

Разработка алгоритма распознавания казахско-латинского алфавита на изображении

¹КУЛМАГАМБЕТОВА Жумажан Калдыгуловна, к.т.н., доцент, kulma_zh@mail.ru,

¹МҰРЗАҒҰЛОВ Дамир Тобылұлы, магистрант, murzagulov.damir97@gmail.com,

²*БИГАЛИЕВА Альфия Замировна, PhD, старший преподаватель, a.bigaliev@kstu.kz,

¹НАО «Актюбинский региональный университет имени К. Жубанова», Казахстан, Актобе, пр. А. Молдагуловой, 34,

²НАО «Карагандинский технический университет имени Абылкаса Сагинова», Казахстан, Караганда, пр. Н. Назарбаева, 56,

*автор-корреспондент.

Аннотация. Работа посвящена исследованию процесса распознавания алфавита на изображении. Целью является реализация нейронной сети на основе архитектуры свёрточных нейронных сетей для распознавания казахско-латинского алфавита. Рассматривается возможность применения нового алгоритма распознавания казахско-латинского алфавита на изображении с использованием свёрточной нейронной сети на практике. Сформирована свёрточная нейронная сеть, которая была обучена различать рукописные английские буквы и цифры, так как для казахско-латинского алфавита отличительной чертой являются специфичные символы и графические особенности. Произведено построение модели нейронной сети для предварительной обработки набора данных. На основе модели синтезирован проверочный набор данных, который используется для оценки и контроля качества модели во время обучения, и тестовый набор данных, предназначенный для окончательной оценки производительности модели после завершения процесса обучения. Представлена общая формулировка эффективного алгоритма, способного автоматически распознавать символы казахско-латинского алфавита на изображениях и представлять их в текстовом формате. Предложен новый подход, основанный на использовании глубоких свёрточных нейронных сетей.

Ключевые слова: алгоритм, текст, символ, распознавание, нейронная сеть, категориальная кроссэнтропия, пиксель, свёртка, казахско-латинский алфавит, матрица.

Введение

Мозг человека содержит миллиарды нейронов – узкоспециализированных клеток, предназначенных для приема, обработки, хранения, передачи и высвобождения информации извне с помощью электрических и химических сигналов [2].

На основе нейронов человеческого мозга была построена искусственная нейронная сеть. Искусственная нейронная сеть – это математическая модель, основанная на принципе организации и работы биологических нейронных сетей живого организма, включая его программную реализацию. Базовое понятие искусственных нейронных сетей – это искусственный нейрон, который сам по себе является соединителем всех поступающих в него сигналов [2].

Основными архитектурами нейронных сетей являются сети прямого распространения и персептроны. Эти нейронные сети очень просты [3].

Существуют также свёрточные нейронные сети. Этот тип нейронной сети сильно отличается от остальных. Свёрточные нейронные сети обыч-

но используются для классификации изображений. Главной особенностью этой нейронной сети является «сканер», который считывает изображение по частям, и процесс «свертки», который уменьшает размер матрицы признаков изображения [4].

Методы исследования

Для распознавания казахско-латинского алфавита на изображении с использованием свёрточной нейронной сети разработан и исследован новый алгоритм.

Основными этапами для реализации этой задачи были выделены следующие:

- 1) Обучение нейронной сети для распознавания букв казахско-латинского алфавита.
- 2) Определение на изображении букв.
- 3) Разбить изображение на части с буквами.
- 4) Распознать на отдельных изображениях находящиеся на них буквы.
- 5) Составить из полученных с изображений букв слово.
- 6) Расставить пробелы между разными

словами.

Для обучения модели была выбрана свёрточная нейронная сеть, которая была обучена различать рукописные английские буквы и цифры. Выбор свёрточной нейронной сети обусловлен тем, что на данный момент у данной разновидности нейронных сетей один из лучших алгоритмов распознавания и классификации изображений. По сравнению с полносвязной нейронной сетью, у нее гораздо меньше количество настраиваемых весов. Главной особенностью свёрточной нейронной сети является «свертка». Процесс «свертки» представляет собой уменьшение размера матрицы признаков входного изображения. Для получения ячейки матрицы уменьшенного размера элементы исходной матрицы в определённой области умножают на вес с последующим суммированием всех элементов в этой области. Чтобы получить следующую ячейку уменьшенной матрицы, происходит сдвиг области и выполнение тех же действий.

