An Image Classification Method Using Hashing Preprocessing

Sergei Ivanov, Tatiana Zudilova, Alexander Ruban, Anantchenko Igor, Lubov Ivanova
ITMO National Research University (ITMO University)
Saint Petersburg, Russian Federation
serg_ie@mail.ru, zudilova@ifmo.spb.ru, ruban_1998@hotmail.com, igor@anantchenko.ru, 45is@mail.ru,

***Abstract*—In the study, the authors consider classical classification methods for the applied problem of image recognition. Accuracy, computation time, classifier size, training time for the following methods are considered: "Fully Connected Neural Network", "Convolutional Neural Network", "Recurrent Neural Network", "Decision Tree", "Gradient Boosted Trees", "Logistic Regression", "Markov", "Naive Bayes", "Nearest Neighbors", "Random Forest", "Support Vector Machine". A new approach "Neural Network with Hash" is proposed, which represents image preprocessing using polynomial hashing. Collision resolution is performed by a fast method of open addressing. A computer experiment on classification by 10 classes was carried out on a dataset of 600 animal images using the Wolfram mathematical package. For the proposed approach with preprocessing, the results showed the same classification accuracy as the classical methods, and a higher training and computational speed than the "Convolutional Neural Network" and "Recurrent Neural Network".**

I. INTRODUCTION

Data classification is one of the most common tasks in machine learning that involves determining the class of an object based on a set of features. There are numerous methods of data classification that are utilized in various fields such as computer vision, medicine, automatic face recognition, robotics, and many more. Image and video processing are integral parts of the modern world in diverse areas such as medicine, science, engineering, advertising, and others.

For image classification, the following groups of methods exist:

- Feature-based methods: these methods are based on extracting features from images and subsequently classifying based on those features. Examples include histogram of oriented gradients (HOG), principal component analysis (PCA), and local binary patterns (LBP) methods.

- Neural network-based methods: these methods use neural networks to classify images. For instance, convolutional neural networks (CNN) are one of the most popular image classification methods used in computer vision.

- Machine learning algorithm-based methods: these methods utilize machine learning algorithms to classify images. Examples include support vector machines (SVM), naive Bayes classifier, and decision trees.

- Deep learning-based methods: these methods use deep neural networks to classify images. Examples include deep convolutional neural networks (DCNN) and recurrent neural networks (RNN).

Existing image classification methods have their limitations, such as high computational complexity, the need for a large amount of data for training, instability under various conditions, and others. Therefore, developing a new image classification method utilizing hashing could be a relevant task.

The utilization of hashing in image classification has the potential to decrease the amount of data required for training and expedite the classification process. Additionally, hashing may exhibit robustness in various circumstances, such as changes in image size and distortions.

The objective of this study is to develop a methodology based on image preprocessing using hashing to enhance the speed of neural network training and computations.

In the second section of the article, various methods of data classification are examined. The third section of the article investigates an approach to preprocessing utilizing polynomial hashing of the matrix. The fourth section presents the outcomes of the computational experiment.

II. REVIEW OF CLASSIFICATION METHODS

Data classification methods involve the use of features, neural networks, machine learning algorithms, and deep learning. However, these methods have their drawbacks, such as computational complexity and instability under different conditions. The application of hashing for image classification may reduce the amount of data required for training and speed up the classification process. Furthermore, hashing can be robust to different conditions, such as changes in image size and distortion.

Recently, a number of scientific articles have been published on the application of deep learning to analyze large volumes of data. One such article is "A survey on deep learning for big data" [1], where the authors describe the current state and future prospects of deep learning methods for analyzing big data. The article discusses the limitations of traditional machine learning methods when working with large data sets, and provides an overview of key concepts and architectures of deep learning, including convolutional neural networks, recurrent neural networks, and deep neural networks with multiple layers.

The authors of the article then examine various applications of deep learning for working with big data, such as image analysis, natural language processing, social network and medical data analysis. The article also provides examples of the use of deep learning for analyzing time series and large data sets.

