Interdisciplinary Journal of Information, Knowledge, and Management

An Official Publication of the Informing Science Institute InformingScience.org

IJIKM.org

Volume 20, 2025

AUTOMATING DIGITIZED DOCUMENT PROCESSING WITH HANDWRITTEN DIGITS IN THE PUBLIC SECTOR USING CONVOLUTIONAL NEURAL NETWORKS

Nadiia Boliubash*	Petro Mohyla Black Sea National University, Mykolaiv, Ukraine	Nadiya.Bolubash@chmnu.edu.ua
Oleksandr Yevtushenko	Petro Mohyla Black Sea National University, Mykolaiv, Ukraine	alnievt@ukr.net

* Corresponding author

ABSTRACT

Aim	Automation of information extraction from digitized complex documents that contain both printed and handwritten text (as well as non-textual information) is one of the actual problems in the digital transformation of public administration. This study proposes an ML-based approach to improve the quality of automated extraction and processing of numerical data from digitized documents with handwritten digits using optical character recognition technology.
Background	Currently, in public institutions in Ukraine, manual intervention is a bottleneck in the process of extracting numerical data from digitized documents and subsequent processing. New approaches to automating these processes are needed.
Methodology	The methodology includes preprocessing of the document image, segmentation and classification of handwritten digits, conversion of the extracted digits into a date format with the possibility of their validation, and performing necessary calculations based on the extracted numerical information. First, handwritten digits from scanned images of document pages are segmented, then they are preprocessed and sent for recognition with a module based on convolutional neural networks. The image preprocessing steps consisted of binarization, application of a Gaussian filter to remove noise, and use of the Hough transform to correct the document's skew angle. A CNN model was used to perform character-by-character classification for the recognition of segmented digits.
Contribution	The study addresses current limitations in the extraction of handwritten digits from complex document images. The segmentation technique utilizes morphological transformations such as erosion and dilation, as well as the connected

Accepting Editor Ahmad Samed Al-Adwan │ Received: February 14, 2025 │ Revised: April 19, May 5, May 20, 2025 │ Accepted: May 21, 2025.

Cite as: Boliubash, N., & Yevtushenko, O. (2025). Automating digitized document processing with handwritten digits in the public sector using convolutional neural networks. *Interdisciplinary Journal of Information, Knowledge, and Management, 20,* Article 19. https://doi.org/10.28945/5527

components method. The study explored different architectures with convolutional neural networks to determine the optimal hyperparameter configuration. The research findings confirm the importance of integrating methods to ensure effective image preprocessing, segmentation, and recognition of extracted digits.

Findings: Experimental results demonstrate that the proposed approach decreases the processing time by a factor of 7.7 and increases the accuracy of numerical data recognition on pages containing fragments of handwritten digits. This indicates that operational tasks are completed with higher accuracy and efficiency.

Recommendations for Practitioners: Software may be developed based on this research and implemented in the Pension Fund of Ukraine to automate the processing of digitized documents to determine service lengths and calculate the amount of pension to be awarded. The solution makes it possible to eliminate manual intervention in the process of data extraction and processing.

Recommendations for Researchers: The study showed high accuracy in recognizing individual handwritten digits – 99.68% with the training data and 99.55% with the test data. However, the accuracy of recognition is lower if complex documents contain both handwritten and printed text as well as non-textual information. This is due to the complexity of segmenting handwritten characters, which requires further research to identify more effective methods for preprocessing and segmentation.

Impact on Society: The proposed methodology allows a significant reduction of human factors in the process of extracting data from digitized documents, accelerating their processing and increasing the efficiency of government institutions.

Future Research: Further research may further explore the areas of automated processing of digitized documents, especially in other branches of the state administration, taking into account the specifics of data extracted for further processing.

Keywords: digital transformation, public administration, document processing automation, optical character recognition (OCR), handwritten text recognition, handwritten digit recognition (HDR), convolutional neural network (CNN)

INTRODUCTION

Due to the rapid growth, spread, and necessity of information across all spheres of modern society, the electronic format is the most convenient mechanism for storing and utilizing information. The relevance of transferring information from paper to electronic media in the public sector is growing through many printed and handwritten documents requiring digitalization (Adnan & Akbar, 2019a; Yevtushenko, 2024). Digitized documents can be easily stored, accessed, shared, and analyzed. Further use of such papers often necessitates information recognition to extract the required data. Most workflows for obtaining and using information from digitized documents are slow because they require manual data extraction, input, and processing. The development of software that automates stages of these processes can significantly reduce the time needed for document processing in public administration while also enhancing the quality and efficiency of their work.