Описанную выше последовательность действий можно записать формулой:

$$(I \cdot K)_{xy} = \sum_{i=1}^h \sum_{j=1}^w K_{ij} \cdot I_{x+i-1,y+j-1}, \quad (1)$$

где I – исходная матрица признака, K – матрица веса, x и y – индексы выбранного блока, h и w – высота и ширина [5].

На рисунке 1 показан процесс свертки матрицы признака размерностью $7 \times 7 = 49$ признаков, которая обозначена буквой I . «Сетка» входной матрицы I размером 3×3 «клетки» умножается поэлементно на элементы матрицы весов K , после чего полученные значения матрицы суммируются, а полученное значение заносится в клетку выходной матрицы. Затем происходит сдвиг «сетки» и повторение вышеописанных действий.

В зависимости от типа проекта, использование различных наборов данных для обучения нейронных сетей может оказаться весьма полезным. Существует несколько популярных классификаций наборов данных, которые могут быть использованы в различных задачах машинного обучения.

Один из таких наборов данных – MNIST. Он представляет собой собрание изображений рукописных цифр от 0 до 9. MNIST широко используется для обучения нейронных сетей в задачах распознавания цифр.

Еще два популярных набора данных – CIFAR-10 и CIFAR-100. Они содержат изображения, разделенные на 10 и 100 классов соответственно. Эти наборы данных широко применяются в области классификации изображений.

Одним из самых масштабных наборов данных является ImageNet. Он включает более 1,4 миллиона изображений, относящихся к более чем 1000 классам. ImageNet является одним из наиболее популярных наборов данных для обучения глубоких нейронных сетей в области классификации

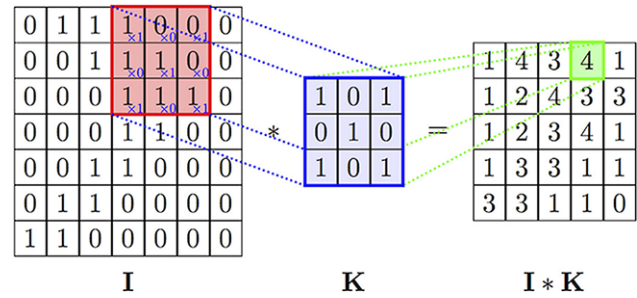


Рисунок 1 – Процесс «свертки»

изображений.

Если речь идет о задаче обнаружения объектов, то широко используется набор данных COCO. Он содержит более 330000 изображений, иллюстрирующих различные объекты в различных контекстах.

Для задач машинного перевода часто применяется набор данных WMT, который представляет собой сборник параллельных текстов на нескольких языках [6].

Однако иногда может возникнуть необходимость создать собственный набор данных. Изначально отсутствовала подходящая готовая база данных для казахско-латинского алфавита. По этой причине был разработан собственный набор данных, состоящий из 99736 изображений, сгенерированных при использовании 410 различных шрифтов. Этот набор данных может быть весьма полезен при обучении нейронных сетей в задачах, связанных с казахско-латинским алфавитом.

Этапы создания и обучения нейросетевой модели представлены ниже. Перед построением модели нейронной сети необходимо провести предварительную обработку набора данных (рисунок 2).

В первую очередь, изображения из готовой базы данных приводятся к определенному размеру. В данной работе размер изображений был изменен до 50×50 пикселей. Важно отметить, что входные изображения могут быть цветными. Поэтому перед использованием их в модели изображения преобразуются из цветовой модели RGB (красный, зеленый, синий) в градиентно-серую (рисунок 3).

Каждый пиксель в обработанном наборе данных имеет значение, которое находится в диапазоне от 0 до 255. Для обеспечения более эффективного обучения нейронной сети выполняется процесс нормализации данных. Нормализация нейронной сети заключается в приведении входных данных к стандартной нормализованной форме. Это позволяет повысить производительность нейронной сети и ускорить ее сходимость к оптимальному решению. Обычно данные нейронной сети имеют различный масштаб и диапазон значений. Это может вызвать проблемы при обучении, поскольку большие входные значения могут

