с оптимизатором Adam, функцией потерь категориального кросс-энтропии и точностью, точностью и полнотой как оценочными метриками.

i) Настройка обратных вызовов: Настройка раннего останова, снижения скорости обучения и сохранения модели для улучшения процесса обучения. Ранний останов устанавливает контроль над валидационной точностью и останавливает обучение, если оно не улучшается за определенный порог. Снижение скорости обучения уменьшает скорость обучения, когда валидационная точность стагнирует. Сохранение модели сохраняет веса модели на эпоху с наименьшей валидационной потерей.

j) Обучение модели: наконец, модель обучается на подготовленном наборе данных с указанным количеством эпох и размером партии. История обучения сохраняется для дальнейшего анализа.

Функции активации ReLu и ELU.

ReLU (Rectified Linear Unit)

ReLU (Rectified Linear Unit) - это нелинейная функция активации, которая широко используется в глубоком обучении. Она преобразует входное значение в значение от 0 до положительной бесконечности. Если входное значение меньше или равно нулю, то ReLU выдает ноль, в противном случае - входное значение.

Математически ReLU определяется следующим образом:

$$\text{ReLU}(x) = \max(0, x)$$

где \max - функция, возвращающая максимальное значение из двух.

Графически ReLU выглядит как линейная функция с нулевым отсечением на оси абсцисс в точке 0. Это значит, что функция имеет постоянный наклон во всех точках, кроме точки 0, где происходит отсечение.



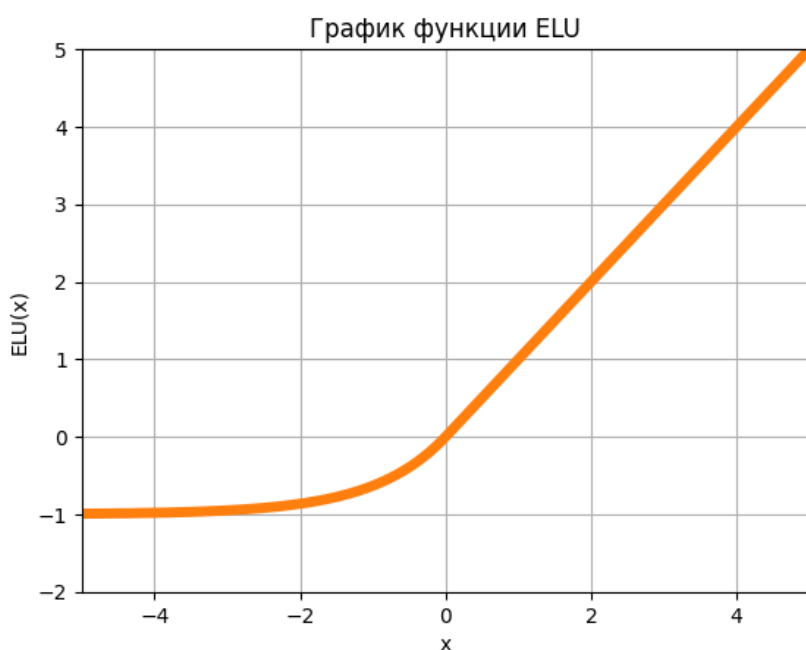
ReLU имеет несколько преимуществ по сравнению со сигмоидной функцией активации. Во-первых, ReLU более вычислительно эффективна, поскольку она является простой и быстрой операцией, которая не требует вычисления экспоненты. Во-вторых, ReLU решает проблему затухания градиента, так как она не вызывает затухания градиента при обратном распространении ошибки, как это происходит в случае с сигмоидной функцией активации.

Однако, ReLU имеет некоторые недостатки. Во-первых, при использовании ReLU, некоторые нейроны могут "умереть" (dead neurons), т.е. они могут получить отрицательное значение и

оставаться неактивными на всем протяжении обучения. Во-вторых, ReLU несимметрична относительно нуля, поэтому может возникнуть проблема "расслоения" (clustering), когда нейроны могут выдавать только положительные значения. Для решения этих проблем могут быть использованы другие функции активации, такие как Leaky ReLU или ELU.

