

<sup>1</sup>Бактыбеков М. Б., <sup>2</sup>Кокумбаев Ш. Б., Султанов Р. К.

<sup>1,2</sup>"МАНАС" КТУнун компьютердик инженерия бөлүмүнүн магистранттары  
ф.-м.и.к., "МАНАС" КТУнун компьютердик инженерия кафедрасы доценти

## ТЕГИЗДИКТЕГИ ГЕОМЕТРИЯЛЫК ФИГУРАЛАРДЫ ТААНУУНУН НЕГИЗИНДЕ МАШИНА ҮЙРӨТҮҮ КЛАССИФИКАТОРЛОРУНУН САЛЫШТЫРМАЛУУ ТАЛДООСУ

### СРАВНИТЕЛЬНЫЙ АНАЛИЗ КЛАССИФИКАТОРОВ МАШИННОГО ОБУЧЕНИЯ НА ОСНОВЕ РАСПОЗНАВАНИЯ ГЕОМЕТРИЧЕСКИХ ФИГУР НА ПЛОСКОСТИ

#### COMPARATIVE ANALYSIS OF MACHINE LEARNING CLASSIFIERS BASED ON RECOGNITION OF GEOMETRIC FIGURES ON A PLANE

**Аннотация:** Бул макалада эң белгилүү болгон машина үйрөтүү классификаторлорунун салыштырма талдоосу келтирилген: конволюциялык нейрон тармактары (CNN); таяныч вектордук машина (SVM); k-жакынкы кошуналар ыкмасы (k-NN). Негизги маалымат берилиштери катары геометриялык фигуралар алынган: үч бурчтук, төрт бурчтук, тегерек жана жылдызча. Айтылган классификаторлорду үйрөтүү жана таануу талдоолору жүргүзүлгөн, ошондой эле берилген маалыматтардын машинаны үйрөтүүдөгү ролу изилделген.

**Түйүндүү сөздөр:** машина үйрөтүү; классификаторлор; конволюциялык нейрон тармактары; таяныч вектордук машина; k-жакынкы кошуналар ыкмасы

**Аннотация:** В данной статье проводится сравнительный анализ наиболее известных классификаторов машинного обучения: сверточных нейронных сетей (CNN); метода опорных векторов (SVM); метода k-ближайших соседей (k-NN). Основными входными данными были взяты геометрические фигуры: треугольник, четырехугольник, окружность и звезда. Проведен сравнительный анализ обучения и распознавания упомянутых классификаторов, а также роль входных данных в качестве обучения.

**Ключевые слова:** машинное обучение; классификаторы; сверточные нейронные сети; метод опорных векторов; метод k-ближайших соседей

**Annotation:** This article provides a comparative analysis of the most famous machine learning classifiers: convolutional neural networks (CNN); support vector machine (SVM); k-nearest neighbors method (k-NN). Geometric shapes were taken as the main input data: triangle, quadrangle, circle and star. A comparative analysis of training and recognition of the mentioned classifiers, as well as the role of input data in the training quality.

**Key words:** machine learning; classifiers; convolutional neural networks; support vector machine; k-nearest neighbors

#### Введение

С ростом быстродействия скорости обмена информацией, мощности вычислений и доступности большим базам данных алгоритмы обнаружение и распознавание объектов на

основе компьютерного зрения является одним из быстро развивающихся направлений в современном этапе развития прикладной науки. Понятно, что применения успехов компьютерного зрения становится актуально во всех сферах от микромира до космических объектов. Например: использование в беспилотниках [1]; обнаружение машин [2]; распознавание людей; слежение за жестами человека [3]; распознавание автомобильных номеров [4].

В этой статье проводится сравнительный анализ наиболее известных классификаторов машинного обучения: Сверточные нейронные сети (Convolutional Neural Network - CNN); Метод опорных векторов (Support Vector Machine - SVM); Метод k-ближайших соседей (K-Nearest Neighbor classifier - k-NN) на основе идентичных входных данных (набор изображений) составленных из геометрических фигур (треугольника, четырехугольника, окружности и звезды). Указанные выше классификаторы используются для решений проблем компьютерного зрения в сфере распознавания объектов.