Stored digitized documents are usually images of scanned pages from paper documents. These are processed using OCR technology, which works well with simple documents. However, documents circulating in the public sector often have a complex structure, incorporating printed and handwritten texts, tables, diagrams, and non-textual information (Butenko et al., 2023). While existing software tools and cloud services on the software market are good at recognizing images of printed text, the accurate recognition of handwritten text remains an unresolved challenge. This task is difficult because people often merge characters when writing by hand, and writing styles vary significantly in shape, size, slant, texture, and background. Non-textual elements in documents – such as forms, signatures, and seals – further complicate text recognition because they are difficult to interpret. It necessitates software focused on specific subject areas that would cope with extracting information from complex documents containing handwritten text.

Scientists are conducting numerous studies to solve the problem of automating recognition of texts containing handwritten letters, digits, and other symbols, using a variety of approaches: template matching and correlation techniques, feature-based recognition, neural networks, ensemble methods, k-nearest neighbors' algorithm, Support Vector Machine, Random Forest, Naive Bayes classifier (Ali et al., 2019; Sueiras et al., 2018; Tabik et al., 2020). Template methods compare each symbol to existing templates. Feature algorithms consider the image as a vector of features, and recognition consists of comparing it to a set of reference vectors of the same dimension. In the studies by Alhamad et al. (2024) and Saqib et al. (2022), Convolutional Neural Networks (CNN) models were shown to be an effective tool for recognizing images of handwritten characters. Popular convolutional neural network models include LeNet, AlexNet, ZfNet, VGG, and GoogleLeNet (Zhao et al., 2024). They have various architectures that transform input data into output, hierarchically extracting and aggregating features and increasing the level of data abstraction in the direction from the inputs to the network's outputs. The goal of research scientists is the development of proprietary CNN architectures and selecting parameters that provide better performance and accuracy when solving image recognition problems of documents with printed and handwritten text.

Convolutional neural networks (CNNs) achieve high accuracy – often exceeding 99% – in recognizing isolated images of handwritten digits. However, their performance drops when applied to complex document images containing handwritten text with digits (Ahlawat et al., 2020; Liu et al., 2016; Saqib et al., 2022). The challenge arises from severe technical difficulties in segmenting handwriting, especially when symbols are fused. Even minor modifications to the image during preprocessing or segmentation can result in incorrect recognition decisions. Therefore, high-quality preprocessing and digitized document segmentation are crucial for accurately recognizing numerical data. Segmentation is important because most modern recognition systems use classifiers of individual characters rather than words or text fragments.

Despite significant progress in optical character recognition, obtaining high accuracy in recognizing handwritten numerical data within text documents of complex structures remains a critical problem (Raja et al., 2021). Solving specialized tasks that require high-precision recognition of digits in such documents requires improving the preprocessing and segmentation of digitized documents and using neural network models adapted for recognizing text with handwritten digits.

This article presents the results of research on automating the processing of digitized documents with handwritten digits in public administration bodies of Ukraine. This study proposes an approach for developing a comprehensive software application that recognizes handwritten digits and dates on images of workbook pages, which are currently processed manually in Pension Fund offices. The system uses the recognized dates to calculate a person's total length of service, which then determines the method used for pension calculation. The existing software for recognizing digitized documents has not been adapted to solve this problem. The current study was motivated by this gap and aims to fill this void by developing a software application that automates the described process. The paper describes the methods and algorithms used to solve this task. The developed methodology can be applied to automate the processing of digitized documents in other government agencies, thereby increasing the societal benefits of this research.

LITERATURE REVIEW

Automation of processing digitized documents increases the productivity of working with these documents and is a promising direction for numerous scientific research studies based on optical character recognition. Among the methods of information extraction, approaches that depend on statistical patterns or rules are predominant. For processing documents with a complex structure, which includes printed and handwritten text, the use of AI-based techniques and machine learning is promising (Adnan & Akbar, 2019b; Ali et al., 2019; Mahadevkar et al., 2024). An essential aspect of developing software for automating workflows related to extracting information from digitized documents is collaboration between organizations and researchers to solve problems associated with data analysis (Mahadevkar et al., 2024).