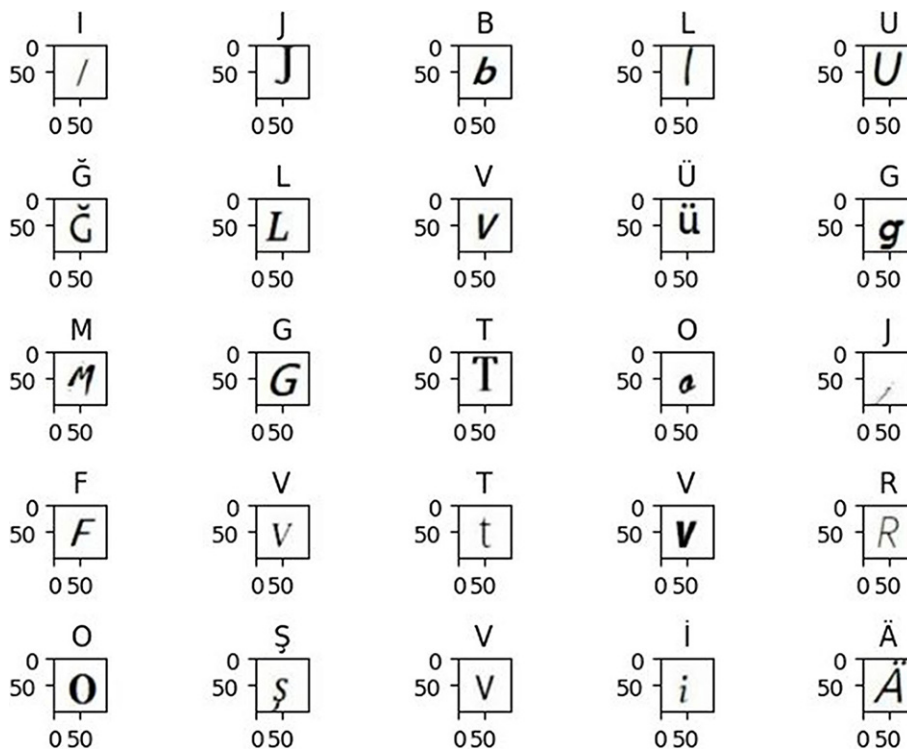


Рисунок 2 – Набор данных

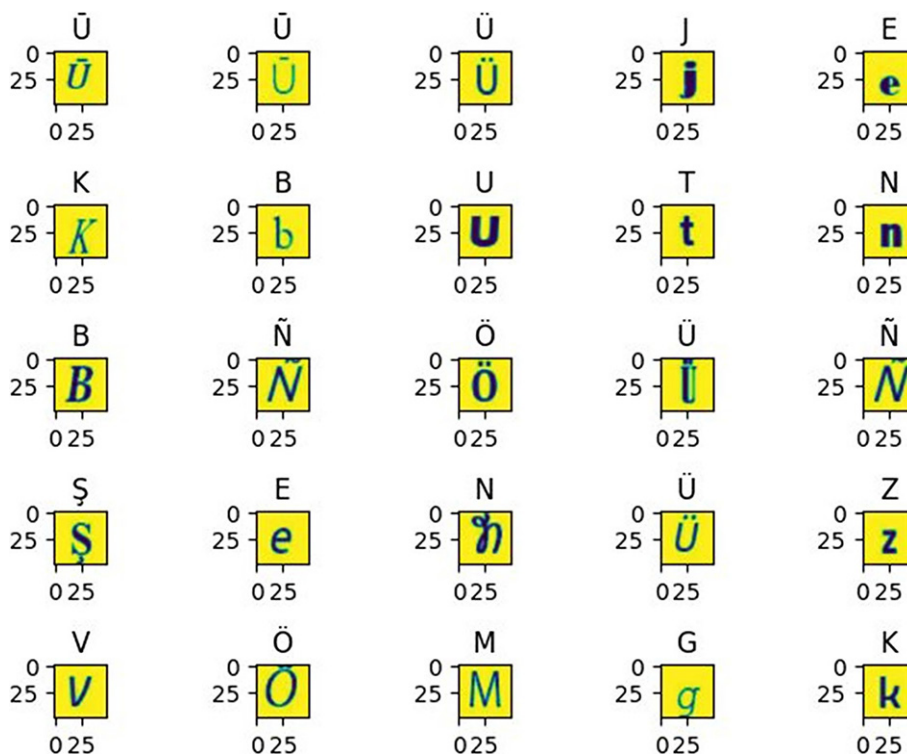


Рисунок 3 – Обработанный набор данных

привести к насыщению функции активации и замедлить сходимость нейронной сети. Кроме того, большие различия в масштабе входных данных могут снизить эффективность градиентного спуска в процессе обучения. Нормализация решает

эти проблемы путем преобразования входных данных в стандартный диапазон значений, например, от 0 до 1 или по стандартному нормальному распределению со средним значением 0 и стандартным отклонением 1. Такой подход делает гра-

диентный спуск более эффективным и быстрым [7].