### **Обзор литературы**

В теме обнаружения и распознавания объектов было произведено много исследований и написано много статей. Ниже приведены некоторые из этих статей и их краткое описание.

В статье [5] изучена гибридная модель интеграции двух самых популярных классификаторов - сверточной нейронной сети (CNN) и машины опорных векторов (SVM) для распознавания рукописного текста. Они доказали результаты в распознавании паттернов разных типов. Модель определяет, делает CNN обучаемым экстрактором признаков, а SVM выступает в качестве распознавателя. Эта гибридная модель автоматически извлекает элементы из необработанных изображений с помощью CNN и создает прогнозы с помощью SVM. Хорошо известная цифровая база данных MNIST содержит эксперименты.

В статье [6] предложена новая схема обнаружения объектов на основе форм для выделения и кластеризации краев в изображениях с использованием метода градиентной векторной опоры (GVG), который приводит к ориентированному графу обнаруженных краев.

В статье [7] был произведен сравнительный анализ готовых моделей SSD-MobileNet-v2, Faster R-CNN-ResNet-101 и R-FCN-ResNet-101 при помощи Tensorflow.

А в статье [8] был произведен сравнительный анализ четырех классификаторов машинного обучения, таких как, support vector machine (SVM), normal Bayes (NB), classification and regression tree (CART) и K nearest neighbor (KNN) с помощью больших изображений.

### **Методология**

Как было написано ранее, в этой статье будет выполнен сравнительный анализ классификаторов машинного обучения: Convolutional Neural Network (CNN); Support Vector Machine (SVM); K-Nearest Neighbor classifier (k-NN). Для обучения классификаторов машинного обучения понадобится набор изображений, содержащий в себе геометрические фигуры на плоскости. С помощью данного набора будут проводиться обучение и тестирование. Каждый из классификаторов пройдет обучение на основе идентичного набора изображений. В качестве эксперимента будут предложены на обнаружение и распознавание часть изображений, ранее не включенных в набор изображений в процессе обучения, а также изображения из того же набора изображений. В конце пройдет сравнение и анализ полученных результатов.

Для реализации Support Vector Machine (SVM); K-Nearest Neighbor classifier (k-NN) была выбрана библиотека **OpenCV**, которая поддерживает оба классификатора [9]. Для реализации Convolutional Neural Network (CNN) была выбрана библиотека **TensorFlow** [10].

### Метод опорных векторов - Support Vector Machines (SVM)

SVM - это алгоритм обучения с учителем, который в основном используется для классификации, но его также можно использовать для регрессии. Основная идея заключается в том, что на основе помеченных данных (обучающих данных) алгоритм пытается найти оптимальную гиперплоскость, которую можно использовать для классификации новых точек данных. В двух измерениях гиперплоскость представляет собой простую линию.

Обычно алгоритм обучения пытается изучить наиболее общие характеристики (то, что отличает один класс от другого) класса, и классификация основана на этих изученных репрезентативных характеристиках (поэтому классификация основана на различиях между классами). SVM работает наоборот. Он находит наиболее похожие примеры между классами. Это будут опорные векторы. SVM используется в много приложений в разных областях, таких как распознавание лиц, биоинформатика и обработка изображений [11].

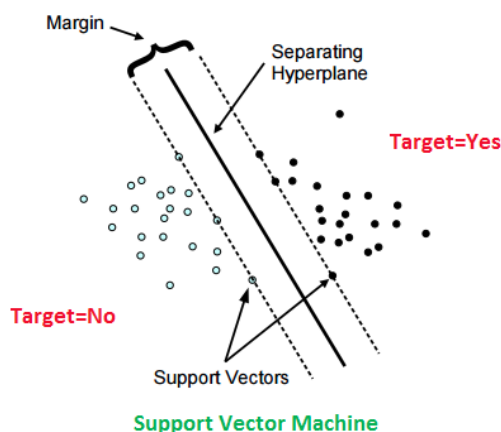
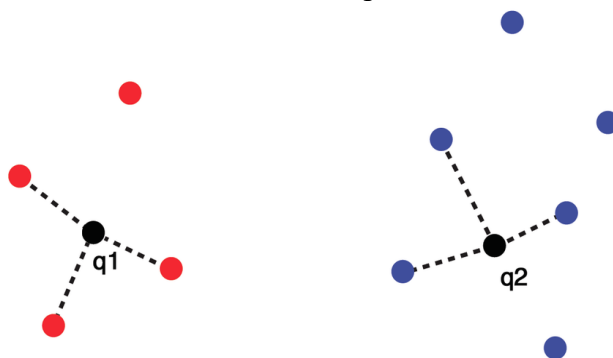