Another interesting article is “Deep learning for classification of hyperspectral data: A comprehensive review” [2], where the authors explore the use of deep learning for classifying hyperspectral data. The article describes various architectures of deep neural networks, such as convolutional neural networks, recurrent neural networks, and autoencoders, as well as discusses issues related to processing large volumes of hyperspectral data.

“Data augmentation for improving deep learning in image classification problem” [3] is another article that addresses the problem of insufficient training data or uneven class balance in data sets, using the example of images of various types of affine transformations. The authors also analyze several data augmentation methods in image classification tasks, using transformations such as rotation, cropping, scaling, and histogram-based methods.

Another noteworthy article is “Deep learning for time series classification: A review” [4], where the authors investigate the use of deep learning in the task of time series classification. The article describes various architectures of deep neural networks, including convolutional and recurrent neural networks, as well as hybrid models. The authors also discuss data preprocessing methods, such as normalization and data augmentation. In addition, the authors analyze the use of deep learning for classifying various types of time series data, such as financial and medical data, among others.

A. Fully Connected Neural Network

A Fully Connected Neural Network (FCNN) is a type of neural network where all neurons in one layer are connected to all neurons in the next layer. These networks are also known as multilayer perceptrons and are widely used in various fields such as computer vision, natural language processing, and speech recognition [5].

The FCNN consists of several layers, each of which consists of neurons. The input data passes through the first layer, where each neuron receives a portion of the information. Then the data is passed through several hidden layers where neurons use internal weights and activation functions to calculate new values. Finally, the data passes through the last layer, which determines the output values, such as classification or regression.

One of the main advantages of FCNN is the ability to automatically extract features from input data without manually defining characteristics. Additionally, FCNN can be trained on large volumes of data and achieve high accuracy in classification and regression tasks.

There are many variations of FCNN, including deep neural networks and convolutional neural networks. Deep neural networks have more than one hidden layer, allowing the model to learn more complex dependencies in data. Convolutional neural networks are specifically designed for image analysis and use convolutional layers to extract features from images.

Various optimization methods are used to train FCNN, such as gradient descent and stochastic gradient descent. There are also various loss functions and metrics used to evaluate the accuracy of the model.

One of the main applications of FCNN is image and video processing, including object recognition, image classification, and motion analysis. They are also used in medical diagnosis, financial forecasting, and other fields.

B. Convolutional Neural Network

Convolutional Neural Network (CNN) is one of the most common approaches to image classification. This model uses convolutional layers to extract features from images and subsequent fully connected layers for classification. CNNs have been applied in many areas, including face recognition, object recognition in images, and medical applications [6].

In their comprehensive review of deep learning techniques for computer vision, including CNNs, Athanasios Voulodimos et al. [1] highlight the inspiration for CNNs from the structure of the visual system and the acquisition of translational invariance by neurons with the same parameters applied to patches of the previous layer at different locations.

CNNs consist of three main types of neural layers: convolutional layers, pooling layers, and fully connected layers. Convolutional layers use different kernels to convolve the entire image as well as intermediate feature maps, creating various feature maps. Pooling layers reduce the spatial dimensions (width and height) of the input volume for the next convolutional layer. After several convolutional and pooling layers, high-level reasoning in the neural network is performed using fully connected layers. Neurons in the fully connected layer are fully connected to all activations in the previous layer, as their name suggests. Therefore, their activation can be computed using matrix multiplication followed by bias addition. Fully connected layers ultimately transform two-dimensional feature maps into a one-dimensional feature vector, which can either be classified into specific categories or treated as a feature vector for further processing.

CNNs have been extremely successful in computer vision applications such as face recognition, object detection, and powering vision in robotics and self-driving cars. The use of CNNs for NLP is a rapidly growing area of research, and their potential applications in this field are still being explored.