После завершения процесса нормализации данных, происходит переход к важному этапу – разделению набора данных на три части: обучающий набор, проверочный набор и тестовый набор. Этот шаг необходим для построения надежной модели нейронной сети, позволяющей оценить качество работы модели на независимых данных и избежать проблемы переобучения.

Обучающий набор данных представляет собой набор, на котором модель будет обучаться. Проверочный набор данных используется для оценки и контроля качества модели во время обучения. Наконец, тестовый набор данных предназначен для окончательной оценки производительности модели после завершения процесса обучения. Существует несколько стратегий разделения набора данных на эти три части, но одним из наиболее распространенных подходов является случайное распределение. В данной работе использовалось соотношение 70% обучающего набора, 20% проверочного набора и 10% тестового набора.

Научные результаты

На ранних этапах исследования был проведен анализ различных типов и архитектур нейронных сетей. В результате была выбрана наиболее эффективная архитектура для алгоритма распознавания казахско-латинского алфавита – свёрточная архитектура (рисунок 4).

Следующим этапом в данной работе является создание модели нейронной сети. Строится модель, состоящая из нескольких слоев, для этого использовалась библиотека Keras.

С использованием класса Sequential в библиотеке Keras есть возможность создать последовательную коллекцию слоев, которые идут друг за другом (рисунок 5). В данной работе добавляются следующие параметры:

Слой Convolution2D: Этот слой позволяет создать свёрточный слой с ядром и получить выходной тензор. Здесь определяется размер ядра с помощью параметра kernel_size, где первый параметр указывает высоту, а второй – ширину окна свертки.

Слой Input_shape: Эта функция преобразует входное двумерное изображение размером 28 x 28 пикселей в одномерный массив из 784 пикселей. Активация 'relu' представляет функцию активации, которая используется для вычисления выходного сигнала нейрона. Функция активации необходима для введения нелинейности в модель.

Слой ReLU: Rectified Linear Unit (ReLU) представляет собой функцию активации, которая применяется к выходу предыдущего слоя. Формула функции ReLU выглядит следующим образом:

$$f(x) = \max(0, x). \quad (2)$$

Функция ReLU помогает в моделировании нелинейных отношений и обеспечивает более гибкую модель (рисунок 6).

Слой ZeroPadding2D: Свёрточные нейронные сети используют заполнение нулями для управления размером вывода после операции свертки. При свертке размерность данных обычно уменьшается. Однако это может привести к потере информации по краям изображений и ограничению видимости нейронной сети. Для избегания этой

