

DOI: 10.15514/ISPRAS-2025-37(6)-44



Векторные представления шрифтов: дополнительный признак для понимания документов

^{1,2} Д.Е. Копылов, ORCID: 0009-0000-6348-4004 <it-daniil@yandex.ru>

^{1,2} М.В. Щурик, ORCID: 0009-0004-9144-1617 <marriiamnii@gmail.com>

¹ Институт динамики систем и теории управления имени В.М. Матросова СО РАН,
664033, Россия, г. Иркутск, ул. А. Лермонтова, д. 134.

² Институт математики и информационных технологий
Иркутского государственного университета
664003, Россия, Иркутск, бульвар Гагарина, д. 20.

Аннотация. В статье представлена модель на основе сверточной нейронной сети, которая ставит в соответствие изображению текста векторы, кодирующие информацию о шрифтах. Модель состоит из двух идентичных сверточных блоков, объединяющих признаки в вектор, который затем анализируется линейными слоями для поиска отличий. Обученная таким образом модель способна различать шрифты, игнорируя содержание текста, что делает ее универсальной для различных типов документов. Векторные представления шрифтов тестируются на дополнительных задачах, таких как классификация текста по жирности начертания и наклону, демонстрируя высокую точность и подтверждая их полезность для анализа стилевых особенностей. Эксперименты с вариативными и ручными шрифтами показывают универсальность модели и ее применимость для работы с разнообразными данными. Результаты сравнения с базовой моделью подтверждают эффективность предложенной архитектуры. Однако выявлены ограничения, связанные с работой на данных низкого качества и мультиязычных текстах, что открывает направления для будущих исследований. Предложенный подход представляет значительный вклад в область обработки документов, расширяя возможности анализа шрифтов и их использования в задачах классификации, поиска и выделения ключевых элементов текста. Код и модели были опубликованы на GitHub (<https://github.com/YRL-AIDA/FontEmb>).

Ключевые слова: сверточные нейронные сети; классификация шрифтов; нейронные сети; компьютерные шрифты.

Для цитирования: Копылов Д.Е., Щурик М.В. Векторные представления шрифтов: дополнительный признак для понимания документов. Труды ИСП РАН, том 37, вып. 6, часть 3, 2025 г., стр. 177–188. DOI: 10.15514/ISPRAS-2025-37(6)-44.

Благодарности: Работа выполнена в рамках государственного задания Министерства науки и высшего образования Российской Федерации (тема № 1023110300006-9).

Vector Representations of Fonts: an Additional Feature for Understanding Documents

¹ D.E. Kopylov, ORCID: 0009-0000-6348-4004 <it-daniil@yandex.ru>

² M.V. Shchurik, ORCID: 0009-0004-9144-1617 <marriiamnii@gmail.com>

¹ Matrosov Institute for System Dynamics and Control Theory of the Siberian Branch of Russian Academy of Sciences (ISDCT SB RAS),
134, Lermontov st., Irkutsk, 664033, Russia.

² Irkutsk State University Institute of Mathematics and Information Technologies,
20, Gagarin Boulevard, Irkutsk, 664003, Russia.

Abstract. The article presents a model based on a convolutional neural network that matches a vector of embeddings encoding information about fonts to a text image. The model consists of two identical convolutional blocks that combine features into a vector, which is then analyzed by linear layers to find differences. The model trained in this way is able to distinguish fonts, ignoring the text content, which makes it universal for various types of documents. Embedding vectors are tested on additional tasks, such as text classification by fatness and tilt, demonstrating high accuracy and confirming their usefulness for analyzing stylistic features. Experiments with variable and manual fonts show the versatility of the model and its applicability to work with a variety of data. The results of the comparison with the base model confirm the effectiveness of the proposed architecture. However, the limitations associated with working with low-quality data and multilingual texts have been identified. The code and models were published on GitHub (<https://github.com/YRL-AIDA/FontEmb>).