C. Recurrent neural networks

Recurrent neural networks (RNNs) are a powerful tool in the field of sequential data analysis, such as text, audio recordings, and time series [1]. They differ from ordinary neural networks in that they have internal states, which allow them to process sequences of arbitrary length. Moreover, RNNs are able to take contextual information into account when analyzing sequences, making them particularly useful for tasks such as classification, text generation, machine translation, and time series analysis.

The main advantage of RNNs is their ability to process sequences of arbitrary length and take contextual information into account when analyzing data. This allows RNNs to achieve high accuracy in classification and text generation tasks, as well as being effective in processing time series. In addition, RNNs can be trained using backpropagation methods, which allows for the use of powerful optimization methods, such as gradient descent, for tuning model parameters.

One of the most popular variants of RNNs is the long short-term memory (LSTM) model [7]. LSTM uses special memory blocks that allow models to learn on long time intervals and avoid the problem of vanishing and exploding gradients. This model has found application in many tasks, including text processing, music generation, and time series forecasting.

D. Decision Tree

One of the most prominent variations of recurrent neural networks (RNNs) is the long short-term memory (LSTM) model, as highlighted in [8]. The LSTM architecture utilizes specialized memory blocks to facilitate learning on long-term sequences, thereby circumventing the issues of exploding and vanishing gradients. Its versatility has led to its wide-ranging applications in several domains, including text processing, music composition, and time-series forecasting.

The fundamental concept behind a decision tree algorithm is to determine the most suitable condition for partitioning data into two or more subsets at each iteration. The construction of a decision tree begins with a root node that encompasses all available data. Subsequently, the data is partitioned into two or more subsets based on the values of one of the features. Each subset is then further divided into sub-subsets, and this process continues until the stopping criterion is met.

Decision trees are commonly employed in both classification and regression tasks. In the case of classification, the decision tree algorithm splits the dataset into several categories based on feature values, whereas in regression, it predicts the target variable value based on the input feature values.

One of the most significant advantages of decision trees is their interpretability, which implies that it is straightforward to comprehend the algorithm's decision-making process based on the specified conditions. Additionally, decision trees are highly versatile and can handle any type of data, including numerical and categorical.

E. Gradient Boosted Trees

Gradient Boosted Trees (GBT) is a machine learning method that allows for the creation of ensembles of decision trees and improves prediction accuracy by sequentially adding trees. GBT is used for classification and regression tasks.

The main idea behind GBT [9] is to create an ensemble of trees, where each tree is trained on the residuals of the previous tree (Fig. 1) [21]. At each iteration, a new tree is added that attempts to reduce the error on the training set. This approach enables GBT to achieve high accuracy and robustness in classification and regression tasks.

There are various GBT variants, such as Gradient Boosting Machine (GBM), XGBoost, and LightGBM [10]. They differ in the way missing values, outliers, and regularization are handled. Additionally, GBT can be extended for ranking and multi-class classification tasks.

To improve the quality of the GBT model, various methods can be used, such as tuning the model's parameters, selecting an optimal loss function, and optimizing the learning rate. Additionally, to reduce training time and increase efficiency, distributed computing and parallel training on graphical processing units can be employed.

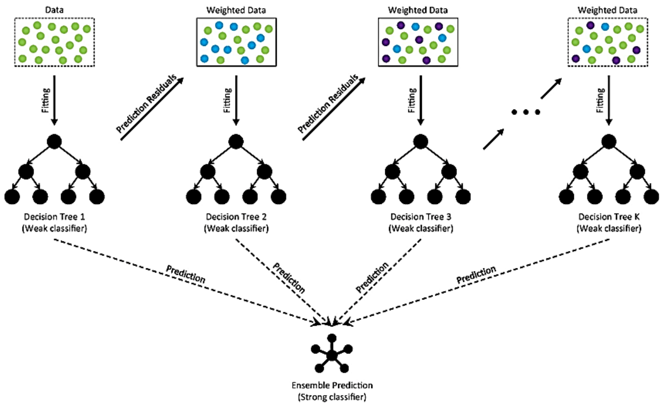

Fig. 1. GBT model

GBT has several advantages, such as high prediction accuracy, robustness to noise and outliers, and the ability to handle heterogeneous data. However, to achieve the best model quality, it is necessary to carefully tune the parameters and select an optimal loss function. Additionally, GBT may suffer from overfitting, which can be avoided through regularization.

