

УДК 004.93'11

DOI: [10.26102/2310-6018/2025.49.2.041](https://doi.org/10.26102/2310-6018/2025.49.2.041)

Методика обучения классификаторов изображений с использованием дополнительных меток

Я.С. Петрова✉

*Московский государственный технический университет им. Н.Э. Баумана, Москва,
Российская федерация*

Резюме. Статья посвящена разработке методики обучения классификаторов с учетом отношений между классами, представленных в виде дополнительных меток. Для этого проведен анализ функций потерь, используемых в классификации, и подходов к внедрению в них дополнительных меток. По результатам анализа в качестве основы методики предложена триплетная функция потерь с гибкой границей, спроектированная на базе оригинальной триплетной функции. Гибкая граница дает возможность корректировать расстояния между векторными представлениями изображений в зависимости от степени различия их классов. Это позволяет моделировать разные степени сходства между классами: на уровне категории, группы, подгруппы. Дополнительно разработана стратегия подбора обучающих триплетов, препятствующая обнулению весов модели и ее застреванию на тривиальном решении. Методика апробирована в задачах классификации продуктов и болезней желудочно-кишечного тракта. В результате применения методики точность классификации увеличилась на 9 % в задаче распознавания болезней и на 6 % при распознавании продуктов. Было снижено количество грубых ошибок классификации. Пространство векторных представлений изображений, сформированное триплетной функцией потерь, позволяет решать задачу кластеризации, распознавать новые классы без дообучения модели.

Ключевые слова: функция потерь, классификация, компьютерное зрение, триплеты, метки, векторное пространство.

Для цитирования: Петрова Я.С. Методика обучения классификаторов изображений с использованием дополнительных меток. *Моделирование, оптимизация и информационные технологии*. 2025;13(2). URL: <https://moitvvt.ru/ru/journal/pdf?id=1928> DOI: 10.26102/2310-6018/2025.49.2.041

Method of training image classifiers using additional labels

I.S. Petrova✉

Bauman Moscow State Technical University, Moscow, the Russian Federation

Abstract. This paper is devoted to the development of a method for training classifiers that takes into account relationships between classes, represented as additional labels. The loss functions used in classification and the approaches to incorporating additional labels into them were analyzed. Based on this analysis, we propose as the foundation of our method a triplet loss with a flexible margin, designed on the basis of the original triplet loss. The flexible margin allows adjusting the distances between the embeddings of images depending on the difference degree between their corresponding classes. This makes it possible to model different levels of similarity between classes: category, group, and subgroup levels. In addition, we develop a triplet mining strategy that prevents the model's weights from collapsing to zero and getting stuck in a trivial solution. The method is validated on tasks of product classification and gastrointestinal disease classification. As a result of applying the method, classification accuracy increased by 9 % in the disease recognition task and by 6 % in the product recognition task. The number of severe classification errors was reduced. The image embedding space

formed by the triplet loss allows clustering and recognition of new classes without additional model training.

Keywords: loss function, classification, computer vision, triplets, labels, vector space.

For citation: Petrova I.S. Method of training image classifiers using additional labels. *Modeling, Optimization and Information Technology*. 2025;13(2). (In Russ.). URL: <https://moitvvt.ru/journal/pdf?id=1928> DOI: 10.26102/2310-6018/2025.49.2.041

Введение

Задача классификации – одна из базовых в компьютерном зрении [1], и ее популярность не спадает даже с увеличением количества вариантов применения искусственного интеллекта в повседневной жизни. Несмотря на распространение больших языковых моделей и фундаментальных моделей для извлечения признаков, потребность в небольших узкоспециализированных моделях, таких как классификаторы, сохраняется в сферах, где требуется:

- 1) конфиденциальность;
- 2) скорость обработки;
- 3) автономная работа на тонком клиенте (смартфон, носимая техника);
- 4) надежность.

В связи с высокой вычислительной сложностью универсальные модели требуют выполнения на специально оборудованном сервере, куда будут пересылаться данные для обработки. Если присутствует пересылка данных, то вышеперечисленные требования сложно или даже невозможно выполнить. И даже при выполнении на сервере классификатор, обученный на подходящем датасете, справляется с задачей лучше, чем универсальная модель [2]. Поэтому и на сегодняшний день остаются актуальными задачи обучения специализированных моделей.

В стандартном подходе к обучению классификаторов изображений классы по умолчанию полагаются независимыми, хотя в реальном мире они часто могут быть объединены в группы или иерархию. К примеру, разные продукты объединяет общая категория – фрукты, а разные сорта объединяет тип – яблоко. В классическом подходе эти отношения игнорируются, хотя являются важной информацией о классах, которую можно использовать в обучении модели.