Keywords: convolutional neural networks; font classification; neural networks; computer fonts.

For citation: Kopylov D.E., Shchurik M.V. Vector representations of fonts: an additional feature for understanding documents. *Trudy ISP RAN/Proc. ISP RAS*, vol. 37, issue 6, part 3, 2025, pp. 177-188 (in Russian). DOI: 10.15514/ISPRAS-2025-37(6)-44.

Acknowledgements. The research was carried out within the state assignment of Ministry of Science and Higher Education of the Russian Federation (theme No. 1023110300006-9).

1. Введение

1.1 Цель генерации векторов шрифтов

Анализ растровых изображений документов (сканированные PDF, фотографии страниц) представляет собой важную задачу компьютерного зрения с приложениями в юриспруденции, делопроизводстве и цифровых архивах. Современные системы обработки документов успешно решают базовые задачи: распознавание текста (OCR), детекцию структурных элементов (таблиц, заголовков) и извлечение семантической информации. Однако эти подходы преимущественно используют текстовые данные и пространственное расположение элементов. Когда в такие модели стали подавать изображение слов [1] их качество выросло.

Между тем, важно не столько изображение, сколько информация о шрифте. Эта информация несет значительную семантическую нагрузку: различие шрифтов часто маркирует смысловые разделы документа (заголовки, цитаты, примечания), а их стилистические особенности могут указывать на тип документа (официальный, рекламный, технический). Традиционные методы обработки документов не учитывают эти особенности, хотя их интеграция могла бы улучшить качество структурного анализа и интерпретации содержимого.

Целью данной работы является разработка компактной модели для генерации векторных представлений шрифтов, независимых от конкретных символов. В отличие от существующих решений, акцент делается на эффективность обработки: малый размер как самой модели, так и выходных векторных представлений, что критически важно для массовой обработки документов, где количество текстовых элементов на порядки превышает разнообразие используемых шрифтов.

1.2 Про существующие подходы

Активные исследования шрифтов начались в конце 1980-х годов. В этих исследованиях основное внимание уделялось оптическому распознаванию символов [2]. В 2000-х годах сформировалось самостоятельное направление, посвященное анализу статистических признаков и классификации шрифтов [3-4]. Эти ранние работы заложили методологическую основу, однако использовавшиеся в них подходы требовали ручного выделения признаков. Значительный прогресс был достигнут в работе [5], где впервые применили сочетание методов глубокого обучения (SCAE) с обширным набором данных, получив в результате 768-мерные векторные представления шрифтов.

С появлением современных архитектур CNN, таких как AlexNet и ResNet, векторные представления шрифтов нашли применение в различных смежных областях: улучшении систем OCR [6], подборе гармоничных сочетаний шрифтов [7], анализе их эмоционального воздействия [8]. Однако сохранялась проблема высокой размерности получаемых представлений: даже сокращение вектора до 200–300 измерений [8-9] оставалось недостаточным для эффективной работы в системах с ограниченными вычислительными ресурсами.

Высокая размерность векторных представлений оправдана в задачах тонкого анализа и настройки шрифтов, где требуется максимальная детализация. Однако при обработке документов, где количество используемых шрифтов редко превышает десяти, а число отдельных элементов может быть крайне велико, более предпочтительными оказываются компактные модели с малым размером выходного вектора. В этом отношении представляет интерес исследование [10], где использовалась малая сеть, формирующая всего 4-мерный вектор признаков.

Отдельное направление исследований связано с адаптацией методов векторного представления для различных языковых систем. Примером могут служить работы по распознаванию арабских шрифтов [11] и созданию шрифтов по описанию и образцу на другом языке [12]. В этих задачах также традиционно применяются векторные представления большой размерности, что поднимает вопрос об оптимальном балансе между детализацией и эффективностью вычислений. С точки зрения данной работы подходы [11-12] интересны, поскольку они не обращают внимание на текстовое содержание.