F. Logistic Regression

Logistic Regression (LR) is a statistical method used to analyze the relationship between a binary dependent variable and one or more independent variables. It finds wide application in binary classification tasks where it is necessary to determine the membership of an object to one of two classes.

The main idea of logistic regression is to build a linear model that transforms the linear combination of input features into the probability of assigning an object to one of the classes [11]. This is achieved using the logistic function (sigmoid), which restricts the output value between 0 and 1. Logistic regression tunes the coefficients of the linear model on the training set using maximum likelihood estimation.

The objective of training a logistic regression model is to determine the optimal parameter values of the model that minimize classification errors. To accomplish this, a loss function is utilized, which quantifies the difference between the predicted and actual outputs

G. Markov

The Hidden Markov Model (HMM) is a statistical model used to model sequences, where each element is an observation or event generated by a hidden Markov chain [12].

A Markov chain is a model of a random process in which the current state depends only on the previous state, and the probability of transition depends only on the current state. In HMM, a finite number of hidden states are used, between which transitions occur according to probability distributions. Observational data, also known as symbols, are only observed when the system is in a certain hidden state, and each observation is associated with only one state.

HMM can be used for various tasks such as speech recognition, handwriting recognition, biological sequence analysis, and many others. For this purpose, the model is trained on training data to determine the parameters of probability distributions describing transitions between hidden states and observation generation. After training, the model can be used for classification of new sequences and generation of new sequences.

H. Naive Bayes

Naive Bayes is a family of probabilistic classification algorithms that employ Bayes' theorem (1) and is utilized for determining the class of a new object based on its features. The algorithm treats the features as independent variables, making it effective when dealing with a large number of features [13]. Naive Bayes has been widely used in various fields such as natural language processing, image recognition, and sentiment analysis.

$$P\left(y_j|X\right)=\frac{P\left(y_j\right)P(X\mid y_j)}{P\left(X\right)}=\frac{P\left(y_j\right)\prod_{i=1}^{n}P(x_i\mid y_j)}{\prod_{i=1}^{n}P\left(x_i\right)} \tag{1}$$

There are various versions of Naive Bayes, depending on the type of probability distribution used to describe the classes. For example, Gaussian Naive Bayes assumes that the features follow a normal distribution, while Multinomial Naive Bayes assumes that the features follow a multinomial distribution. In addition to textual document classification [8], Naive Bayes can also be used in spam filtering, medical diagnosis, and fraud detection. With the rise of big data and machine learning, Naive Bayes is becoming increasingly important in the field of data science.

I. Nearest Neighbors

Nearest Neighbors (NN) is a machine learning method used for data classification and regression. It is based on the principle of nearest neighbors: for a new observation, a certain number of training examples closest to it are selected, and their labels are used to predict the label of the new observation.

The main idea of the NN method is to determine the distance between each pair of observations in the feature space. Then, for each new observation, the k nearest neighbors are found from the training set, and their labels are used to predict the label of the new observation [14].

Different distance metrics can be used in NN, such as Euclidean distance, Manhattan distance, cosine distance, etc. Various methods can also be used to determine k, which is the number of nearest neighbors to be used for predicting the label of the new observation.

The NN method can be used to solve classification and regression problems. In the classification problem, NN selects the class to which the majority of k nearest neighbors belong. In the regression problem, NN predicts a numerical value for the new observation, using the mean or median of the k nearest neighbors.