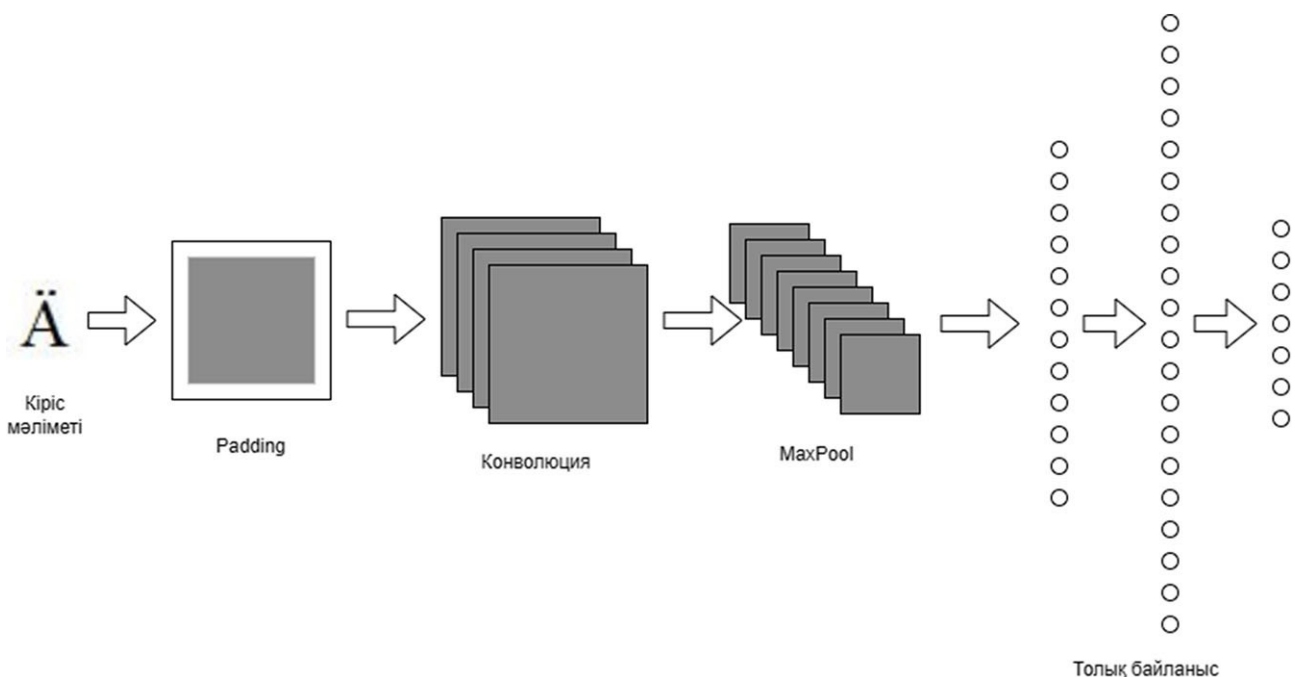


Рисунок 4 – Архитектура нейронной сети

```

model = Sequential()

model.add(ZeroPadding2D(padding=(1,1)))
model.add(Conv2D(8,(3,3),1, activation="relu", input_shape=(50,50,1)))
model.add(MaxPooling2D())

model.add(ZeroPadding2D(padding=(1,1)))
model.add(Conv2D(16,(3,3),1, activation="relu"))
model.add(MaxPooling2D())

model.add(ZeroPadding2D(padding=(1,1)))
model.add(Conv2D(32,(3,3),1, activation="relu"))
model.add(MaxPooling2D())

model.add(Flatten())

model.add(Dense(248, activation="relu"))
model.add(Dense(128, activation="relu"))
model.add(Dense(31, activation="softmax"))

model.compile("adam", loss=tf.losses.CategoricalCrossentropy(), metrics=["accuracy"])
    
```

Рисунок 5 – Программа создания модели

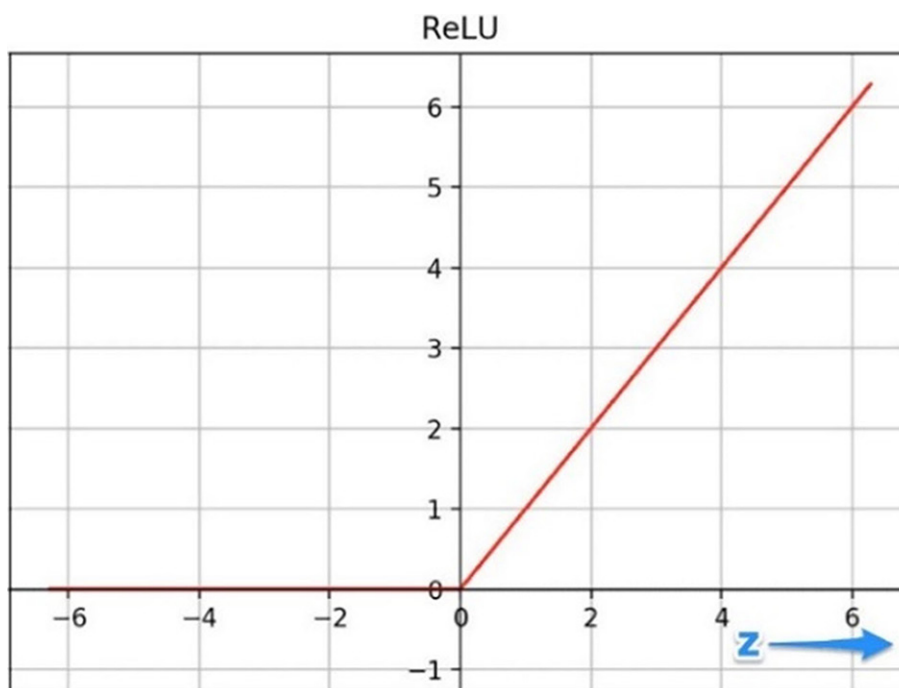


Рисунок 6 – График активации «ReLU»

проблемы используется слой ZeroPadding2D, который добавляет нули вокруг входных данных перед применением операции свертки. Это позволяет сохранить размерность данных и предотвратить сужение поля зрения.