1.3 Предлагаемый в работе подход

Большинство существующих методов анализа шрифтов требуют либо полного алфавита, либо целых слов для работы. В реальных документах такие идеальные условия встречаются редко. В статья предлагается более практическое решение – модель, которая определяет схожесть шрифтов по отдельным сегментам изображения слова.

Предлагаемый подход использует простую сверточную сеть, которая анализирует изображение символа и выдает компактный вектор признаков. Затем линейные слои сравнивают два таких вектора и выдают бинарный ответ: 1, если шрифты совпадают, 0, если они разные. Главное преимущество – модель работает с любыми отдельными сегментами, не требуя алфавита и даже распознанных символов.

Такой подход особенно полезен при обработке сканов и фотографий документов. Модель легко встроить в существующие системы анализа документов, чтобы учитывать шрифтовую информацию при распознавании структуры и содержания.

2. Данные

2.1 Признаки шрифтов

Для анализа и классификации шрифтов в документах используются различные визуальные и стилевые признаки. Модель должна различать эти признаки независимо от содержания текста, что особенно важно для задач, где требуется выделить ключевые элементы документа

(например, заголовки, сноски или важные фрагменты). Однако для успешного обучения модели, способной отличать шрифты друг от друга, необходимо обеспечить достаточную вариативность этих признаков в данных.

Важным аспектом при подготовке данных является их сбалансированность. Под сбалансированностью в данном контексте понимается не равное количество примеров для каждого шрифта, а баланс между различными комбинациями признаков. В наборе данных может присутствовать много примеров текста со шрифтом Times New Roman, но всего несколько примеров текста со шрифтом Comic Sans MS. Шрифт Times New Roman может быть представителем сразу нескольких признаков, а Comic Sans MS представлять только рукописные шрифты с заданной комбинацией других признаков.

Также сразу следует отметить, что в данной работе не ставится задача точной идентификации конкретного шрифта (например, определение, что перед нами именно Times New Roman). Вместо этого модель должна научиться строить векторные представления, которые отражают схожесть шрифтов на основе их визуальных и стилевых признаков. В табл. 1 приведены основные признаки шрифтов, которые могут быть использованы для обучения модели. Эти признаки не являются исчерпывающими, но позволяют выделить 144 различные комбинации свойств, что достаточно для практических задач, поскольку в документах обычно используются шрифты, которые явно отличаются друг от друга.

Табл. 1. Признаки шрифтов и примеры.

Table 1. Font signs and examples.

Тип	Вариации	Пример
По начертанию	Тонкие	Courier New, Leelawadee UI Semilight, Segoe Script
	Нормальные	Arial, Times New Roman, Calibri
	Жирные	Impact, Arial Black
По наклону	Наклонные	Monotype Corsiva, Segoe Script
	Не наклонные	Arial, Times New Roman, Calibri
Моноширинные или нет	Моноширинные	Courier New, Consolas, Cascadia Mono
	Не моноширинные	Arial, Times New Roman, Calibri
По засечкам	Гротески	Arial, Calibri, Helvetica, Verdana
	Антиквы	Times New Roman, Book Antiqua
	Брусковые	Impact, Arial Black, Constantia
По схожести с рукописными	Схожи с рукописными	Segoe Script, Comic Sans MS, Monotype Corsiva
	Печатные	Arial, Times New Roman, Calibri
По форме	Прямоугольные	Arial, Arial Black, Impact
	Круглые	Cambria, Constantia, Courier New

2.2 Вариативные шрифты

Одним из наиболее интересных и перспективных направлений развития шрифтов являются вариативные шрифты [13] (Variable Fonts). В отличие от традиционных шрифтов, где каждый стиль (например, обычный, жирный, курсив) представлен отдельным файлом, вариативные шрифты позволяют динамически изменять параметры шрифта, такие как толщина, наклон, ширина и даже форма символов, в рамках одного файла. Это достигается за счет использования осей вариативности, которые определяют, как изменяется внешний вид шрифта. Например, ось "wght" регулирует толщину символов, а ось "ital" – их наклон.