J. Random Forest

Random Forest is a machine learning algorithm that is used for solving classification and regression tasks. It is based on building an ensemble of decision trees, where each tree is trained on a random subset of data and a random subset of features [8, 15]. Then, for each new object, the algorithm predicts its class based on the predictions of all trees in the ensemble.

The main advantage of Random Forest is its ability to handle a large number of features and to detect and utilize non-linear dependencies between features. In addition, Random Forest has good resilience to overfitting and noise in the data.

There are several variations of the Random Forest algorithm, such as Extra Trees and Extremely Randomized Trees, which are variations of the algorithm that use different approaches for selecting random subsets of data and features. These variants can improve the quality of predictions in some cases, especially when there is little data.

Random Forest can be effectively applied for image analysis, including image classification and segmentation. For example, in medical applications, Random Forest can be used for automatic image segmentation to highlight regions of interest for further analysis. The algorithm can also be used for object classification in images, such as face recognition and object recognition in photographs.

K. Support Vector Machine

Support Vector Machine (SVM) is a classification method that employs hyperplanes to separate data into two classes. The primary idea is to find a hyperplane that divides objects of different classes while maximizing the distance between it and the closest points of both classes. This approach allows SVM to achieve high accuracy and stability in classification and regression tasks, even in the presence of noise and overlapping classes [8].

There are various versions of SVM, including linear SVM and nonlinear SVM with kernel functions. SVM can also be extended for multiclass classification and regression tasks. Moreover, recent studies have shown that SVM can be effectively applied to handle large volumes of data, including images and texts.

One of the popular applications of SVM is image classification, where it is necessary to automatically determine which class an image belongs to. This method can be used with both linear and nonlinear kernel functions. For example, SVM using the RBF kernel can be effectively applied to classify images with a nonlinear boundary between classes. Additionally, to improve the quality of classification, various methods of data preprocessing can be used, such as feature extraction, image segmentation, and dimensionality reduction.

III. HASHING-BASED PREPROCESSING

Hashing is a critical aspect of modern computing and data security. It plays a vital role in many areas, including cryptography, digital signatures, and authentication. The technique is used to convert arbitrary data into a fixed size hash, which can then be used to identify the original data or to verify its integrity.

Hashing algorithms also play a crucial role in ensuring data integrity during transmission. By hashing the data before sending it, the sender can ensure that the recipient receives an unaltered copy of the data. If the recipient receives a different hash value, they can conclude that the data has been tampered with during transmission.

In contemporary intelligent applications across various domains, numerous hashing techniques are employed to ensure data integrity and mitigate the risk of data corruption. Some of the widely used hashing techniques include Adler 32-bit cyclic redundancy check, 32-bit cyclic redundancy check, 128-bit MD2 code, 128-bit MD4 code, 128-bit MD5 code, 160-bit RIPEMD code, RIPEMD-160 following SHA-256 (as used in Bitcoin), 160-bit SHA-1 code, 256-bit SHA code, Double SHA-256 code (as used in Bitcoin), 384-bit SHA code, 512-bit SHA code, and hash codes based on polynomial hashing of matrices.

Using hashing techniques helps to ensure the security of data and to prevent unauthorized access. By implementing these techniques, organizations can effectively protect their sensitive information and reduce the risk of data breaches.

An interesting approach to using hashing for retrieving similar objects in large datasets is presented in the article "Locality-Sensitive Hashing Scheme Based on Heap Sort of Hash Bucket" [16]. The authors describe an algorithm based on LSH that allows for fast and efficient retrieval of similar objects in large datasets, such as images and audio files. The proposed method relies on a combination of heap sort and hash bucketing to achieve high computational efficiency and accuracy in identifying nearest neighbors.

The article "Deep Multi-View Enhancement Hashing for Image Retrieval" [17] proposes a novel method for image retrieval using deep multi-view enhancement hashing. The authors argue that traditional methods for image retrieval based on hashing often struggle to achieve high accuracy due to the difficulty of effectively capturing the complex features of images.