Рисунок 1. Метод опорных векторов

### Метод k-ближайших соседей- K-Nearest Neighbor classifier (k-NN)

K-nearest neighbour (KNN) является одним из самых простых алгоритмов в машинном обучении. В классификации данных данный метод является методом без параметров (непараметрический метод). В данном методе классификация происходит следующим образом. Узнается как точка данных примерно относится к одному или другому классу в зависимости от того, какой набор точек данных находятся самые близкие к нему. После этого он получает набор точек в пространстве, а затем считает две аналогичные точки, чтобы расстояние между ними в этом пространстве были в правильных метриках. Тогда алгоритм работает, решая какие из этих пунктов учебного набора более похожи, учитывая при выборе класса для прогнозирования нового наблюдения, тем самым выбирая k ближайших точек данных к этому наблюдению и давая самый общий среди этих классов. Таким образом, положительное целое число установлено для k, кроме нового образца. Затем k записей выбираются в базе данных, которые являются ближайшими к новому образцу. Тогда находится самая распространенная классификация среди них. Наконец, это будет

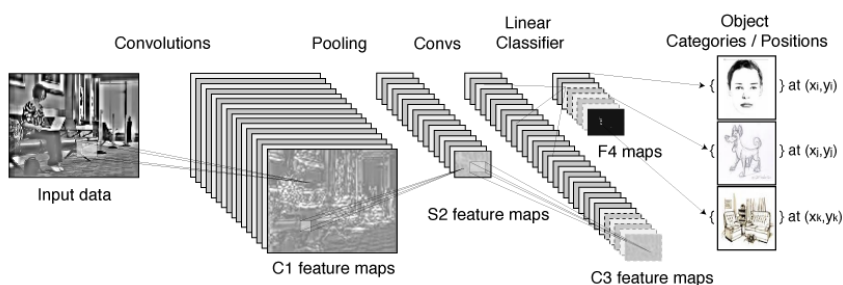
классификация, которой будет дан новый образец. На рисунке показан пример использования большинства голосов в кНН для определения классов в набор данных [12].



**Рисунок 2.** Метод k-ближайших соседей

### Сверточные нейронные сети - Convolutional Neural Network (CNN)

Сверточные нейронные сети - это класс глубокого обучения который в основном используется для классификации изображений, кластеризации изображения по их сходству, а также выполнение распознавания объектов в представлениях. CNN состоит из нейронов, которые самооптимизируются путем обучения. Архитектура CNN состоит из трех слоев; сверточный слой, объединяющий слой и полностью связанный слой.



**Рисунок 3.** Сверточные нейронные сети

Алгоритмы CNN имеют два ключевых процесса; свертка и отбор проб. Первый процесс фильтр, который может быть обученный используется на входном изображении, и он свернут, и это изображение объекта каждого слоя, которое называется картой объектов, а затем к этому добавляется уклон; это происходит в сверточном слое. Этот сверточный слой определяет выход нейронов, которые связаны с местными регионами ввода путем расчета произведения между их весами и областью, связанной с входной громкостью. Выпрямленная линейная единица (ReLU) выполняет поэлементную функцию активации на выходе, которая производится на предыдущем слое. Слой пула, где происходит процесс выборки. Полностью связанный слой пытается создать баллы за активацию, которые затем могут быть использованы для классификации [13].

#### Основная часть

##### Создание входных данных - dataset creation

Для эксперимента был произведен поиск готового набора изображений геометрических фигур в свободном доступе. Такой набор данных был найден[14], он состоял из 16000 изображений (треугольник, четырехугольник, окружность и звезда, каждый по 4000 изображений).