Вариативные шрифты предоставляют значительную гибкость в дизайне документов. Они позволяют не только варьировать признаками между двумя крайними значениями (например, тонкий и жирный), но и использовать промежуточные значения, что делает их особенно полезными для создания адаптивных и визуально привлекательных интерфейсов. Среди популярных примеров вариативных шрифтов можно выделить: Advent Pro, Roboto, EB Garamond, Caveat, Sofia Sans Condensed, Victor Mono, Ysabeau Infant, Finlandica, Piazzolla.

В данной работе мы рассматриваем такие свойства, как написание в верхнем регистре, подчеркивание, зачеркивание как свойства шрифта. Эти свойства также могут варьироваться, что позволяет создавать дополнительные вариации шрифтов. Например, подчеркивание может быть реализовано не только стандартным способом, но и путем отрисовки прямоугольника внизу слова, что имитирует подчеркивание.

Использование вариативных шрифтов и их свойств при обучении модели позволяет значительно увеличить разнообразие данных и улучшить способность модели к обобщению.

2.3 Генерация изображений шрифтов

Помимо выбора шрифтов, важным аспектом подготовки данных является отрисовка слов с использованием этих шрифтов. Для обучения модели, которая должна различать шрифты, не обращая внимания на текст и его позицию, необходимо обеспечить, чтобы изображения содержали случайный текст и случайное расположение символов. Это позволяет модели фокусироваться исключительно на стилевых и визуальных особенностях шрифтов, а не на семантике или структуре текста.

Другой важный момент: поскольку мы оцениваем свойство шрифта, а не символов и слов, то не обязательно, чтобы слова целиком попадали в область. Достаточно взять прямоугольник и допустить возможность, что в окно модели попадет не больше 3-х символов (в худшем случае один символ попадет целиком и два будут обрезаны по бокам). Отметим, что на практике можно разбивать изображение строки или слова на одинаковые блоки (размером совпадающим с высотой) и в качестве результата возвращать вектор со средними значениями, уже характеризующий строку или слово. Чтобы показать разнообразие символов, текст генерируется как на английском, так и на русском, но символы не смешиваются, чтобы исключить особенности стилей языков.

Изображения группируются парами с одинаковым шрифтом и разным. Последние включают еще вариант совпадения текста. В пропорции пары с одинаковым шрифтом составляют – 50%, пары с разным шрифтом и разным текстом – 25%, пары с разным шрифтом и одинаковым текстом – 25%. Пары с одинаковым шрифтом и одинаковым текстом исключены из рассмотрения, чтобы избежать ситуации, при которой модель фокусируется на содержании текста.

Также при сравнении модель не должна обращать внимание на качество, поэтому изображения уменьшаются в случайное число раз и измененные возвращаются к исходному размеру. Размер шрифта выбирается также случайно.

Такой подход к генерации изображений позволяет создать разнообразный и сбалансированный набор данных, направленный на обучение модели, способной различать шрифты на основе их визуальных и стилевых особенностей. Это особенно важно для последующего применения модели в задачах, где требуется анализ шрифтов, независящий от содержания текста. Число всех пар равняется 1 миллиону для набора, часть которого приведена в табл. 1. Также число пар с использованием вариативных шрифтов составляет 1 миллион изображений. Размер каждого изображения из пары равняется 40x40 пикселей.

3. Архитектура модели, ее обучение и тестирование

3.1 Архитектура

Модель, предложенная в данной работе, основана на простой сверточной нейронной сети (CNN), архитектура которой была взята из работы [10].