Слой MaxPooling2D используется в нейронных сетях для обработки двумерных данных, таких как изображения. Он выполняет уменьше-

ние размерности пространства признаков, выбирая наибольшее значение из каждой подобласти входных данных (обычно размером 2x2 или 3x3) и перемещаясь по изображению с заданным шагом (2 или 3). Это позволяет уменьшить количество параметров в модели, упростить вычисления и избежать переобучения, выделяя наиболее важные особенности изображения.

Слой Flatten является промежуточным слоем, который преобразует многомерный входной тензор в одномерный тензор. Обычно этот слой размещается перед полносвязными слоями (Dense) нейронной сети. Например, если размер входного тензора составляет (размер пакета, высота, ширина, каналы), то после применения слоя Flatten получается одномерный тензор размерности (размер пакета, высота * ширина * каналы). Этот слой позволяет преобразовать выходные данные предыдущих слоев в форму, которая может быть использована полносвязными слоями, ожидающими одномерный входной тензор. Кроме того, слой Flatten может уменьшить размерность входных данных, что способствует упрощению модели и уменьшению вычислительной сложности.

Слой Dense, также известный как полносвязный слой, является типом слоя в нейронной сети, где каждый нейрон входного слоя связан с каждым нейроном выходного слоя. Это означает, что все нейроны входного слоя соединены со всеми нейронами выходного слоя. Во время обучения нейронной сети слой Dense выполняет линейное преобразование входных данных, взвешивая каждый вход с помощью соответствующей функции, а затем применяет нелинейную функцию активации (например, ReLU, Sigmoid, Tanh) для получения выходных данных.

$$\text{soft max}(z_i) = \frac{e^{z_i}}{\sum_{j=1}^k e^{z_j}}. \quad (3)$$

Функция активации Softmax, представленная формулой (3), широко используется в задачах классификации множества классов. Она принимает входной вектор z и вычисляет вероятности для каждого класса i , масштабируя индекс z_i относительно суммы всех индексов. Таким образом, Softmax функция преобразует выходные значения в вероятности, где каждое значение представляет вероятность принадлежности к соответствующему классу [8].

Количество классов обозначается переменной k .

Это очень важно, так как функция Softmax позволяет получить вероятности для каждого класса, где сумма всех вероятностей равна 1. Это полезно, поскольку нейронная сеть может предсказывать вероятности для каждого класса, а не только определять наиболее вероятный класс. В процессе обучения нейронной сети с использованием функции активации Softmax обычно минимизируются кросс-энтропийные потери, которые измеряют расхождение между предсказанными вероятностями и фактическими метками классов.

При компиляции модели определяют факторы, влияющие на дальнейший процесс обучения нейронной сети. В частности, метод компиляции принимает следующие важные параметры:

Оптимизатор: это алгоритм оптимизации, используемый во время обучения. Здесь есть воз-

можность выбора из различных оптимизаторов, таких как Адам, SGD (стохастический градиентный спуск) и других. Каждый оптимизатор имеет свои особенности и может быть более или менее подходящим для конкретной задачи обучения нейронной сети.

Функция потерь: это функция, которая измеряет несоответствие между выходными значениями нейронной сети и целевыми значениями. Например, для задачи двоичной классификации есть возможность выбора функции потерь «binary_crossentropy», а для задачи классификации нескольких классов – «categorical_crossentropy». Выбор подходящей функции потерь важен для успешного обучения модели.

Метрики: это метрики, используемые для оценки производительности модели во время обучения. Например, выбор метрики «accuracy» (точность), которая измеряет долю правильно классифицированных образцов, или другие метрики, такие как «precision» (точность), «recall» (полнота) и т.д. Выбор подходящих метрик зависит от конкретной задачи и требований к модели.

Метод оптимизации Adam (Adaptive Moment Estimation) является эффективным способом обучения нейронных сетей. Он объединяет преимущества двух других методов – AdaGrad и RMSProp. Алгоритм Adam использует первый и второй моменты градиента для настройки скорости обучения и момента. При использовании Adam вычисляется градиент функции потерь с помощью весов, обновляем первый и второй моменты градиента с помощью экспоненциального скользящего среднего, рассчитываем скорость обучения и время на текущем шаге оптимизации и обновляем веса с учетом рассчитанной скорости обучения и импульса. Adam имеет преимущество в том, что он может изменять скорость обучения в разных частях функции потерь, что позволяет достичь оптимальных значений функции потерь. Этот метод считается одним из наиболее эффективных для оптимизации нейронных сетей.