При работе с данным набором изображений выяснилось, что изображение в данном наборе имеют схожий размер и расположение относительно центра изображения. В ходе обучения и распознавания средняя точность распознавания была равна 95.92% при использовании изображений из этого же набора (Таблица 1), но при тестировании изображений, ранее не использованных ранее в обучении, процент распознавания стал сильно низким 23.3% (Таблица 2)

**Таблица 1 - Обучение и тестирование с одним и тем же набором изображений**

Фигура	Точность распознавания %	Ошибочность распознавания %
Треугольник	97.1	2.9
Четырехугольник	99.9	0.1
Окружность	86.7	13.3
Звезда	100	0
Среднее	95.92	4.08

Было замечено, что каждый элемент был в основном распознан как треугольник. Ниже представлены результаты обучения на основе готового набора изображений:

**Таблица 2 - Обучение и тестирование с разным набором изображений**

Фигура	Точность распознавания %	Ошибочность распознавания %
Треугольник	61.7	38.3
Четырехугольник	14.2	85.8
Окружность	0.3	99.7
Звезда	16.9	83.1
Среднее	23.27	76.73

По причине схожести изображений в готовом наборе и низкой точности распознавания, было решено синтезировать собственный набор изображений с помощью библиотеки Python Imaging Library (PIL) [15]. Размер изображений был выбран 100x100 пикселей.

В итоге получили новый набор изображений с различными геометрическими объектами и с различным расположением на изображении. Созданный набор состоит из 60 000 изображений для обучения и 4000 изображений для тестирования (Таблица 3).

**Таблица 3 - Готовые и синтезированные наборы изображений**

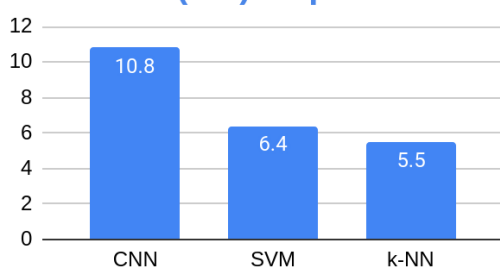
Фигуры	Заявленное кол-во изображений в готовом наборе	Фактическое кол-во изображений в наборе	Кол-во изображений из готового набора	Кол-во изображений из готового набора	Кол-во синтезированных изображений для	Кол-во синтезированных изображений для тестов

	наборе	готовом наборе	для обучения	для тестов	обучения	
Треугольник	4000	3720	2000	1000	15000	1000
Четырехугольник	4000	3765	2000	1000	15000	1000
Окружность	4000	3720	2000	1000	15000	1000
Звезда	4000	3765	2000	1000	15000	1000
Общее	<b>16000</b>	<b>14902</b>	<b>8000</b>	<b>4000</b>	<b>60000</b>	<b>4000</b>

## Обучение и эксперименты

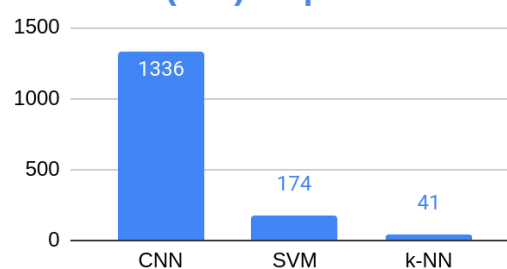
Для обучения классификаторов были взяты 8000 изображений из готового набора и 60000 изображений из синтезированных. В итоге каждый классификатор был обучен на основе 68000 изображений и количество эпох во процессе обучение была равно семи. Также в процессе исследование также был произведен анализ используемых ресурсов (времени потраченным на обучение (Рисунок 4) и объемом потребленной оперативной памяти (Рисунок 5)) каждым алгоритмом в процессе обучения.