ELU (Exponential Linear Unit)

ELU (Exponential Linear Unit) - это функция активации, которая была предложена в 2015 году в статье "Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs)". Она представляет собой измененную версию ReLU (Rectified Linear Unit), которая помогает ускорить обучение глубоких нейронных сетей и справляется с проблемой "мертвых нейронов" (dead neurons).



ELU определяется следующим образом:

$$ELU(x) = \begin{cases} x, & \text{if } x > 0 \\ \alpha(e^x - 1), & \text{if } x \leq 0 \end{cases}$$

где α (alpha) - это параметр, который может быть установлен в значение 1 по умолчанию.

ELU работает так же, как и ReLU, возвращая исходное значение входа, если он больше нуля. Однако, если значение входа меньше или равно нулю, то ELU использует экспоненциальную функцию, чтобы получить значение, которое ближе к нулю, чем значение, возвращаемое ReLU. Это позволяет избежать "мертвых нейронов" и ускорить обучение глубоких нейронных сетей.

Кроме того, ELU имеет свойство гладкости, которое так же помогает избежать проблемы "взрывающегося градиента" (exploding gradient), которая может возникать при использовании других функций активации, таких как ReLU. Это делает ELU более стабильной и более эффективной функцией активации для обучения глубоких нейронных сетей.

Обнаружение лиц на видео с веб-камеры и определение эмоций на этих лицах с помощью предварительно обученной модели.

Обнаружение эмоций

Функция `detect_emotion` принимает изображение лица в качестве входных данных и выполняет следующие шаги:

- a) Изменение размера изображения до 48x48 пикселей для ускорения обработки и соответствия требованиям модели.
- b) Конвертация изображения из цветного в оттенки серого, так как модель работает только с градационными изображениями.
- c) Добавление дополнительных измерений к массиву изображения для соответствия формату, который ожидает модель.
np.expand_dims(image, axis=-1): Этот шаг добавляет дополнительное измерение в конец массива изображения (axis=-1), добавление канала цвета (в нашем случае 1 канал — оттенки серого)
np.expand_dims(image, axis=0): Этот шаг добавляет дополнительное измерение в начало массива изображения (axis=0), для обеспечения совместимости формы входных данных с тем, что ожидает модель.
- d) Прогнозирование эмоции с помощью предобученной модели (`model.predict`). Модель возвращает вероятности для каждой из семи возможных эмоций.
- e) Определение индекса эмоции, которая имеет наибольшую вероятность (`np.argmax`).
- f) Возвращение названия эмоции из списка `emotion_labels`, используя найденный индекс.

На выходе функция возвращает название эмоции, распознанной на переданном изображении.

Интеграция обнаружения лиц и определения эмоций

- a) Инициализация классификатора Haar cascades для обнаружения лиц.
- b) Открытие видеопотока с веб-камеры.
- c) Создание директории для сохранения изображений лиц с распознанными эмоциями, если она еще не существует.

d) Цикл чтения кадров с веб-камеры:

- Конвертация каждого кадра в оттенки серого.
- Обнаружение лиц на кадре.
- Если лицо обнаружено, извлекается изображение лица и определяется его эмоция с помощью функции `detect_emotion`.
- Если эмоция отличается от предыдущей, на изображении лица рисуется текст с названием эмоции, а также изображение сохраняется в директорию.
- Отображение исходного кадра с возможной отметкой эмоции на лице.
- Завершение цикла при нажатии клавиши 'q'.

После завершения записи видео, код загружает все сохраненные изображения лиц с эмоциями, изменяет их размер до 64x64 пикселей, преобразует в массив NumPy и отображает сетку случайно выбранных изображений с использованием функции `plot_image_grid`.

Таким образом, этот код позволяет автоматически обнаруживать лица на видео с веб-камеры, определять эмоции на этих лицах и сохранять результаты в виде изображений.