When extracting handwritten characters offline, the input images of scanned document pages require intensive and high-quality preprocessing. The preprocessing stage includes binarization, noise reduction, document skew correction, and text slant removal (Baviskar et al., 2021; Prum, 2017). Many researchers also focused on image resizing, blurring, and morphological operations at this stage, using the Open Source Library OpenCV (Eken et al., 2019; Xue et al., 2020; Q. Ye & Doermann, 2015). Binarization converts an image into binary black and white to facilitate further processing. OpenCV implements binarization methods: binarization Otsu, adaptive thresholding, and simple thresholding (Subudhi & Sahu, 2014). A low-pass filter is often applied to remove noise from scanned document images. Morphological operations help to remove unnecessary details or noise in the image and smooth the contours of objects. It is possible to remove noise such as salt and pepper and Gaussian noise using the OpenCV library (Boiangiu et al., 2020; Johnson et al., 2018). Among the methods for skew angle detection and correction of a document, which may occur during scanning, researchers are using projection profile analysis, Hough transforms, and morphological transforms (Boiangiu et al., 2020; Gari et al., 2017; Prum, 2017; Sahare & Dhok, 2018; Sakila & Vijayarani, 2017).

Segmentation is essential for accurate digit recognition, as it separates handwritten text fragments and the background on a document image (Kaur et al., 2015). Handwritten text segmentation involves sequential segmentation of lines, words, and characters. Character segmentation is an operation that decomposes an image into subimages of individual symbols. Popular methods of segmentation are X-Y-tree decomposition (Sahare & Dhok, 2018), connected component labeling (Laubrock & Dunst, 2020), Hough transforms (Singh et al., 2020), and histogram projection techniques (Mehul et al., 2014). The histogram method is used to segment text into lines and words, which involves constructing a histogram of the black pixels of the image. The algorithm assumes that the number of black pixels in the intervals between lines is significantly smaller than in lines of text. Word segmentation is performed similarly. Neural network techniques and the connected component labeling method are applied to segment words into characters.

Text segmentation is facilitated when extracting data from structured and semi-structured documents that are fully or partially in a specific format (containing tables, forms, or standardized templates). Using the OpenCV library, tables with predefined dimensions can be extracted from a digitized document as separate images. These images are then used to recognize handwritten text characters in the table cells. However, researchers have found that complex document layouts, such as tables, forms, or documents with mixed printed and handwritten text, create challenges for optical character recognition models (Liu et al., 2016). These challenges make it difficult to accurately extract numerical and textual data.

After segmentation, smoothing and normalization are applied to individual symbols to improve further analysis (Sahare & Dhok, 2018). This makes it easier to detect features when recognizing characters. Smoothing includes filling in small gaps and omissions in binary symbol images, as well as thinning to reduce image parameters. Normalization involves scaling the symbol image to the required dimensions and can be performed using the OpenCV library.

The following steps in recognition are feature extraction and classification. In OCR, the methods widely used for feature extraction are statistical and structural (Clausner et al., 2020; Sahare & Dhok, 2018; Y. Ye et al., 2018). Statistical methods identify the statistical features of a symbol. Structural methods identify the structural features of symbols, such as horizontal and vertical lines, endpoints, intersections between lines, and strokes. After feature extraction, classification is performed to identify and assign a symbol to a specific class. The most popular methods and algorithms used for classification in OCR research include k-nearest neighbors (Mehul et al., 2014), recurrent and convolutional neural networks (Y. Ye et al., 2018), Naive Bayes classifier (Liu et al., 2016), and Support Vector Machine (Kanya & Ravi, 2016).

Analyses of the existing research have shown that convolutional neural networks are the undisputed leader in image recognition when building a classifier model (Ahlawat et al., 2020; Alhamad et al., 2024; Ali et al., 2019). Numerous CNN models achieve high classification accuracy. This makes them the most suitable tool for recognizing images of individual symbols. Top-1 accuracy in handwritten digit recognition with modern CNNs, such as ResNet and EfficientNet, typically exceeds 99.5%. Top-5 accuracy often reaches 100% due to the small number of classes (only 10). Top-1 accuracy means the model's most probable prediction matches the correct answer. Top-5 accuracy indicates that the correct character is among the five most probable predictions. For more complex handwritten datasets, accuracy decreases slightly, especially for letters. However, well-optimized models still achieve 94-97% accuracy.