To address this challenge, the authors propose a deep multi-view enhancement hashing method that utilizes multiple views of images to better capture their features. Specifically, the method uses a deep neural network to extract features from multiple views of each image and then combines these features into a single enhanced view. The enhanced view is then used to generate hash codes for each image, which can be used for efficient retrieval.

The authors evaluate their method on several benchmark datasets and demonstrate that it outperforms existing hashing-based methods in terms of accuracy and efficiency. They also conduct a comprehensive analysis of the various components of their method to show the effectiveness of each component in improving the overall performance.

The article "Scalable Deep Hashing for Large-Scale Social Image Retrieval" [18] addresses the challenge of large-scale image retrieval in social media networks and proposes a solution based on deep learning. The authors note that hashing is an effective method for image retrieval in large-scale collections due to its high computational efficiency and low storage cost.

In the article "Deep Transfer Hashing for Image Retrieval" [19], the authors present a novel deep learning method for efficient image retrieval. The work is based on the assumption that conventional image labels contain only a small amount of information necessary to determine image similarity.

Instead, the authors propose using knowledge from a more complex deep learning model and transferring this knowledge to a simpler model to solve the task of fast image retrieval. To achieve this, they employ knowledge distillation for model compression and deep hashing for fast image retrieval.

They minimize the distribution of hashed codes between the teacher and student hashing layers to improve search performance. Through experiments on CIFAR-10 and NUS-WIDE datasets, the authors demonstrate that the proposed method significantly improves the baseline retrieval results compared to other state-of-the-art methods.

This section discusses a methodology that involves preprocessing with the use of polynomial hashing.

Similarly, in [20], authors enhance the Neural ODE method by incorporating a special numerical method scheme. The developed scheme offers high computational accuracy and optimal complexity. Experimental results demonstrate acceptable classification accuracy when using neural differential equations with the embedded numerical method scheme.

At the first stage of this methodology, the hash code for a normalized black and white image of an animal with a size of 100x100 pixels is calculated.

At the first stage, we compute the hash code for a normalized black and white image of an animal with a size of 100x100 pixels using polynomial hashing of the matrix F based on the formula:

$$Hash\left(F_{x1\le i\le x2,\, y1\le i\le y2}\right)=k_1^{x1}k_2^{y1}\sum_{i=x1}^{x2}\sum_{j=x2}^{x2}F_{ij}k_1^{i}k_2^{j} \qquad (2)$$

This choice was determined by the optimal complexity and speed of hash computation. To resolve collisions, a fast open addressing method is applied.

At the next stage, we apply Fully-connected neural networks, which consist of a set of fully connected layers. Fully-connected neural networks are deep learning networks that are independent of the input data structure and have a universal architecture with "universal approximators" capable of learning any function. Fully-connected neural networks can be directly stacked.

The practical application of Fully-connected neural networks for the classification task on the considered data sets has demonstrated high performance and acceptable recognition accuracy.

IV. Results of the computational experiment

This section provides a detailed description of the conducted computational experiment, which involved training and recognition of a dataset of 600 animal images using the Wolfram mathematical package. After obtaining hash codes for the images, we applied fully-connected neural networks.

Fig. 2 presents the results of the training time.

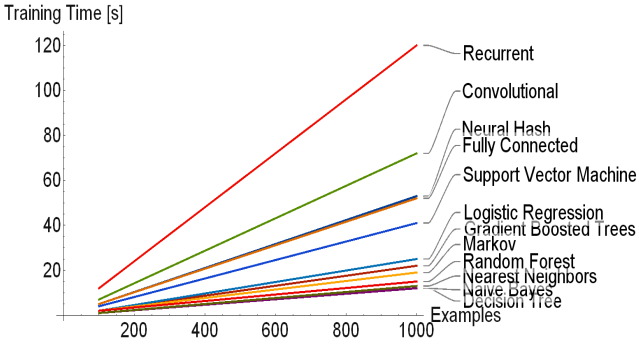

Fig. 2. The training time

Fig. 2 shows the shortest training time for the presented method with preprocessing and the longest time for the recurrent neural network. The presented method with hashing is 10% faster in training time than FullyConnected, 50% faster than Convolutional, and 90% faster than Recurrent neural network.