**RAM used (GB) 7 epoch**



**Рисунок 4.** Использование ОЗУ при обучении

**Runtime (sec) 7 epoch**



**Рисунок 5.** Потраченное время при обучении

В ходе эксперимента были использованы изображения из набора ранее не использованных в процессе обучения, то есть обученная машина должна была распознать данное изображение на основе схожих с ней изображений. 4000 изображений были взяты из готового набора и 4000 из синтезированных. Итого 8000 изображений геометрических фигур было подготовлено для распознавания.

## Результаты и обсуждение

Для сравнения результатов классификации было решено использовать матрицу ошибок [16] и её метрики. Матрица ошибок — это таблица, которая позволяет визуализировать эффективность алгоритма классификации путем сравнения прогнозируемого значения целевой переменной с её фактическим значением. Столбцы матрицы представляют наблюдения в прогнозируемом классе, а строки — наблюдения в фактическом классе (или наоборот).

### Название и формулы для метрик

**Accuracy или Classification Rate (Процент классификации)** - Процент целей, точно классифицированных моделью при применении к набору проверочных данных.

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

**Recall (Возврат)** - Процент фактических положительных целей, которые были правильно спрогнозированы.

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

**Precision (Точность)** - Процент прогнозных положительных целей, которые фактически являются положительными целями.

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

**F-Measure (F-Оценка)** - Среднее гармоническое от точности и возврата (с равными весами возврата и точности).

$$F - \text{measure} = \frac{2 * \text{Recall} * \text{Precision}}{\text{Recall} + \text{Precision}}$$

Где, TP (True Positive, истинно положительные) - число корректно спрогнозированных положительных целей.

FN (False Negative, ложноотрицательные) - число фактических положительных целей, спрогнозированных как отрицательные.

FP (False Positive, ложноположительные) - число фактических отрицательных целей, спрогнозированных как положительные.

TN (True Negative, истинно отрицательные) - число корректно спрогнозированных отрицательных целей.

**Общая точность** рассчитывается как общее количество правильно классифицированных значений, деленное на общее количество тестовых значений.

По результатам обучения классификаторов и последующим экспериментом по распознаванию геометрических фигур были получены следующие данные:

В таблице показана матрица ошибок для классификатора K-Nearest Neighbor classifier (k-NN)

**Таблица 4 - Матрица ошибок для k-NN**

	Треугольник	Четырехугольник	Окружность	Звезда
Треугольник	1752	101	47	100
Четырехугольник	154	1818	8	20
Окружность	23	1	1766	210
Звезда	73	12	488	1427

Общая точность: 84.54%

**Таблица 5 - Метрики матрицы ошибок для k-NN**

Class	n (classified)	Accuracy	Precision	Recall	F-Measure
Треугольник	2000	93.78%	0.88	0.88	0.88
Четырехугольник	2000	96.30%	0.91	0.94	0.92
Окружность	2000	90.29%	0.88	0.76	0.82
Звезда	2000	88.71%	0.71	0.81	0.76

В таблице показана матрица ошибок для классификатора Support Vector Machine (SVM)

**Таблица 6 - Матрица ошибок SVM**

	Треугольник	Четырехугольник	Окружность	Звезда
Треугольник	1424	395	15	166
Четырехугольник	1096	887	8	9
Окружность	90	2	1488	420
Звезда	159	12	379	1450

Общая точность: 65.61%

**Таблица 7 - Метрики матрицы ошибок для SVM**

Class	n (classified)	Accuracy	Precision	Recall	F-Measure
Треугольник	2000	75.99%	0.71	0.51	0.6
Четырехугольник	2000	80.98%	0.44	0.68	0.54
Окружность	2000	88.58%	0.74	0.79	0.77
Звезда	2000	85.69%	0.72	0.71	0.72

В таблице показана матрица ошибок для классификатора Convolutional Neural Network (CNN)

**Таблица 8 - Матрица ошибок для CNN**

	Треугольник	Четырехугольник	Окружность	Звезда
Треугольник	1710	146	144	0
Четырехугольник	46	1893	57	4
Окружность	3	40	1957	0
Звезда	0	1	0	1999