Еще одна проблема использования стандартного подхода при распознавании изображений – ошибки классификаторов неочевидны, потому что визуальные признаки, на которые опирается модель, не всегда совпадают с признаками, на которые обращает внимание человек. Обычно для модели доминируют простейшие признаки вроде цвета или паттерна, фон может превалировать над основным объектом [3]. Информация о том, какие классы похожи с человеческой точки зрения, потенциально может обучить модель формировать более объяснимое пространство признаков и снизить количество грубых ошибок.

Поэтому цель данной статьи – предложить методику обучения классификаторов, использующую информацию об отношениях между классами. Для этого нужно решить следующие задачи:

- 1) провести анализ функций потерь, используемых в классификации, и подходов к внедрению в них дополнительной информации;
- 2) разработать функцию потерь, учитывающую отношения между классами.

Материалы и методы

Классический подход к обучению классификатора предполагает использование бинарной или категориальной кросс-энтропии в качестве функции потерь. Это

накладывает два важных ограничения на классы: количество классов заранее определено и фиксировано, множество классов единое.

Второе ограничение означает, что в кросс-энтропии есть две крайности. Или у каждого объекта строго один класс – категориальная кросс-энтропия. Или у каждого объекта сколько угодно классов из единого множества – бинарная кросс-энтропия. Если нужно обучить классификатор выбирать по одной метке из нескольких групп, например, уровней иерархии, то это можно реализовать только с помощью усложнения архитектуры нижних слоев модели и, соответственно, процесса их обучения [4].

Альтернативный подход к обучению классификатора основан на использовании метрических функций потерь. В таком случае снимаются вышеописанные ограничения на классы, и метод обучения разбивается на два этапа [5].

1. Обучение модели для создания векторных представлений изображений.
2. Классификация полученных векторных представлений.

Первый этап выполняется с помощью метрической функции, второй – любым простым классификатором или несколькими. Метрические функции потерь обучают модель располагать изображения одного класса близко в векторном пространстве, а разных классов – далеко. Для обучения не нужен точный список или конкретные названия классов. Таким образом, первый этап обучения не зависит от конкретных меток, только от их сходства или различия. Несмотря на это, базовые метрические функции потерь, такие как контрастная и триплетная, предполагают, что у каждого объекта только одна метка класса. Поэтому чтобы использовать дополнительные метки в процессе обучения с помощью метрических функций, нужно модифицировать функцию потерь.

Существующие модификации базовых триплетных функций можно разделить на две категории [6]:

- 1) прокси-ориентированные;
- 2) модификации расстояния между экземплярами.

В прокси-ориентированные функции дополнительные признаки классов закладываются через изменение расчета границы – минимального расстояния, на котором должны находиться друг от друга экземпляры разных классов. В рамках использования дополнительных меток отношений между классами наиболее интересна работа [7], в которой в разработанную авторами метрическую функцию потерь внедряется информация об иерархии между классами. Моделирование иерархии происходит с помощью разных параметров границ: если классы различаются на верхнем уровне иерархии, то расстояние между соответствующими им изображениями должно быть больше, чем если они различаются на нижнем уровне иерархии.

Однако известным недостатком функций, использующих прокси, является предположение, что векторные представления всех изображений класса стягиваются к одной точке, хотя на самом деле это не всегда соответствует реальному распределению. Использование прокси-ориентированных функций потерь может негативно влиять на качество распознавания, так как естественные визуальные различия между изображениями внутри одного класса заглушаются необходимостью формировать единый кластер в пространстве. Этот недостаток может быть компенсирован расчетом прокси положительных соседей вместо прокси класса, однако такой подход потенциально усложняет функцию потерь и ведет к появлению в ней дополнительных параметров.

В предлагаемой в статье методике применяются идеи [7] о расчете параметра границы в зависимости от различий между иерархиями классов двух объектов. Однако данная методика не предполагает расчет прокси для классов и, соответственно, лишена вышеописанного недостатка прокси-ориентированных функций. Проблема подбора

подходящих триплетов решается не за счет прокси, а за счет разработки отдельной стратегии их выбора. Кроме того, формула предлагаемой функции потерь основана на триплетной метрической функции, которая рассчитывается проще, так как не содержит операций логарифмирования и возведения в степень.

Другие модификации метрических функций потерь также предполагают изменение расчета границы, но уже между конкретными экземплярами внутри пары или триплета. К примеру, выбор отдельного параметра границы для каждого класса [6]. Некоторые модификации развивают оригинальную формулу метрической функции потерь, например, чтобы повлиять на траекторию перемещения векторных представлений в пространстве в процессе обучения модели [8]. Однако большая часть модификаций построена на идее границы, настраиваемой в процессе обучения. Например, ее уменьшение или увеличение в зависимости от сложности текущего триплета [9], использование разных значений границы для пар изображений внутри триплета [10].