Convolutional Neural Networks have an architecture based on alternating convolutional and pooling layers. These are followed by one or more fully connected layers at the output. The alternation of layers allows the creation of new feature maps from previous ones, enabling the recognition of complex feature hierarchies (Ahlawat et al., 2020; Ali et al., 2019). The pooling layer works on each feature map to perform scaling. Common types of pooling are finding the maximum value and calculating the average value on the feature map. It should be noted that there may be several convolutional layers before a pooling layer. Changes in the architecture of modern CNN models have occurred in precisely this direction (Alzubaidi et al., 2021). While such changes increase network training time, more convolutional operations improve feature detection.

Fully connected layers perform high-level reasoning of complex patterns based on the features identified by convolutional layers. The last fully connected layer, the output layer, contains n neurons. Here, n is the number of categories used for classification. For handwritten digit recognition, n equals 10, corresponding to digits from 0 to 9 (Ahlawat et al., 2020).

A wide range of activation functions is available to train neural network models. The most commonly used activation functions are sigmoid, tanh, ReLU, and leaky ReLu (Alzubaidi et al., 2021). ReLU is used most frequently in convolutional layers. This function implements a simple threshold transition to zero. In the output layer, the Softmax activation function is typically used. It normalizes the network's output data to a probability distribution over recognized classes (Ali et al., 2019; Alzubaidi et al., 2021; Cui & Bai, 2019).

Researchers in the recognition field have established that convolutional neural network performance depends on hyperparameter choice (Cui & Bai, 2019). These hyperparameters are selected when designing the neural network architecture before training begins. Key hyperparameters include the activation function, number of epochs, number of hidden layers and their type, kernel size and stride size of the convolution, types of pooling operations, and the pooling window size. It is essential to determine the number of hidden layers and the number of feature maps in the convolutional layers. These parameters are chosen empirically by the researcher when building the classifier model. Some works have investigated the effect of the size of the convolution kernel on the model's accuracy. The results showed that the accuracy is inversely proportional to the size of the convolution kernel (Khanday et al., 2021). Developing a CNN classifier model requires research to identify the optimal configuration of hyperparameters. Poorly chosen hyperparameters can lead to low neural network performance and insufficient recognition accuracy (Ahlawat et al., 2020).

CNN classifier models train on datasets with many handwritten characters reflecting different writing styles. For handwritten digit recognition, the MNIST dataset is considered a benchmark. It is publicly available and includes a large number of handwritten digits for both training and testing the model (Ahlawat et al., 2020; Memon et al., 2020).

Research analysis showed that effective CNN model training depends on the proper configuration of the training process (Alzubaidi et al., 2021). This includes choosing the optimal loss function, optimizer, and regularization techniques. For multiclass classification, cross-entropy is often used as the loss function. It quantifies model error by measuring the difference between the predicted and actual results. The optimizer iteratively updates model parameters to minimize the loss function. Stochastic gradient descent (SGD) and its modifications, such as Adam, RMSprop, AdaGrad, and AdaDelta, are the most widely used (Saqib et al., 2022). Regularization helps prevent overfitting the classifier model. This is done by adding normalization and dropout layers to the CNN architecture. Dropout layers randomly exclude a fraction of neurons in a layer (Ahlawat et al., 2020; Ali et al., 2019; Memon et al., 2020). Model quality is evaluated using such metrics as accuracy, recall, precision, and F1-score (Reul et al., 2019).

Promising approaches for processing complex-structured documents include convolutional segmentation models. These models can identify and segment different regions within a document (Tüselmann & Fink, 2024). However, interpreting and classifying their content requires an intermediate step of OCR. Transformer-based models can enhance OCR by offering error correction and contextual recognition (Mahadevkar et al., 2024; Tüselmann & Fink, 2024). Hybrid deep learning AI-based techniques, which combine transformer models and CNNs, increase text recognition accuracy due to understanding the context (Aparna & Rajchandar, 2024). However, the use of convolutional segmentation models and transformer-based models in processing documents within public administration bodies faces limitations. Creating datasets for their training requires access to restricted sources containing confidential information. Meanwhile, pre-trained models are universal, but their performance on highly specialized tasks may be suboptimal.