Fig. 3 presents the results of the computation time.

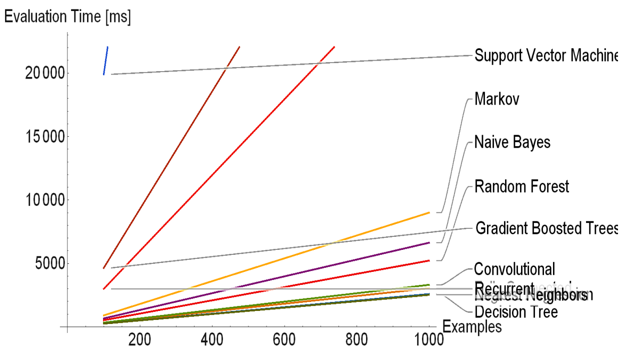

Fig. 3. The evaluation time

Fig. 3 shows the shortest computation time for the presented method with preprocessing and the longest time for the algorithms "Markov", "Naive Bayes", "Random Forest", "Support Vector Machine". The presented method with hashing is 30% faster in computation time than FullyConnected, 40% faster than Convolutional, and 90% faster than Recurrent neural network.

Fig. 4 presents the size of the classifier.

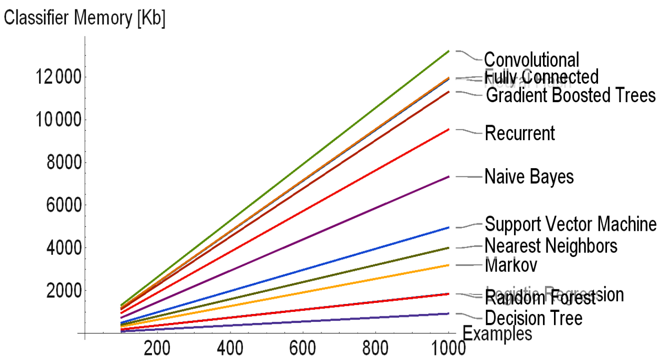

Fig. 4. The classifier memory

Fig. 4 shows the largest size of the classifier for the "Convolutional Neural Network"

Fig. 5 presents the results of the classification accuracy.

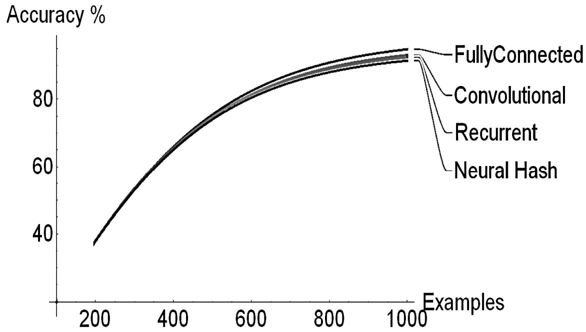

Fig. 5. The accuracy

Fig. 5 shows that the presented method with preprocessing achieves an accuracy of over 92% in classification.

The computations and verification were performed using the Wolfram mathematical package.

The results of the conducted computer experiment using mathematical software show the applicability of the method with pre-hashing for solving classification tasks.

IV. Conclusion

The results of the study examine classical classification methods for the applied task of image recognition. Accuracy, computation time, classifier size, and training time are considered for the following methods: "Fully Connected Neural Network", "Convolutional Neural Network", "Recurrent Neural Network", "Decision Tree", "Gradient Boosted Trees", "Logistic Regression", "Markov", "Naive Bayes", "Nearest Neighbors", "Random Forest", "Support Vector Machine". A new approach, "Neural Network with Hash", is proposed that involves preprocessing images using polynomial hashing. Collision resolution is performed using a fast open addressing method. A computer experiment for classification of 10 classes was conducted on a dataset of 600 animal images using the Wolfram mathematical package. For the proposed approach with preprocessing, the results showed classification accuracy equal to classical methods and higher than "Convolutional Neural Network" and "Recurrent Neural Network" in terms of speed of learning and computation.