Внедрение дополнительных меток классов напрямую исследовалось для триплетной функции потерь в задаче классификации типа объекта и его угла поворота на синтетических изображениях [11]. Авторы рассчитывали расстояние между векторными представлениями изображений в зависимости от сходства углов поворота между объектами в триплете. Другой способ внедрения [12] предполагает использование триплетной функции потерь в качестве вспомогательной при обучении. Тогда основная функция – кросс-энтропия – зависит от класса изображения, триплетная – от дополнительной метки, а итоговая функция потерь равна их сумме. Так же исследовалось внедрение информации об иерархии классов в контрастную функцию потерь в работе [13], где параметр границы настраивался по аналогии с [7].

В предлагаемой методике используются идеи [11] о внедрении дополнительных меток именно в триплетную функцию потерь, так как она специально спроектирована для моделирования отношений сходства и различия. Однако параметр границы настраивается более гибко и предлагает больше возможностей в моделировании отношений между классами за счет возможности выбирать отдельный параметр для каждого уровня иерархии.

Предлагаемая методика основана на метрической функции потерь, поэтому так же предполагает два этапа:

- 1) обучение модели для получения векторных представлений;
- 2) классификация векторных представлений.

Для первого этапа предлагается использовать функцию потерь с гибкой границей, о которой пойдет речь ниже. Для второго – один из методов классического машинного обучения.

Функция потерь с гибкой границей. Функция потерь с гибкой границей – это предлагаемая модификация оригинальной триплетной функции потерь, которая легла в основу разработанной методики. Формула оригинальной функции потерь приведена ниже.

$$L(x_a, x_p, x_n) = \max(D(x_a, x_p) - D(x_a, x_n) + \text{margin}, 0). \quad (1)$$

Здесь x_a, x_p, x_n – векторные представления якорного, положительного и отрицательного экземпляров. Якорный и положительный экземпляры по определению относятся к одному классу, отрицательный – к другому. Слагаемое *margin* – это константный параметр границы, выражающий минимальную необходимую разницу между расстоянием от якорного объекта до положительного и расстоянием от якорного объекта

до отрицательного. D – функция расчета расстояния между векторами, по умолчанию используется евклидово.

Для использования в функции потерь отношения между классами должны быть представлены в виде меток. Для удобства дальнейшей обработки количество меток у каждого объекта желательно унифицировать, например, дублировать метки нижнего уровня иерархии для объектов, у которых меньше меток. К примеру, если для класса «гала» указаны метки иерархии: «фрукт», «яблоко», а для класса «апельсин» только «фрукт», можно продублировать «апельсин», чтобы количество уровней иерархии совпало. Если дополнительные метки обозначают не иерархии, а группы, то для тех классов, у которых меньше меток групп, можно ввести искусственные метки «Другое».

Если количество меток для всех классов одинаково, то дополнительные метки можно хранить в двумерном массиве и обрабатывать быстрее, чем массив списков разной длины.

Чтобы иметь возможность настраивать степень влияния дополнительных меток на формирование векторного пространства, в триплетную функцию потерь (1) вводится гибкая граница M . В результате получается следующая формула:

$$L(x_a, x_p, x_n, l_a, l_n) = \max(D(x_a, x_p) - D(x_a, x_n) + M(l_a, l_n), 0). \quad (2)$$

Здесь l_a, l_n – метки якорного и отрицательного экземпляров. M – функция расчета параметра границы для заданного триплета. Она будет рассмотрена подробнее в подразделе.

При использовании нескольких меток важно уточнить понятия якорного, положительного и отрицательного экземпляров. В данном исследовании положительными считались экземпляры, все метки которых полностью совпадают с якорным. Отрицательные экземпляры – те, которые хотя бы на одну метку отличаются от якорного.

Вычислительная сложность функции потерь с гибкой границей выше, чем для оригинальной триплетной функции потерь, так как в ней есть дополнительные параметры – метки отношений между классами, которые нужно обрабатывать. Однако вычислительная сложность изменяется на константу, кратную количеству уровней иерархии. Поэтому до тех пор, пока количество уровней иерархии много меньше, чем количество обучающих триплетов в мини-батче, существенного роста сложности не возникает. Однако при увеличении количества меток на несколько порядков или хранение их в виде списков неунифицированной длины может значительно замедлить расчет функции потерь.

Выбор параметра границы. Функция потерь с гибкой границей проектировалась с учетом того, что дополнительные метки классов могут обозначать как группы, так и иерархии классов. Ключевое отличие этих типов меток состоит в том, что для иерархий работает правило: если у классов совпадают метки нижнего уровня, то метки верхнего тоже совпадают. Для групп это не выполняется: принадлежность класса одной группе не означает принадлежность другой. Поэтому выбор параметра границы для групп и иерархий осуществляется по-разному.