4. Результаты

Результатами исследования стали следующие:

Оценки модели классификации:

1. As is активация «ReLU»

	precision	recall	f1-score	support
0	0.59	0.47	0.52	991
1	0.76	0.44	0.56	109
2	0.56	0.30	0.39	1024
3	0.78	0.83	0.81	1798
4	0.44	0.59	0.51	1216
5	0.74	0.76	0.75	800
6	0.53	0.60	0.57	1240
micro avg	0.61	0.61	0.61	7178
macro avg	0.63	0.57	0.59	7178
weighted avg	0.62	0.61	0.61	7178
samples avg	0.61	0.61	0.61	7178

2. As is активация «ELU»

	precision	recall	f1-score	support
0	0.53	0.51	0.52	991
1	0.72	0.52	0.61	109
2	0.53	0.41	0.46	1024
3	0.82	0.79	0.80	1798
4	0.47	0.51	0.49	1216
5	0.74	0.77	0.76	800
6	0.52	0.60	0.56	1240
micro avg	0.61	0.61	0.61	7178
macro avg	0.62	0.59	0.60	7178
weighted avg	0.62	0.61	0.61	7178
samples avg	0.61	0.61	0.61	7178

3. Веса с активацией «ReLU»

	precision	recall	f1-score	support
0	0.52	0.52	0.52	991
1	0.51	0.62	0.56	109
2	0.52	0.41	0.46	1024
3	0.80	0.79	0.79	1798
4	0.48	0.44	0.46	1216
5	0.74	0.77	0.75	800
6	0.50	0.60	0.55	1240
micro avg	0.60	0.60	0.60	7178
macro avg	0.58	0.60	0.59	7178
weighted avg	0.60	0.60	0.60	7178
samples avg	0.60	0.60	0.60	7178

4. Веса с активацией «ELU»

	precision	recall	f1-score	support
0	0.56	0.49	0.52	991
1	0.65	0.57	0.60	109
2	0.48	0.48	0.48	1024
3	0.79	0.79	0.79	1798
4	0.51	0.42	0.46	1216
5	0.77	0.76	0.76	800
6	0.51	0.65	0.57	1240
micro avg	0.61	0.61	0.61	7178
macro avg	0.61	0.59	0.60	7178
weighted avg	0.61	0.61	0.61	7178
samples avg	0.61	0.61	0.61	7178

5. SMOTE с активацией «ReLU»

	precision	recall	f1-score	support
0	0.54	0.49	0.51	991
1	0.83	0.58	0.68	109
2	0.54	0.46	0.50	1024
3	0.78	0.82	0.80	1798
4	0.49	0.47	0.48	1216
5	0.74	0.78	0.76	800
6	0.52	0.61	0.56	1240
micro avg	0.62	0.62	0.62	7178
macro avg	0.64	0.60	0.61	7178
weighted avg	0.62	0.62	0.62	7178
samples avg	0.62	0.62	0.62	7178

6. SMOTE с активацией «ELU»

	precision	recall	f1-score	support
0	0.56	0.50	0.53	991
1	0.84	0.56	0.67	109
2	0.55	0.47	0.50	1024
3	0.77	0.82	0.79	1798
4	0.50	0.49	0.50	1216
5	0.76	0.76	0.76	800
6	0.52	0.62	0.57	1240
micro avg	0.62	0.62	0.62	7178
macro avg	0.64	0.60	0.62	7178
weighted avg	0.62	0.62	0.62	7178
samples avg	0.62	0.62	0.62	7178

Пояснение:

Precision (точность): Это доля правильно классифицированных положительных примеров среди всех примеров, которые модель классифицировала как положительные. Например, для класса 0 точность равна 0.56, что означает, что 56% примеров, классифицированных моделью как принадлежащих классу 0, действительно принадлежат этому классу.

Recall (полнота): Это доля правильно классифицированных положительных примеров среди всех фактически положительных примеров. Например, для класса 0 полнота равна 0.50, что означает, что модель обнаружила 50% всех примеров, которые действительно принадлежат классу 0.

F1-score: Это гармоническое среднее между точностью и полнотой. Оно позволяет учесть оба этих параметра одновременно. Например, F1-оценка для класса 0 равна 0.53, что указывает на баланс между точностью и полнотой в классификации этого класса.

Support: Это количество примеров каждого класса в тестовой выборке. Например, у класса 0 991 примера.