Сверточная часть в работе [10] состоит из двух VGG-блоков (visual geometry group), каждый из которых состоит из двух последовательно идущих сверточных слоев и слоя понижения разрешения или подвыборки (pooling). За ними следуют полносвязные слои. В данной работе архитектура была видоизменена (рис. 1). Сверточная часть (CNNModel) стала состоять из одного блока. Каждое изображение из пары проходит через эту сеть, на выходе для каждого изображения строится векторное представление. Векторные представления конкатенируются. И только после этого полученный вектор передается в полносвязные слои.

VGG-блок состоит из двух сверточных слоев. Первый слой состоит из одного входного канала (изображение в градациях серого) и 16 выходных каналов, при этом; размер ядра свертки: 3x3. Второй сверточный слой имеет аналогичные параметры, но с 16 входными и 32 выходными каналами. Последним идет операция выбора максимального значения в окне 2x2 с шагом 2 (слой, известный как pooling layer), что уменьшает размерность изображения вдвое по каждому измерению. На выходе сверточных слоев и операций уменьшения размерности данные преобразуются в одномерный вектор и передаются в полносвязный слой, который имеет входной размер 3200 (что соответствует объединенному размеру карт признаков 32x10x10).

Линейная часть представляет собой два полносвязных слоя с входным размером 3200 и выходным 64 для первого и с входным размером 64 и выходным размером 8 для второго.

Для введения нелинейности после каждого сверточного и полносвязного слоя используется функция активации ReLU.

Для обучения модели вводится вспомогательная модель (на рис. 1 – DiffModel), которая принимает на вход два вектора, возвращаемых основной моделью, конкатенирует их и пропускает через два полносвязных слоя с функцией активации ReLU. Первый полносвязный слой имеет вход и выход равный 16; второй полносвязный слой имеет вход равный 16 и выход равный 1 (бинарная классификация).

3.2 Обучение

Модель обучается решать задачу по сравнению шрифтов. Основная цель обучения вспомогательной модели (DiffModel) – научиться определять, являются ли два входных вектора, возвращаемых после свертки, представлениями одного и того же шрифта или разных. Если шрифты одинаковы, модель DiffModel возвращает 1, иначе 0.

Сама по себе задача сравнения стилей шрифтов не представляет большого интереса. Наиболее важным является обучение сверточной сети CNNModel, которая формирует векторные представления для шрифтов. Эти векторы должны кодировать стилистическую информацию о шрифтах, что делает их полезными для решения других задач.

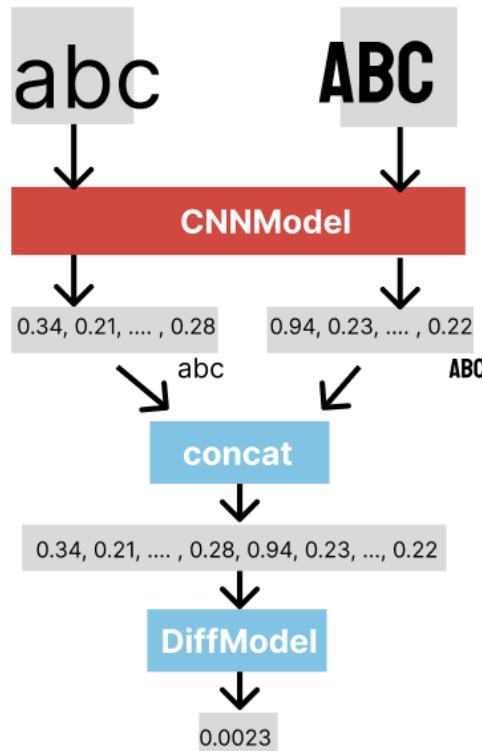


Рис. 1. Архитектура модели
Fig. 1. Model architecture.

Для обучения модели используется набор данных, состоящий из 1 миллиона изображений, для тестирования модели – набор, состоящий из 10000 изображений. Обучение проводится с использованием оптимизатора Adam с шагом обучения 0.0005. Размер порции данных за один шаг обучения составляет 256, что обеспечивает баланс между скоростью обучения и стабильностью обновления параметров модели. Модель обучается в течение 100 эпох.