Категориальная кроссэнтропия – это функция потерь, широко используемая в задачах классификации с несколькими классами. В нейронных сетях она используется для измерения разницы между предсказанными вероятностями классов и фактическими метками классов. Когда модель нейронной сети предсказывает вероятности принадлежности к каждому классу, категориальная кроссэнтропия вычисляет сумму логарифмов предсказанных вероятностей отрицательных истинных классов. Это позволяет оценить затраты на классификацию и минимизировать их [9].

Выводы

В ходе исследования были проанализированы существующие методы распознавания текста и основные методы машинного обучения. Также была реализована нейронная сеть на основе архитектуры сверточных нейронных сетей для распоз-

навания казахского латинского алфавита. Данная задача требовала предварительной обработки и подготовки данных, а также выбора подходящей архитектуры нейронной сети. Сверточные ней-

ронные сети являются мощным инструментом для работы с изображениями и текстовыми данными, поэтому они были выбраны для решения данной задачи.

СПИСОК ЛИТЕРАТУРЫ

1. Oralbayeva N. Amirova A. Telisheva Zh. Zhanatkyzy A. Aimysheva A. Sandygulova A. Montessori-based Design of Long-term Child-Robot Interaction for Alphabet Learning // ACM/IEEE International Conference on Human-Robot Interaction, 2023. – Pp. 691-695.
2. Омеляненко Я. Эволюционные нейросети на языке Python. – М.: ДМК-Пресс, 2020. – 310 с.
3. Кук Д. Машинное обучение с использованием библиотеки H2O. – М.: ДМК-Пресс, 2018. – 252 с.
4. Шарден Б., Боскетти А., Массарон Л. Крупномасштабное машинное обучение вместе с Python. – М.: ДМК-Пресс, 2018. – 360 с.
5. Что такое свёрточная нейронная сеть [Электронный ресурс]. – 2016. <https://habr.com/ru/post/309508>
6. Thevenoux R., LE V.L., Villessèche H., Buisson A., Beurton-Aimar M., Grenier E., Folcher L., Parisey N. Image based species identification of Globodera quarantine nematodes using computer vision and deep learning [Электронный ресурс] // 2021. <https://doi.org/10.1016/j.compag.2021.106058>
7. Sukanya J., Rajiv Gandhi K., Palanisamy V. An assessment of machine learning algorithms for healthcare analysis based on improved MapReduce. [Электронный ресурс] // 2022. <https://doi.org/10.1016/j.advengsoft.2022.103285>
8. Николенко С., Архангельская Е., Кадуринов А. Глубокое обучение. – СПб: Питер, 2020. – 480 с.
9. Гудфеллоу Я., Курвилль А., Бенджио И. Глубокое обучение. – М.: ДМК-Пресс, 2018. – 654 с.

Кескіндегі қазақ-латын әліпбиін тану алгоритмін әзірлеу

¹**КУЛМАГАМБЕТОВА Жумажан Калдыгуловна**, т.ғ.к., доцент, kulma_zh@mail.ru,

¹**МҰРЗАҒҰЛОВ Дамир Тобылұлы**, магистрант, murzagulov.damir97@gmail.com,

²***БИГАЛИЕВА Альфия Замировна**, PhD, аға оқытушы, a.bigaliev@kstu.kz,

¹«Қ. Жұбанов атындағы Ақтөбе өңірлік университеті» КеАҚ, Қазақстан, Ақтөбе, Ә. Молдағұлова даңғылы, 34,

²«Әбілқас Сағынов атындағы Қарағанды техникалық университеті» КеАҚ, Қазақстан, Қарағанды,

Н. Назарбаев даңғылы, 56,

*автор-корреспондент.

Аңдатпа. Жұмыс суреттегі алфавитті тану процесін зерттеуге арналған. Мақсаты қазақ латын әліпбиін тану үшін конволюциялық нейрондық желілердің архитектурасы негізінде нейрондық желіні іске асыру болып табылады. Қазақ-латын әліпбиін танудың жаңа алгоритмін суретте конволюциялық нейрондық желіні қолдана отырып қолдану мүмкіндігі қарастырылуда. Конвульсиялық нейрондық желі құрылды, ол қолжазба ағылшын әріптері мен сандарын ажыратуға үйретілді, өйткені қазақ-латын әліпбиі үшін ерекше белгілер мен графикалық ерекшеліктер айрықша белгі болып табылады. Деректер жиынтығын алдын-ала өңдеу үшін нейрондық желі моделі жасалды. Модель негізінде оқыту кезінде модельдің сапасын бағалау және бақылау үшін қолданылатын тексеру деректер жиынтығы және оқу процесі аяқталғаннан кейін модельдің өнімділігін түпкілікті бағалауға арналған сынақ деректер жиынтығы синтезделді. Суреттердегі қазақ-латын әліпбиінің таңбаларын автоматты түрде тануға және оларды мәтіндік форматта ұсынуға қабілетті тиімді алгоритмнің жалпы тұжырымы ұсынылған. Терең конволюциялық нейрондық желілерді қолдануға негізделген жаңа тәсіл ұсынылды.