У иерархий есть важное преимущество – теоретически они могут формировать в пространстве не пересекающиеся кластеры верхнего уровня, в которые полностью включены кластеры нижних уровней. Для создания такого пространства можно выбрать значения границ, возрастающие от частного к общему. Тогда различные классы, относящиеся к одной ветви иерархии, будут размещаться ближе друг к другу в пространстве, чем классы из разных ветвей иерархии. В таком пространстве должно быть меньше грубых ошибок классификации, потому что даже если классификатор

неправильно определит метку конечного класса, его предсказание более вероятно окажется в области похожих классов, относящихся к той же ветви дерева иерархии. Для иерархий граница M подсчитывается по формуле:

$$M(l_a, l_n) = \max((m_1, m_2, \dots, m_h) \cdot \text{mask}(l_a, l_n)), \quad (3)$$

где $m_1, m_2, \dots, m_h$ – константные значения границы для каждого из h уровней иерархии, а $\text{mask}(l_a, l_n)$ – маска, рассчитываемая по формуле:

$$\text{mask}(l_{a_i}, l_{n_i}) = \begin{cases} 1, & \text{если } l_{a_i} = l_{n_i} \text{ для } \forall i \in (1, \dots, h). \\ 0 & \text{иначе} \end{cases} \quad (4)$$

Значение границы выбирается как максимальное из границ, соответствующих уровням иерархии, на которых метки классов якорного и отрицательного объекта не совпали. К примеру, если у классов не совпали первая и вторая метки, то значение границы будет выбрано как максимальное среди параметров границ первого и второго уровней иерархии.

Чтобы смоделировать иерархию в векторном пространстве, для первого (верхнего) уровня иерархии было взято максимальное евклидово расстояние между векторами единичной длины – 2. Для последнего уровня иерархии – небольшое, но достаточное для разделения классов в пространстве – 0,5. То есть константы m_i выбираются так, чтобы соответствовать неравенству: $m_1 > m_2 > \dots > m_h$.

В случае меток-групп параметр границы выбирается иначе, потому что объекты из разных групп могут перекрывать друг друга в векторном пространстве. Поэтому нужно отразить в обучении то, что различие классов по одним группам важнее, чем по другим. Это можно моделировать с помощью формулы (3), тогда в качестве констант $m_1, m_2, \dots, m_h$ нужно задать метрики границ для каждой группы. При этом не обязательно, чтобы выполнялось вышеупомянутое неравенство, характерное для иерархий. Но в некоторых случаях может быть эффективнее учитывать не важность несовпавших групп, а их количество. К примеру, все группы одинаково важны, потому что соответствуют некоторым базовым признакам классов. Тогда важным становится не то, какие признаки не совпали, а сколько таких несовпадений накопилось. В таком случае граница M может быть рассчитана по формуле:

$$M(l_a, l_n) = \sum((m_1, m_2, \dots, m_h) \cdot \text{mask}(l_a, l_n)), \quad (5)$$

в которой все параметры границ для несовпавших групп суммируются. При выборе констант m_i важно, чтобы выполнялась следующая формула:

$$\sum_1^h m_i = 2, \quad (6)$$

где 2 – максимальное евклидово расстояние между векторами единичной длины. При изменении метрики расстояния это число может быть другим. Такой способ расчета границы предпочтителен, если все группы одинаковы по важности и среди них нет доминирующих. Иначе стоит использовать формулу (3).

Важно, что группа или уровень иерархии, для которых выбрано большее значение границы, будет сильнее других влиять на формирование векторного пространства.

Стратегия подбора обучающих триплетов. В качестве основы методики выбрана функция потерь на основе триплетов. Известная проблема таких функций – зависимость от стратегии подбора триплетов [14]. Существует две основные стратегии: выбор полусложных и самых сложных триплетов (Рисунок 1) [15].

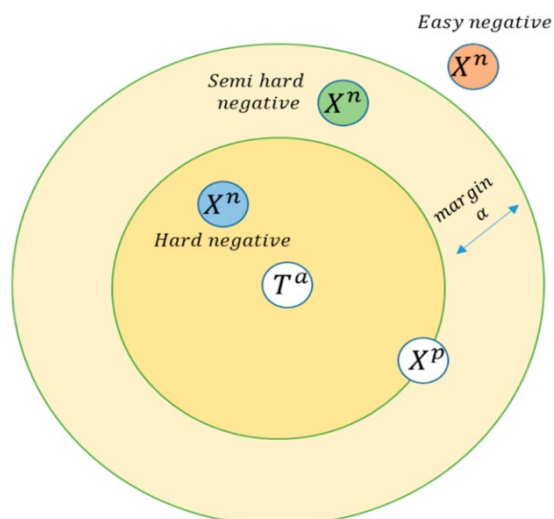


Рисунок 1 – Стратегии подбора триплетов
Figure 1 – Triplet mining strategies

Проблема выбора триплетов существует в связи с ростом датасетов. Чем больше объем датасета, тем больше возможных комбинаций троек объектов, тем сложнее выбрать из них нетривиальные, которые помогут точнее подобрать веса модели.