В качестве функции потерь используется функция бинарной кросс-энтропии (Binary Cross-Entropy, BCE), которая минимизирует ошибку при классификации пар шрифтов (одинаковые/разные). Это позволяет сети CNNModel научиться формировать такие векторы, которые сохраняют информацию о стилевых особенностях шрифтов.

3.3 Оценка качества векторов

Как неоднократно подчеркивалось, главной задачей является не сравнение шрифтов, а формирование векторов, которые могут характеризовать их стилистические особенности. Для проверки полезности этих векторов решаются две дополнительные задачи:

1. Классификация текста по жирности начертания: модель должна определить, является ли текст жирным или обычным начертанием.
2. Классификация текста по наклону: модель должна определить, является ли текст прямым или курсивным.

Для решения этих задач используются модели, аналогичные DiffModel. Они также состоят из двух полно связанных слоев с функцией активации ReLU, но число узлов на первом слое уменьшено в два раза по сравнению с моделью DiffModel, как и на последующем слое. Выходной

слой остается таким же, как и в DiffModel.

Эти задачи позволяют оценить, насколько хорошо векторы, возвращаемые моделью CNNModel, кодируют информацию о стилевых особенностях шрифтов. Если модель успешно справляется с этими задачами, это свидетельствует о том, что векторное представление содержит полезную информацию, которая может быть использована для решения других задач, таких как классификация документов, поиск похожих шрифтов или анализ стилей текста.

4. Эксперименты

В данном разделе представлены результаты экспериментов, проведенных для оценки эффективности предложенной модели. Основное внимание уделяется сравнению с моделью, состоящей из двух VGG-блоков, решению задач классификации по жирности и наклону, а также анализу влияния различных типов шрифтов на качество модели.

4.1 Оценка качества векторов

Для оценки эффективности предложенной архитектуры была проведена серия экспериментов, в которых сравнивались результаты базовой модели с одним и двумя VGG-блоками (табл. 2). Обе модели обучались на одном и том же наборе данных, содержащем изображения текста с различными шрифтами.

Результаты показывают, что в отличие от работы [10] для данной задачи достаточно одного VGG-блока.

4.2 Классификация по жирности начертания и наклону

Для проверки информативности векторных представлений шрифтов была решена задача классификации текста по жирности начертания (обычный и жирный). Новая модель (вместо DiffModel) была дообучена на 10 тысяч изображений. Аналогично была решена задача классификации текста по наклону (прямой и курсив). Результаты представлены в табл 3.

Табл. 2. Архитектура сверточной модели.

Table 2. Architecture convolution models.

Метрика	Два VGG-блока	Один VGG-блок
точность	0.8760	0.8954
F1-мера	0.8849	0.9007
Число параметров CNNModel	837 880	210 184

Табл. 3. Классификация жирности начертания и курсива.

Table 3. Bold and italic classification.

		Два VGG-блока	Один VGG-блок
Жирность	точность	0.8750	0.9060
	F1-мера	0.8713	0.9052
Курсив	точность	0.5390	0.5920
	F1-мера	0.4743	0.5920

Результаты исследования демонстрируют, что полученные векторные представления содержат информацию о таких характеристиках шрифтов, как жирность начертания и наклон. Это делает их полезными для решения задач, связанных с анализом шрифтов.

Анализ показал, что при поиске отличий между шрифтами модель в меньшей степени ориентируется на наклон символов, в то время как жирность является ключевым признаком, который модель использует для дифференциации шрифтов.

Модель может демонстрировать ограниченную переносимость (снижение точности при работе с данными, которые отличаются от обучающей выборки) при решении подобных задач. Это обусловлено тем, что сравнение по жирности и начертанию эффективно только при анализе двух конкретных шрифтов. Анализ отдельно взятого шрифта не позволяет сделать выводы об особенностях начертания.