Дополнительные строки:

micro avg: Средние значения всех метрик по всем классам, рассчитанные на уровне отдельных предсказаний.

macro avg: Средние значения всех метрик по всем классам, рассчитанные на уровне классов.

weighted avg: Средние значения всех метрик по всем классам, взвешенные по количеству примеров каждого класса.

samples avg: аналогично micro avg.

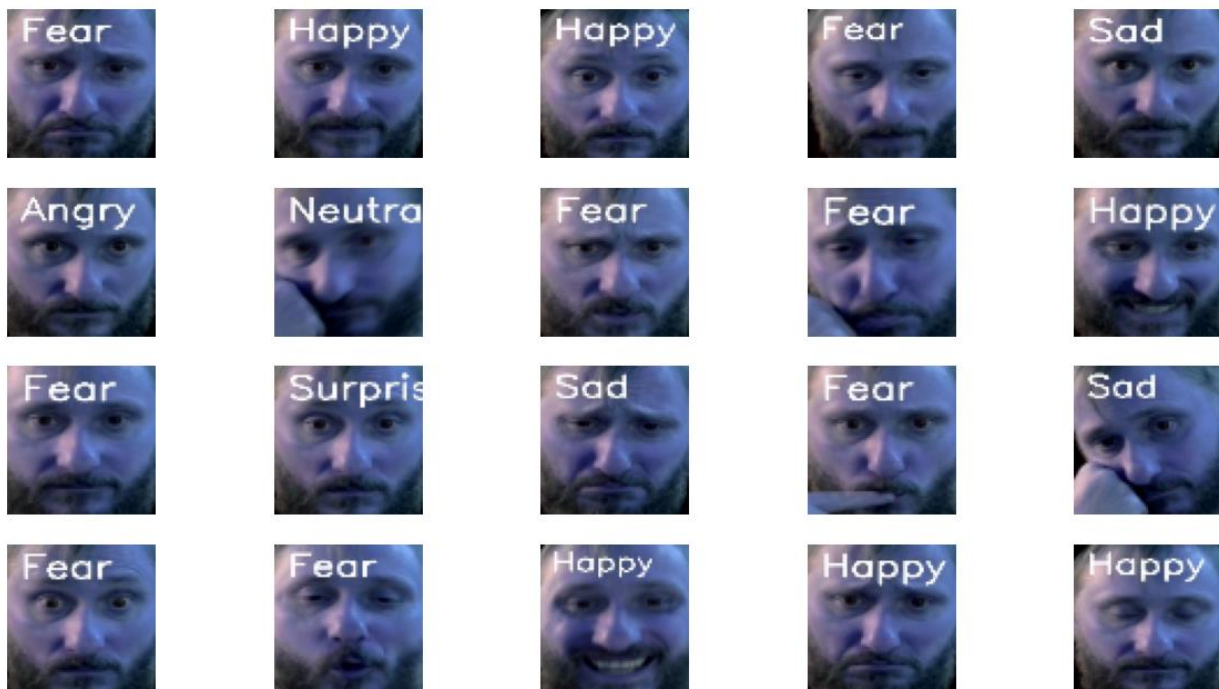
Итоги обучения модели:

Исходя из полученных выше результатов, обучение модели CNN с активацией «ELU» с процессом переобучения данных с использованием метода Synthetic Minority Over-sampling Technique (SMOTE) для решения проблемы дисбаланса классов в наборе данных показала себя лучше остальных, хоть и не значительно. Во всех обучении было использовано 50 эпох, лучшая модель достигла метрик micro avg - 0.62, macro avg - 0.62, weighted avg - 0.62. Метрики по отдельным классам получились соответственно: 0 – 0.53, 1 – 0.67, 2 – 0.50, 3 – 0.79, 4 – 0.50, 5 – 0.76, 6 – 0.57. Модель лучше обучилась распознавать те объекты, которые больше представлены в обучающей выборке.

Обнаружение лиц на видео с веб-камеры и определение эмоций на этих лицах с помощью предварительно обученной модели.