Самые сложные триплеты – это такие, где векторы объектов разных классов ближе друг к другу в пространстве, чем векторы объектов одного класса. Для этих триплетов значение функции потерь максимально и, соответственно, значение градиента для изменения весов больше, модель обучается быстрее. Однако при использовании этой стратегии модель может прийти к тривиальному решению – обнулить все веса, представлять все изображения в виде векторов нулей. Тогда значение функции потерь будет равно параметру границы, так как остальные слагаемые обнулятся. Это локальный минимум, в котором модель может застрять. Такая ситуация происходит, потому что при обучении модель не получает информации о том, как должны выглядеть правильные триплеты. Поэтому в начале обучения веса подбираются хаотично, в надежде случайно наткнуться на правильное направление их изменения.

Чтобы увеличить шансы корректной сходимости модели, используется стратегия выбора полусложных триплетов. Это триплеты, в которых векторы объектов одного класса ближе друг к другу, чем векторы объектов разных классов. Однако векторы объектов разных классов все еще находятся недостаточно далеко друг от друга, а одинаковых – недостаточно близко. На таких триплетах модель обучается медленнее, потому что значение функции потерь меньше, чем для самых сложных триплетов, но стабильнее. Стабильность достигается за счет того, что модели известно, как должны выглядеть правильные триплеты, достаточно только продолжать отодвигать объекты внутри них.

Чтобы обучать модель быстро, но при этом не застрять в локальном минимуме, была разработана следующая стратегия подбора триплетов. Обучение было разделено на два этапа: начальный и основной. Начальными считались первые 30 эпох. В них использовалась стратегия выбора полусложных триплетов, чтобы «подтолкнуть» коэффициенты модели в правильном направлении и увести от тривиального решения. На начальном этапе приоритетом считалась не скорость, а стабильность сходимости, однако к его концу веса модели уже близки к оптимальным.

Далее, в основном этапе, использовались самые сложные триплеты. Такая стратегия ускоряет сходимость модели, и значение функции потерь быстрее достигает плато.

Важно, что на начальном этапе используется максимально возможная скорость обучения, при которой модель способна сходиться, потому что основная задача этапа – преодолеть локальные минимумы. При переходе к основному этапу необходимо значительно снизить скорость обучения, чтобы не кардинально изменить уже подобранные на начальном этапе веса, а только скорректировать их. Эта мера необходима еще и из-за того, что среднее значение функции потерь при выборе самых сложных триплетов выше, поэтому при одинаковой скорости обучения сложные триплеты имеют большее влияние на веса модели, чем полусложные.

Таким образом, предлагается комбинация популярных стратегий подбора триплетов, чтобы стабилизировать обучение модели, но при этом не замедлять его.

Выбор подходящих триплетов предлагается осуществлять в пределах мини-батча, то есть порции данных для шага обучения. В мини-батч рекомендуется включать минимум 2 изображения каждого класса, чтобы можно было подобрать для него триплет. Важно сбалансировать количество изображений каждого класса в мини-батче, чтобы модель корректно распознавала все классы.

Классификация векторных представлений. Второй этап обучения классификатора при использовании метрических функций – это выбор и настройка модели, осуществляющей классификацию векторных представлений. Так как векторные представления уже сгруппированы по классам в векторном пространстве, нет необходимости обучать еще одну нейронную сеть для их классификации. Достаточно использовать методы классического машинного обучения. Наиболее естественным выбором для векторного пространства можно считать классификацию методом k ближайших соседей, потому что он учитывает возможность того, что экземпляры одного класса могут сформировать несколько небольших кластеров, а значит иметь не один центроид. Параметры для метода подбираются эмпирически в зависимости от задачи. В качестве значений по умолчанию в данной методике было выбрано $k = 5$ ближайших соседей, взвешенных по расстоянию. То есть вклад каждого из соседей в результат классификации объекта обратно пропорционален его расстоянию от объекта.

Результаты

Предлагаемая методика была проверена на двух общедоступных датасетах: The Hyper Kvasir [16] и Sku Classification¹. Оба датасета содержат три уровня иерархии меток: категория, подкатегория, класс. Первый посвящен классификации болезней желудочно-кишечного тракта, второй – продуктов в магазине. Были выбраны датасеты из разных предметных областей, чтобы качественнее оценить влияние выбранной методики.

Основные параметры датасета болезней ЖКТ: 10000 фото, 23 класса. Основные параметры датасета продуктов: 5500 фото, 81 класс.

В обоих датасетах количество изображений в классах не сбалансировано, поэтому подсчет метрик точности производился только для тех классов, в которых их было больше 50. В противном случае вклад статистически малозначимых классов в итоговую метрику оказывался слишком велик. Мини-батч для подбора триплетов составлялся из 2 изображений каждого класса. Большие размеры мини-батча не продемонстрировали положительного влияния на обучение.

¹ siva. SKUCLASSIFICATION Computer Vision Project. Roboflow Universe. URL: <https://universe.roboflow.com/siva-4or6j/skuclassification> (дата обращения: 03.03.2025).