4.3 Варианты данных

Для анализа влияния вариативных шрифтов на качество модели был проведен эксперимент с использованием шрифтов, таких как Roboto, Oswald и других, которые позволяют варьировать толщину, наклон и другие параметры. Также были добавлены стили, собранные вручную, чтобы оценить, как модель справляется с нестандартными шрифтами.

Модель работает одинаково хорошо с различными шрифтами. Она также способна распознавать шрифты, с которыми на которых не была обучена, но при этом точность ее работы снижается на 20% (табл. 4).

Табл. 4. Вариативные и обычные шрифты.

Table 4. Variable and classic fonts.

Набор для обучения	Тестирование на обычных шрифтах	Тестирование на вариативных шрифтах	Разница в метрике правильности
Обычный + Вариативный	0.8854	0.9054	0.0200
Обычный	0.9077	0.7193	0.1884
Вариативный	0.7155	0.9020	0.1865

5. Обсуждение результатов

Предложенная модель, основанная на простой сверточной нейронной сети, демонстрирует значительный потенциал в задаче генерации векторных представлений для анализа шрифтов. Использование двух идентичных сверточных блоков с последующим объединением в вектор и линейными слоями для поиска отличий позволяет модели эффективно различать шрифты, игнорируя при этом содержание текста. Это особенно важно для задач, где стилевые особенности шрифтов играют ключевую роль, таких как классификация документов или выделение ключевых элементов текста.

Одним из наиболее интересных аспектов работы является тестирование векторных представлений шрифтов на дополнительных задачах, таких как классификация по жирности и наклону. Высокие показатели точности для жирности (90.5%) свидетельствуют о том, что векторные представления содержат как минимум информацию о жирности начертания шрифтов. Такие результаты открывают возможности для использования модели в более сложных задачах, таких как анализ структуры документов.

Следует отметить, что модель все еще имеет некоторые ограничения. Например, хотя она успешно справляется с игнорированием повторяющихся символов, ее эффективность на данных с низким разрешением или значительными искажениями требует дальнейшего изучения. Как выяснилось, это особенно актуально, по крайней мере, для информации о курсиве. Это может стать направлением для будущих исследований.

6. Заключение

В данной работе представлена модель, основанная на простой сверточной нейронной сети, которая ставит в соответствие изображению текста векторные представления, кодирующие информацию о шрифтах. Основная цель модели – научиться различать шрифты, игнорируя при этом содержание текста, что делает ее универсальной для различных типов документов. Модель состоит из двух идентичных сверточных блоков, которые извлекают признаки из изображения, объединяют их в вектор и передают через линейные слои для поиска отличий. Обученная таким образом модель демонстрирует высокую точность в задаче сравнения стилей шрифтов, что подтверждается результатами экспериментов.

Важным аспектом работы является тестирование векторных представлений шрифтов на дополнительных задачах, таких как классификация текста по жирности и наклону. Результаты показывают, что полученные векторные представления содержат полезную информацию о стилевых особенностях шрифтов, что делает их применимыми для решения более сложных задач, таких как анализ структуры документов, поиск похожих шрифтов или выделение ключевых элементов текста. Кроме того, эксперименты с вариативными шрифтами и шрифтами, собранными вручную, демонстрируют универсальность модели и ее способность работать с разнообразными типами данных.

Предложенный подход оказался весьма эффективным и открыл перспективы для применения полученных результатов в различных областях. В будущих исследованиях можно сосредоточиться на оптимизации модели для работы с более сложными и разнообразными данными, а также на интеграции ее в существующие системы обработки документов.