Manual validation at the post-processing stage is crucial for ensuring the accuracy of handwritten text recognition in digitized documents in public administrations. This is especially important when handling complex or critical information (Boliubash, 2024; Reul et al., 2019). Post-processing includes checking the correctness of character recognition and correcting errors detected after image processing using OCR technology. Human intervention in verifying and correcting is critically essential for numerical data documents. Even minor mistakes in such data can lead to incorrect results during further processing. The software application designed to process the extracted information further should automatically receive recognized symbols from the CNN classifier. It must also support analytics for government needs through an interface where recognized data can be checked, corrected, or confirmed.

COMPUTATIONAL APPROACH

The methodology can be broken into two main parts, as described in Figure 1. This research aims to develop an application for automating the processes of extracting and further processing numerical information from images of digitized documents containing printed and handwritten text within table cells.

The first part of the methodology involves the extraction of handwritten digit images from scanned pages of digitized documents. The first step in this part is the preprocessing of the document page images. It includes binarization, noise removal, and document skew correction. The second step is to highlight the table markup on the page image and segment handwritten digits within table cells that contain numeric data. The third step involves classifying the segmented digits using a convolutional neural network. The selection of optimal hyperparameters of the neural network and its training were carried out on the MNIST dataset.

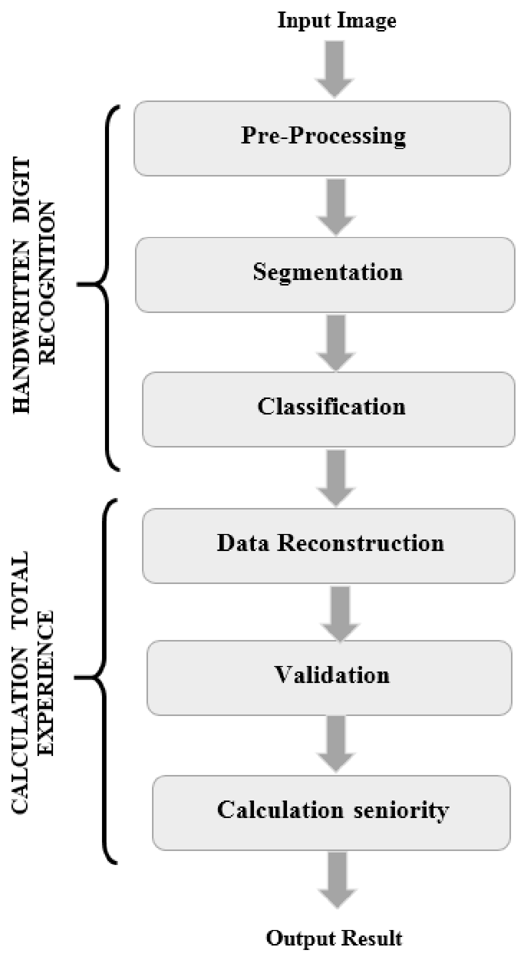


Figure 1. Methodology used to automate workflows for processing digitized documents with handwritten digits

The final stage of the research involved an experimental evaluation. The goal was to measure the processing speed of numerical data extracted from images of digitized documents containing handwritten digits using the described methodology.

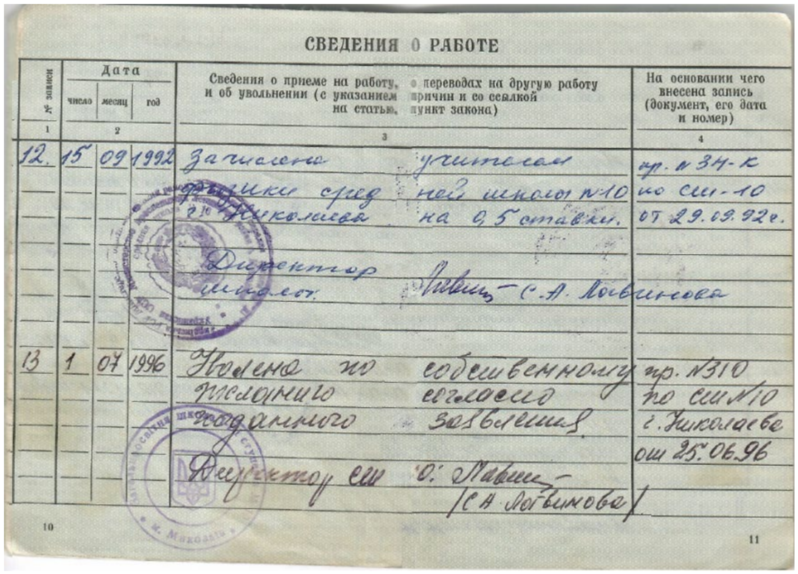
IMAGE PREPROCESSING

Image preprocessing is a critically important stage in optimizing workflows for extracting information from scanned documents. The main challenges at this stage are different types of distortions: geometric, background, and noise. Figure 2 shows an example of how the image of a scanned document page changes during preprocessing.