Для проверки методики обучения классификатора с нуля, то есть без предобученных весов, использовалась простая FaceNet-подобная сверточная архитектура. После четырех сверточных блоков следовали два полносвязных слоя:

- 1) слой с 256 нейронами и функцией активации ReLU;
- 2) выходной слой с активацией softmax для подхода с кросс-энтропией или без активации, но с L2-нормализацией для метрического подхода.

Общее количество параметров сети около 2,7 миллионов, точное количество незначительно колеблется в зависимости от подхода.

Входные изображения были уменьшены до 100×100 пикселей с сохранением соотношения сторон. Размерность векторных представлений изображений, получаемых из модели – 128.

Для триплетной функции потерь было проведено обучение отдельно по стратегиям подбора самых сложных и полусложных триплетов, в итоговую таблицу была записана лучшая из полученных оценок точности. Для каждого подхода к обучению были использованы одинаковые мини-батчи, чтобы снизить их влияние на оценку результата.

В предлагаемой методике присутствуют дополнительные параметры: значения границы для каждого уровня иерархии. Они были выбраны в соответствии с гипотезой о расположении вложенных классов в пространстве и экспериментально проверены: [2, 1, 0,5].

Здесь и далее оценка точности распознавания рассчитывалась по формуле макро-F1, которая обычно используется для оценки классификации несбалансированных классов. Результаты обучения классификаторов с нуля приведены в Таблице 1.

Таблица 1 – Оценка классификаторов, обученных с нуля
Table 1 – Evaluation of classifiers trained from scratch

Подход к обучению	F1 классификации болезней ЖКТ, %	F1 классификации продуктов, %
Категориальная кросс-энтропия	71	72
Триплетная функция потерь	76	82
Предлагаемая методика	85	88

Далее эффективность методики была проверена в донастройке предобученной фундаментальной модели для извлечения признаков. К модели для извлечения признаков были добавлены два слоя, аналогичные последним двум слоям модели, обучаемой с нуля. Предобученные веса модели, используемые для извлечения признаков, были заморожены на протяжении обучения, изменялись только веса добавленного слоя. Размерность вектора признаков, извлекаемых предобученной моделью – 2048. Размерность итоговых векторных представлений – 128. Размер входного изображения – 384×384 пикселя, в соответствии с требованиями выбранной предобученной модели. В качестве такой модели была выбрана ConvNext-xlarge, обученная на датасете ImageNet, так как она продемонстрировала лучшую точность распознавания в ранее проведенном исследовании [17]. Общее количество параметров модели составило около 348 миллионов, но обучаемых из них – только 545 тысяч.

В Таблице 2 приведены результаты дообучения предобученных классификаторов.

Таблица 2 – Оценка классификаторов признаков
Table 2 – Evaluation feature classifiers

Подход к обучению	F1 классификации болезней ЖКТ, %	F1 классификации продуктов, %
Категориальная кросс-энтропия	83	93
Триплетная функция потерь	87	93
Предлагаемая методика	90	96

В предыдущих подразделах предобученные классификаторы показали лучшие результаты в решении задачи, поэтому именно они использовались для анализа грубых ошибок. В данном исследовании грубой ошибкой считалось ложное предсказание классификатора на верхнем уровне иерархии. В случае классификации продуктов это ошибка на уровне категории: фрукты, овощи, пакеты. В случае классификации болезней ЖКТ это ошибка определения отдела ЖКТ: верхнего или нижнего. В Таблице 3 приведены результаты подсчета грубых ошибок при обучении классификатора с помощью разных подходов.

Таблица 3 – Количество грубых ошибок классификаторов
Table 3 – Number of severe errors of classifiers

Подход к обучению	Количество ошибок в определении отдела ЖКТ	Количество ошибок в определении категории продуктов
Категориальная кросс-энтропия	7	4
Триплетная функция потерь	11	6
Предлагаемая методика	5	1

Обсуждение

Анализ результатов экспериментов в Таблицах 1 и 2 показал, что применение методики для обучения классификаторов улучшило оценку точности классификации на 6-9 % при обучении классификатора с нуля. При дообучении предобученных моделей для извлечения признаков оценка точности так же возросла на 3 % в сравнении с другими подходами к обучению. Важно, что в модели, обученной с нуля, параметров в 100 раз меньше, чем в предобученной. При этом в диагностировании болезней ЖКТ предлагаемая методика показала на меньшей модели точность на 2 % выше, чем кросс-энтропия на большей модели. Особенно примечательно сравнение с триплетной функцией потерь, так как для этого подхода к обучению и предлагаемой методики модели были полностью идентичны. Отличались только используемые функции потерь и стратегии подбора обучающих триплетов. И несмотря на это небольшое различие, предлагаемая методика продемонстрировала точность на 6-9 % выше.

При использовании методики для дообучения модели получилось достичь точности 90 % и выше для обеих задач, несмотря на очень разные предметные области.