References

[1] Zhang, Qingchen, “A survey on deep learning for big data.” *Information Fusion* 42, 2018, pp. 146-157.

[2] Audebert, “Deep learning for classification of hyperspectral data: A comparative review.” *IEEE geoscience and remote sensing magazine* 7, no. 2, 2019, pp. 159-173.

[3] Mikołajczyk, “Data augmentation for improving deep learning in image classification problem.” *International interdisciplinary PhD workshop, May. 2018,* pp. 117-122.

[4] Ismail Fawaz, “Deep learning for time series classification: a review.” *Data mining and knowledge discovery* 2019, pp. 917-963.

[5] Zhang, "On the learnability of fully-connected neural networks." *Artificial Intelligence and Statistics*, 2017, pp. 83-91.

[6] Voulodimos, "Deep learning for computer vision: A brief review." *Computational intelligence and neuroscience,* 2018.

[7] Yu Yong, "A review of recurrent neural networks: LSTM cells and network architectures." *Neural computation* 31, no. 7, 2019, pp. 1235-1270.

[8] Li Ruiguang, "A Review of Machine Learning Algorithms for Text Classification." *Cyber Security,* 2022, p. 226.

[9] Huan Zhang, "Gradient boosted decision trees for high dimensional sparse output." *International conference on machine learning*, 2017, pp. 3182-3190.

[10] Al Daoud, "Comparison between XGBoost, LightGBM and CatBoost using a home credit dataset." *International Journal of Computer and Information Engineering* 13, no. 1, 2019, pp. 6-10.

[11] LaValley, "Logistic regression." *Circulation* 117, no. 18, 2008, pp. 2395-2399.

[12] Dymarski *Hidden Markov models: Theory and applications*. BoD–Books on Demand, 2011.

[13] Berrar, "Bayes' theorem and naive Bayes classifier." *Encyclopedia of Bioinformatics and Computational Biology: ABC of Bioinformatics* 403, 2018, p. 412.

[14] Taunk Kashvi, "A brief review of nearest neighbor algorithm for learning and classification.' *2019 International Conference on Intelligent Computing and Control Systems (ICCS)*, 2019, pp. 1255-1260.

[15] Fawagreh, "Random forests: from early developments to recent advancements." *Systems Science & Control Engineering: An Open Access Journal* 2, no. 1, 2014, pp. 602-609.

[16] Fang Bo, "Locality-Sensitive Hashing Scheme Based on Heap Sort of Hash Bucket." *14th International Conference on Computer Science & Education (ICCSE)*, 2019, pp. 5-10.

[17] Yan Chenggang, "Deep multi-view enhancement hashing for image retrieval." *IEEE Transactions on Pattern Analysis and Machine Intelligence* 43, no. 4, 2020, pp. 1445-1451.

[18] Cui Hui, "Scalable deep hashing for large-scale social image retrieval." *IEEE Transactions on image processing* 29, 2019, pp. 1271-1284.

[19] Zhai Hongjia, "Deep transfer hashing for image retrieval." *IEEE Transactions on circuits and Systems for Video Technology* 31, no. 2, 2020, pp. 742-753.

[20] A. Televnoy, S. Ivanov, T. Zudilova and T. Voitiuk, "Neural ODE Machine Learning Method with Embedded Numerical Method*", 28th Conference of Open Innovations Association (FRUCT)*, 2021, pp. 451-457

[21] Deng, Haowen, Youyou Zhou, Lin Wang, and Cheng Zhang. "Ensemble learning for the early prediction of neonatal jaundice with genetic features." *BMC medical informatics and decision making* 21, 2021, pp 1-11.