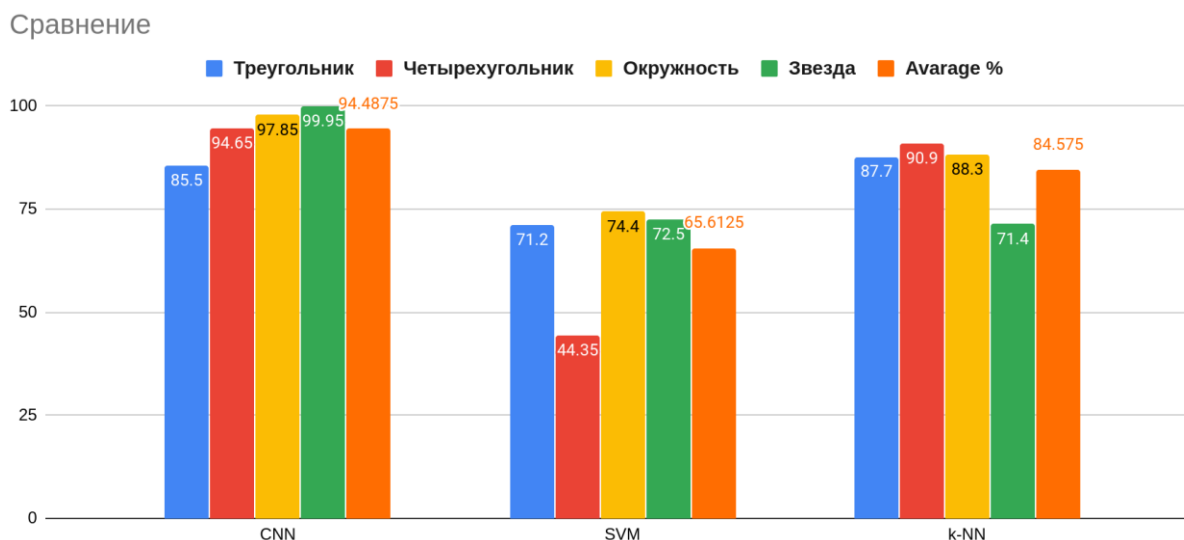
Общая точность: 94.49%

**Таблица 9 - Метрики матрицы ошибок для CNN**

Class	n (classified)	Accuracy	Precision	Recall	F-Measure
Треугольник	2000	95.76%	0.85	0.97	0.91
Четырехугольник	2000	96.33%	0.95	0.91	0.93
Окружность	2000	96.95%	0.98	0.91	0.94
Звезда	2000	99.94%	1.0	1.0	1.0



Ниже приведена сравнительная таблица всех классификаторов для каждой фигуры и их средняя точность распознавания (рисунок 6)



**Рисунок .** Сравнительная таблица точности распознавания для классификаторов

**Заключение:** При обучении и тестировании на основе набора изображений взятых из открытого доступа, точность распознавания была высокой только для изображений из этого же набора изображений, а вот для иных изображений точность распознаваний была неудовлетворительной. Использование однотипных изображений привело бы к заучиванию обучающейся машины, и в следствие этого, к большому количеству ошибочных распознаваний.

Как показал приведенный эксперимент с разным типом набора изображений, правильно подобранный и содержащий в себе большое количество уникальных изображений, играет большую роль в обучении машины.

На основе синтезированного набора изображений, состоящих из четырех видов геометрических фигур (треугольник, четырехугольник, окружность и звезда), удалось достичь точности распознавания для сверточных нейронных сетей (CNN) - 94.49%, Метода опорных векторов (SVM) - 65.61% и Метода k-ближайших соседей (k-NN) - 84.54%.

Среди трех методов наибольшая точность распознавания была достигнута для сверточных нейронных сетей (CNN) - 94.49%, хотя для обучения машины данным методом расходуется немного больше ресурсов. Как показал эксперимент, данный метод показал себя с хорошей стороны для фигур, имеющих большое количество деталей, например, звезда.

На основе этого эксперимента, выяснилось, что точность распознавания даже простых геометрических фигур недостаточно высока и требует дальнейшего исследования методов для классификаторов машинного обучения.