При переходе от обучения с нуля к дообучению другие подходы показали резкий рост оценки точности классификации. Например, в задаче классификации продуктов: 72 % выросли до 93 %, 82 % – до 93 %. То есть разница в точности между большей и меньшей моделью составила от 11 до 21 %. В предлагаемой методике 88 % превратились в 96 %, то есть рост составил всего 8 %. Кроме того, при увеличении количества параметров модели различие в точности между предлагаемой методикой и оригинальной триплетной функцией уменьшилось. Это может значить, что методика позволяет более эффективно использовать небольшой объем параметров.

Анализ Таблицы 3 показывает, что в обеих задачах при использовании предлагаемой методики обучения количество грубых ошибок уменьшилось. Особенно заметна разница между триплетной функцией потерь и предлагаемой методикой. Учитывая, что оба подхода предполагают классификацию векторных представлений с помощью метода k ближайших соседей, разница в количестве грубых ошибок означает, что при обучении по предлагаемой методике векторное пространство сформировалось с большим влиянием иерархии классов, чем в других подходах.

Сравнивая предлагаемую методику с кросс-энтропией, важно отметить, что в ней отсутствует необходимость переобучать модель, чтобы добавить новый класс.

Заключение

В статье была предложена методика обучения классификаторов на основе специально разработанной триплетной функции потерь с гибкой границей. Проведенное исследование показывает, что применение предлагаемой методики улучшает точность обученного классификатора и снижает количество грубых ошибок распознавания.

Дальнейшие исследования предполагают эксперименты в других прикладных задачах, чтобы оценить влияние различных внедряемых дополнительных меток на точность распознавания обученного классификатора. Так же необходимо исследовать влияние аугментации на триплетную функцию потерь с гибкой границей, так как метрические функции известны восприимчивостью к ней. Развитие методики можно продолжить экспериментами с классификаторами векторных представлений: в данной статье применялся метод k ближайших соседей, однако есть множество других перспективных методов. Для классификации векторов подходят как методы классификации, так и кластеризации, поэтому выбор методов велик и это перспективное направление исследований.

СПИСОК ИСТОЧНИКОВ / REFERENCES

1. Евстраткин К.С., Султанова А.Р., Ерпелев А.В. OPENCV: варианты использования компьютерного зрения. В сборнике: *Цифровые технологии: наука, образование, инновации: материалы III Международного научного Форума профессорско-преподавательского состава и молодых ученых, 09 ноября 2020 года, Москва, Россия*. Москва: Московский государственный технологический университет «СТАНКИН»; 2021. С. 28–31.
2. Лобзин И.А. Исследование возможностей системы ИИ ChatGPT по решению задачи классификации. В сборнике: *Потенциал устойчивого инновационного развития: концепции, модели и практическое приложение: сборник статей по итогам Международной научно-практической конференции, 15 июня 2023 года, Пермь, Россия*. Уфа: Агентство международных исследований; 2023. С. 157–162.
3. Moayeri M., Pope Ph., Balaji Yo., Feizi S. A Comprehensive Study of Image Classification Model Sensitivity to Foregrounds, Backgrounds, and Visual Attributes. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 18–24 June 2022, New Orleans, LA, USA*. IEEE; 2022. P. 19065–19075. <https://doi.org/10.1109/CVPR52688.2022.01850>
4. Elhamod M., Diamond K.M., Maga A.M., et al. Hierarchy-Guided Neural Network for Species Classification. *Methods in Ecology and Evolution*. 2021;13(1):642–652. <https://doi.org/10.1111/2041-210X.13768>
5. Иванова Г.С., Петрова Я.С. Анализ методов компьютерного зрения для систематизации живописных полотен. *Нейрокомпьютеры: разработка, применение*. 2022;24(6):20–29. <https://doi.org/10.18127/j19998554-202206-02>