По итогам применения модели к видеопотоку, с большой долей вероятности, можно сделать вывод о том, что модель выучила основные паттерны, для определения эмоций. В тоже время остается еще много возможностей для улучшения модели. Например, можно заметить ложные срабатывания, когда модель определяет разные эмоции на одной и той же картинке. Возможно необходимо доработать не только модель, но и код по определению эмоции в режиме реального времени. Возможные варианты решения данной проблемы предложены далее.

Ознакомиться с результатами работы модели можно в репозитории:
https://github.com/dubovik-alexander/Netology-DS/tree/main/Diploma/final/py_files/face_images_final.



5. Выводы и заключение

В ходе проведенного исследования была получена модель детекции эмоций, а так же написан код для обнаружения лиц на видео с веб-камеры и определение эмоций на этих лицах с помощью предварительно обученной модели. Данная модель основана на архитектуре CNN. Целевые метрики достигли следующих результатов: micro avg - 0.62, macro avg - 0.62, weighted avg - 0.62. Метрики по отдельным классам: 0 – 0.53, 1 – 0.67, 2 – 0.50, 3 – 0.79, 4 – 0.50, 5 – 0.76, 6 – 0.57.

Для дальнейшей настройки модели рекомендуется:

- Увеличить размер обучающей выборки. 35887 изображений не достаточный объем для обучения нейронной сети;
- Добавление (выравнивание) данных по тем классам, которые имеют самый маленький набор данных, физическим методом, т.е. добавление изображений.
- Проводить обучение на изображениях с более высоким разрешением;
- Разнообразить изображения в обучающей выборке, добавить объекты с разными масштабами, относительно размеров изображения;
- Обучить модель на большем количестве эпох;
- Использование предварительно обученных моделей: таких как VGG19, ResNet, MobileNet или EfficientNet, которые уже были обучены на больших наборах данных ImageNet. Эти модели могут служить хорошим стартом и сократить время обучения.
- Data Augmentation: применение техник увеличения объема данных, таких как повороты, масштабирование, перевороты и другие трансформации, чтобы увеличить разнообразие обучающих примеров и сделать модель более устойчивой к изменениям.
- Fine-tuning: после использования предварительно обученной модели, можно продолжить обучение на специальном наборе данных для задачи определения эмоций. Это позволяет модели адаптироваться к новым данным без полного обучения с нуля.
- Ensemble Methods: использование комбинации нескольких моделей для повышения точности прогнозов. Например, можно обучить несколько различных архитектур CNN и усреднить их прогнозы.
- Regularization Techniques: использовать параметры L1 и L2 в обучении моделей сверточных нейронных сетей (CNN) для регуляризации весов сети. Регуляризация помогает предотвратить переобучение, уменьшая сложность модели путем добавления штрафных функций к функции потерь. Так же можно исследовать другие методы регуляризации, такие как dropout или batch normalization, для дальнейшего улучшения обобщающей способности модели.

В целом обнаружение лица и детекция эмоции соответствует ожиданиям, но для получения более точных результатов обнаружение лиц на видео с веб-камеры и определение эмоций на

этих лицах с помощью предварительно обученной модели, помимо дальнейшей настройки модели, доработать кода так, чтобы была задержка по времени обработки или определение эмоции только при смене эмоции (выражения лица). Либо детекцией изменения положения лица.

6. Список источников

- Датасет, используемый в работе - <https://www.kaggle.com/datasets/ashishpatel26/facial-expression-recognitionferchallenge/data>
- Исследование возможности детектирования эмоций, А. Г. Лойко кафедра инженерной кибернетики НИТУ «МИСиС» Москва, Россия. - https://www.researchgate.net/publication/377271090_Issledovanie_vozmoznosti_detektirovania_emocij
- Выбор слоя активации в нейронных сетях: как правильно выбрать для вашей задачи <https://habr.com/ru/articles/727506/>
- Классификация изображений CNN в TensorFlow с шагами и примерами <https://www.guru99.com/ru/convnet-tensorflow-image-classification.html>
- Распознавание лиц при помощи Python и OpenCV <https://pythonist.ru/raspoznavanie-licz-pri-pomoshhi-python-i-opencv/>
- Лекции Нетология. Курс «Data Scientist»