#### Список использованной литературы:

1. С. а. N. G. Kanellakis, «Survey on computer vision for UAVs: Current developments and trends,» Journal of Intelligent & Robotic Systems, т. 87, № 1, pp. 141--168, 2017.
2. L.-W. а. H. J.-W. а. F. K.-C. Tsai, «Vehicle detection using normalized color and edge map,» IEEE transactions on Image Processing, т. 16, № 3, pp. 850--864, 2007.
3. G. R. Bradski, «Computer vision face tracking for use in a perceptual user interface,» 1998.
4. S.-Z. а. L. H.-J. Wang, «Detection and recognition of license plate characters with different appearances,» Proceedings of the 2003 IEEE International Conference on Intelligent Transportation Systems, т. 2, pp. 979--984, 2003.
5. X.-X. а. S. C. Y. Niu, «A novel hybrid CNN--SVM classifier for recognizing handwritten

- digits,» Pattern Recognition, т. 45, № 4, pp. 1318--1325, 2012.
6. H. a. V. S. N. a. R. R. Moballegh, «Gradient vector gridding: An approach to shape-based object detection in robocup scenarios,» Robot Soccer World Cup, pp. 162--173, 2011.
  7. K. Agarwal, «Object detection in refrigerators using Tensorflow,» 2018.
  8. Y. a. Z. W. a. Y. J. a. L. W. a. H. L. Qian, «Comparing machine learning classifiers for object-based land cover classification using very high resolution imagery,» Remote Sensing, т. 7, № 1, pp. 153--168, 2015.
  9. Y. a. B. L. a. B. Y. a. H. P. LeCun, «Gradient-based learning applied to document recognition,» Proceedings of the IEEE, т. 86, № 11, pp. 2278--2324, 1998.
  10. P. Goldsborough, «A tour of tensorflow,» arXiv preprint arXiv:1610.01178, 2016.
  11. L. Wang, Support vector machines: theory and applications, т. 177, Springer Science & Business Media, 2005.
  12. O. Sutton, «Introduction to k nearest neighbour classification and condensed nearest neighbour data reduction,» University lectures, University of Leicester, pp. 1--10, 2012.
  13. T. a. F. S. a. Z. Y. a. W. P. a. Z. J. Liu, «Implementation of training convolutional neural networks,» arXiv preprint arXiv:1506.01195, 2015.
  14. «Four Shapes,» [В Интернетe]. Available: <https://www.kaggle.com/smeschke/four-shapes>. [Дата обращения: 02 03 2020].
  15. P. Umesh, «Image Processing in Python,» CSI Communications, т. 23, 2012.
  16. «SAP Help Portal,» [В Интернетe]. Available: <https://help.sap.com/viewer/41d1a6d4e7574e32b815f1cc87c00f42/3.2/ru-RU/9c144a376f004058b4e9fe56727359af.html>. [Дата обращения: 05 03 2020].

**Рецензент: т.и.к., доц. м.а. Исмаилова Р**

УДК 37

DOI 10.33514/1694-7851-2020-2-50-55

**Быковченко Н.С.**

И. Арабаев атындагы КМУнун педагогика кафедрасынын доценти

## **БОЛОЧОК МУГАЛИМДИ КЕСИБИНЕ ДАЯРДОДО ПЕДАГОГИКА КУРСУНУН МААНИСИ**

### **ЗНАЧЕНИЕ КУРСА ПЕДАГОГИКИ В ПРОФЕССИОНАЛЬНОЙ ПОДГОТОВКЕ БУДУЩЕГО УЧИТЕЛЯ**

### **THE IMPORTANCE OF THE PEDAGOGY COURSE IN THE PROFESSIONAL TRAINING OF THE FUTURE TEACHER**

**Аннотация:** Макалада педагогика жана педагогиканын тарыхы сыяктуу илим тармактарын окутуу курсунда педагогикалык даярдоонун мааниси жөнүндө сөз болот. Педагогиканын тарыхы курсунун болочок мугалимдердин педагогикалык илимди, дүйнөлүк жана ата мекендик билим берүүнүн аксиологиялык аспектерин таануда, дүйнө таанымын жана педагогикалык маданиятын калыптандыруудагы орду көрсөтүлгөн. Педагогиканын