- Ivanova G.S., Petrova Y.S. Analysis of Computer Vision Methods for Painting Systematization. *Neurocomputers*. 2022;24(6):20–29. (In Russ.). <https://doi.org/10.18127/j19998554-202206-02>
6. Wang Yi., Liu P., Lang Yi., Zhou Q., Shan X. Learnable Dynamic Margin in Deep Metric Learning. *Pattern Recognition*. 2022;132. <https://doi.org/10.1016/j.patcog.2022.108961>
7. Sun Yi., Zhu Yu., Zhang Yu., et al. Dynamic Metric Learning: Towards a Scalable Metric Space to Accommodate Multiple Semantic Scales. In: *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, 19–25 June 2021, Online*. IEEE; 2021. P. 5393–5402. <https://doi.org/10.1109/CVPR46437.2021.00535>
8. Сулоев К.К., Шешкус А.В., Арлазаров В.Л. Сферические ограничения в триплетной функции потерь. *Труды Института системного анализа Российской академии наук*. 2023;73(2):50–58. <https://doi.org/10.14357/20790279230205>
Suloev K.K., Sheshkus A.V., Arlazarov V.L. Spherical Constraints in the Triplet Loss Function. *Proceedings of the Institute for Systems Analysis Russian Academy of Sciences*. 2023;73(2):50–58. (In Russ.). <https://doi.org/10.14357/20790279230205>
9. Nguyen Kh., Nguyen H.H., Tiulpin A. AdaTriplet: Adaptive Gradient Triplet Loss with Automatic Margin Learning for Forensic Medical Image Matching. In: *Medical Image Computing and Computer Assisted Intervention – MICCAI 2022: 25th International Conference: Proceedings: Part VIII, 18–22 September 2022, Singapore*. Cham: Springer; 2022. P. 725–735. <https://doi.org/10.48550/arXiv.2205.02849>
10. Wang Zh., Wang Yi., Dong B., Pracheta S., Hamlen K., Khan L. Adaptive Margin Based Deep Adversarial Metric Learning. In: *2020 IEEE 6th International Conference on Big Data Security on Cloud (BigDataSecurity), IEEE International Conference on High Performance and Smart Computing, (HPSC) and IEEE International Conference on Intelligent Data and Security (IDS), 25–27 May 2020, Baltimore, MD, USA*. IEEE; 2020. P. 100–108. <https://doi.org/10.1109/BigDataSecurity-HPSC-IDS49724.2020.00028>
11. Zakharov S., Kehl W., Planche B., Hutter A., Ilic S. 3D Object Instance Recognition and Pose Estimation Using Triplet Loss With Dynamic Margin. In: *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 24–28 September 2017, Vancouver, BC, Canada*. IEEE; 2017. P. 552–559. <https://doi.org/10.1109/IROS.2017.8202207>
12. Shaik S., Bucher B., Agrafiotis N., Phillips S., Daniilidis K., Schmenner W. Learning Portrait Style Representations. arXiv. URL: <https://doi.org/10.48550/arXiv.2012.04153> [Accessed 3rd May 2025].
13. Новикова К.Н. Тенденции развития методов классификации объектов на основе модификаций контрастного обучения. В сборнике: *Химия и инженерная экология – XXIII: сборник трудов международной научной конференции (школа молодых ученых), посвященной сотрудничеству с союзными государствами, 25–26 сентября 2023 года, Казань, Россия*. ИП Сагеев А.Р.; 2023. С. 200–205.
Novikova K.N. Development Trends of Object Classification Methods Based on Modifications Contrastive Learning. In: *Khimiya i inzhenernaya ekologiya – XXIII: sbornik trudov mezhdunarodnoi nauchnoi konferentsii (shkola molodykh uchenykh), posvyashchennoi sotrudnichestvu s soyuznymi gosudarstvami, 25–26 September 2023, Kazan, Russia*. IP Sagiev A.R.; 2023. P. 200–205. (In Russ.).
14. Рогачёв Н.Е. Проблема коллапса нейронных сетей при использовании Triplet Loss. В сборнике: *Веб-программирование и интернет-технологии WebConf2018: тезисы докладов 4-й Международной научно-практической конференции, 14–18 мая 2018 года, Минск, Беларусь*. Минск: Белорусский государственный университет; 2018. С. 92.

15. Abdullah T., Bazi Ya., Al Rahhal M.M., Mekhalfi M.L., Rangarajan L., Zuair M. TextRS: Deep Bidirectional Triplet Network for Matching Text to Remote Sensing Images. *Remote Sensing*. 2020;12(3). <https://doi.org/10.3390/rs12030405>
16. Borgli H., Thambawita V., Smedsrud P.H., et al. *HyperKvasir*, a Comprehensive Multi-Class Image and Video Dataset for Gastrointestinal Endoscopy. *Scientific Data*. 2020;7. <https://doi.org/10.1038/s41597-020-00622-y>
17. Петрова Я.С., Иванова Г.С. Сравнение моделей извлечения признаков из изображений с несколькими аннотациями. *Математические методы в технологиях и технике*. 2023;(2):71–74.
Petrova I.S., Ivanova G.S. Comparison of Feature Extraction Models from Images with Several Annotations. *Mathematical Methods in Technologies and Technics*. 2023;(2):71–74. (In Russ.).

ИНФОРМАЦИЯ ОБ АВТОРАХ / INFORMATION ABOUT THE AUTHORS

Петрова Яна Сергеевна, ассистент, **Iana S. Petrova**, assistant, Bauman Moscow Московский государственный технический State Technical University, Moscow, the Russian университет им. Н.Э. Баумана, Москва, Federation.

Российская Федерация.

e-mail: ypetrova@bmstu.ru

ORCID: [0009-0005-4377-7981](https://orcid.org/0009-0005-4377-7981)

Статья поступила в редакцию 27.04.2025; одобрена после рецензирования 25.05.2025; принята к публикации 07.06.2025.

The article was submitted 27.04.2025; approved after reviewing 25.05.2025; accepted for publication 07.06.2025.