Кілт сөздер: алгоритм, мәтін, символ, тану, нейрондық желі, категориялық кроссентропия, пиксель, конволюция, қазақ-латын әліпбиі, матрица.

Development of an Algorithm for Recognition of the Kazakh-Latin Alphabet in the Image

¹**KULMAGAMBETOVA Zhumazhan**, Cand. of Tech. Sci., Associate Professor, kulma_zh@mail.ru,

¹**MURZAGULOV Damir**, Master Student, murzagulov.damir97@gmail.com,

²***BIGALIYEVA Alfija**, PhD, Senior Lecturer, a.bigaliev@kstu.kz,

¹NPJSC «K. Zhubanov Aktobe Regional University», Kazakhstan, Aktobe, A. Moldagulova Avenue, 34,

²NPJSC «Abylqas Saginov Karaganda Technical University», Kazakhstan, Karaganda, N. Nazarbayev Avenue, 56,

*corresponding author.

Abstract. The work is devoted to the study of the alphabet recognition process in the image. The goal is to implement a neural network based on the architecture of convolutional neural networks for recognition of the Kazakh Latin alphabet. The possibility of applying a new algorithm for recognizing the Kazakh-Latin alphabet in an image using a convolutional

neural network in practice is considered. A convolutional neural network was formed, which was trained to distinguish handwritten English letters and numbers, since specific symbols and graphic features are a distinctive feature for the Kazakh-Latin alphabet. A neural network model has been constructed for preprocessing a data set. Based on the model, a test data set is synthesized, which is used to evaluate and control the quality of the model during training, and a test data set intended for final evaluation of the model's performance after the training process is completed. The general formulation of an effective algorithm capable of automatically recognizing the symbols of the Kazakh-Latin alphabet in images and presenting them in text format is presented. A new approach based on the use of deep convolutional neural networks is proposed.

Keywords: *algorithm, text, symbol, recognition, neural network, categorical crossentropy, pixel, convolution, Kazakh-Latin alphabet, matrix.*

REFERENCES

1. Oralbayeva N. Amirova A. Telisheva Zh. Zhanatkyzy A. Aimysheva A. Sandygulova A. Montessori-based Design of Long-term Child-Robot Interaction for Alphabet Learning // ACM/IEEE International Conference on Human-Robot Interaction, 2023. – Pp. 691-695.
2. Omel'yanenko YA. Evolyucionnye nejroseti na yazyke Python. – Moscow: DMK-Press, 2020. – 310 p.
3. Kuk D. Mashinnoe obuchenie s ispol'zovaniem biblioteki H2O. – Moscow: DMK-Press, 2018. – 252 p.
4. SHarden B., Bosketti A., Massaron L. Krupnomasshtabnoe mashinnoe obuchenie vmeste s Python. – Moscow: DMK-Press, 2018. – 360 p.
5. CHto takoe svyortochhnaya nejronnaya set' [Elektronnyj resurs]. – 2016. <https://habr.com/ru/post/309508>
6. Thevenoux R., LE V.L., Villessèche H., Buisson A., Beurton-Aimar M., Grenier E., Folcher L., Parisey N. Image based species identification of Globodera quarantine nematodes using computer vision and deep learning [Elektronnyj resurs] // 2021. <https://doi.org/10.1016/j.compag.2021.106058>
7. Sukanya J., Rajiv Gandhi K., Palanisamy V. An assessment of machine learning algorithms for healthcare analysis based on improved MapReduce. [Elektronnyj resurs] // 2022. <https://doi.org/10.1016/j.advengsoft.2022.103285>
8. Nikolenko S., Arhangel'skaya E., Kadurin A. Glubokoe obuchenie. – Saint Petersburg: Piter, 2020. – 480 p.
9. Gudfellow YA., Kurvil' A., Bendzhio I. Glubokoe obuchenie. – Moscow: DMK-Press, 2018. – 654 p.