Список литературы / References

- [1]. Xu Y., Li M., Cui L. Huang S. Zhou M. LayoutLM: Pre-training of Text and Layout for Document Image Understanding. In Proc. of the 26th ACM SIGKDD, 2020, pp. 1192-1200. DOI:10.1145/3394486.3403172.
- [2]. Brzakovic D., Tou J. T. An approach to computer-aided document examination. International journal of computer & information sciences, vol. 14, 1985, pp. 365-385.
- [3]. Allier B., Emptoz H. Type extraction and character prototyping using Gabor filters. In Proc. of the 7th ICDAR, 2003, pp. 799-803. DOI: 10.1109/ICDAR.2003.1227772.
- [4]. O'Donovan P., Lübeck J., Agarwala A., Hertzmann A. Exploratory font selection using crowdsourced attributes. ACM Transactions on Graphics, vol. 33, pp. 1–9. DOI:10.1145/2601097.2601110.
- [5]. Wang Z., Yang J., Jin H., Shechtman E., Agarwala A., Brandt J., Huang, T.S. DeepFont: Identify Your Font from An Image. In Proc. of the 23rd ACM MM, 2015, pp. 813-814. DOI:10.1145/2733373.2807988.
- [6]. Tensmeyer C., Saunders D., Martinez T.R. Convolutional Neural Networks for Font Classification. In Proc. of 14th IAPR ICDAR, 2017, pp. 985-990. DOI:10.1109/ICDAR.2017.164.
- [7]. Jiang S., Wang Z., Hertzmann A., Jin H., Fu Y. Visual font pairing. IEEE Transactions on Multimedia, 2019, 22(8), pp. 2086-2097. DOI:10.1109/TMM.2019.2952266.
- [8]. Yasukochi N., Hayashi H., Haraguchi D., Uchida S. Analyzing Font Style Usage and Contextual Factors in Real Images. In Proc. of the 17th ICDAR, 2023, pp. 331-347. DOI:10.1007/978-3-031-41682-8_21.
- [9]. Kulahcioglu T., De Melo G. Fonts like this but happier: A new way to discover fonts. In Proc of the 28th ACM MM, 2020, pp. 2973-2981. DOI:10.1145/3394171.3413534.
- [10]. Bychkov O., Merkulova K., Dimitrov G., Zhabska Y., Kostadinova I., Petrova P., Petrov P., Getova I., Panayotova G. Using Neural Networks Application for the Font Recognition Task Solution. In Prec. of 55th ICEST, 2020, pp. 167-170. DOI: 10.1109/ICEST49890.2020.9232788.
- [11]. Slimane F., Ingold R., Hennebert J. ICDAR2017 Competition on Multi-Font and Multi-Size Digitally Represented Arabic Text. In Prec. of 14th IAPR ICDAR, 2017, vol. 1, pp. 1466-1472. DOI: 10.1109/ICDAR.2017.239.
- [12]. Tatsukawa Y. et al. FontCLIP: A Semantic Typography Visual-Language Model for Multilingual Font Applications. Computer Graphics Forum, 2024, 43(2), p. e15043. DOI: 10.1111/cgf.15043.
- [13]. Phinney T. Variable Fonts Are the Next Generation. Communication Arts, 2016.

Информация об авторах / Information about authors

Даниил Евгеньевич КОПЫЛОВ – магистрант направления подготовки «Прикладная математика и информатика» Иркутского государственного университета, сотрудник Института динамики систем и теории управления имени В.М. Матросова Сибирского отделения Российской академии наук. Сфера научных интересов: прикладная математика, анализ данных.

Daniil Evgenievich KOPYLOV is master's student of Irkutsk State University, employee of Matrosov Institute for System Dynamics and Control Theory of Siberian Branch of Russian Academy of Sciences. Research interests: applied mathematics, data analysis.

Мария Викторовна ЩУРИК – бакалавр направления подготовки «Прикладная математика и информатика» Иркутского государственного университета. Сфера научных интересов: прикладная математика, анализ данных, искусственный интеллект.

Maria Viktorovna SHCHURIK is a bachelor's student of Irkutsk State University. Research interests: applied mathematics, data analysis, artificial intelligence.