In the first step of preprocessing, the image is converted to grayscale by calculating the brightness of each pixel using the formula:

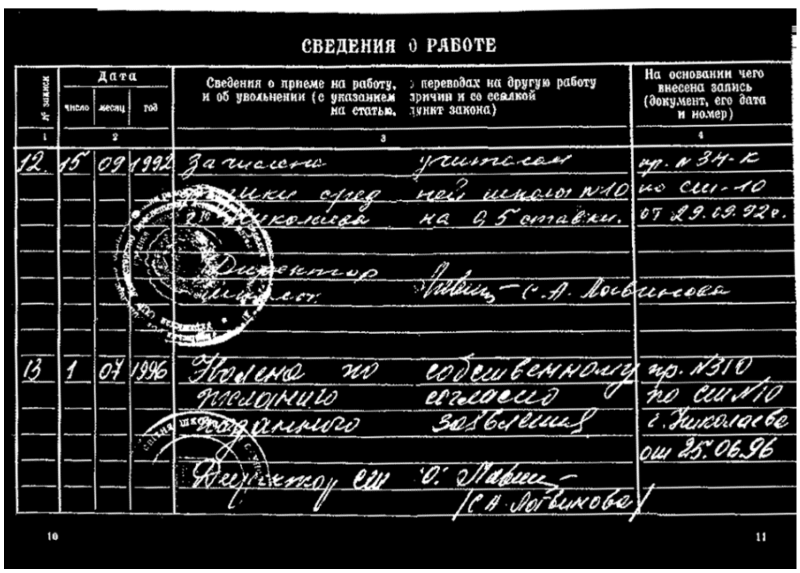
$$I = R \cdot 0.299 + G \cdot 0.587 + B \cdot 0.144 \tag{1}$$

where *R*, *G*, and *B* are the brightness values of red, green, and blue colors, respectively, each ranging from 0 to 255.



№ записи	Дата			Сведения о приеме на работу, о переводах на другую работу и об увольнении (с указанием причин и со ссылкой на статью, пункт закона)	На основании чего внесена запись (документ, его дата и номер)
	число	месяц	год		
1	2			3	4

(a) the original image



№ записи	Дата			Сведения о приеме на работу, о переводах на другую работу и об увольнении (с указанием причин и со ссылкой на статью, пункт закона)	На основании чего внесена запись (документ, его дата и номер)
	число	месяц	год		
1	2			3	4

(b) the image after preprocessing

Figure 2. The image of a document page before and after preprocessing

In the second step, a Gaussian filter is applied to reduce noise. Filtering is performed in two passes using a one-dimensional kernel: first in the horizontal direction, then in the vertical. During each pass, for each pixel, the weighted average brightness value of the neighboring pixels is calculated using the Gaussian function:

$$I'(i) = \sum_{j=-k}^{k} I(i-j) \cdot G(j) \tag{2}$$

where $I(i-j)$ is the brightness of a pixel shifted by *j* positions relative to the *i*-th pixel, $G(j)$ is the Gaussian function, and *k* is the kernel size.

In image processing, the discrete approximation of the Gaussian function is calculated using the following formula:

$$G(j)=\frac{1}{\sqrt{2\pi\sigma^2}}e^{-\frac{j^2}{2\sigma^2}} \tag{3}$$

where σ is the standard deviation of the Gaussian distribution, which determines the degree of blurring. The calculation of σ was performed using the following formula:

$$\sigma=0.3\cdot((k-1)\cdot0.5-1)+0.8 \tag{4}$$

where k is the kernel size, which was set to 5 in this work.

Applying Gaussian blurring removes high-frequency content in the image (such as edges and noise), resulting in clearer segmentation. This improves the recognition of handwritten digits in the document image.

The next step is to binarize the image, which involves separating pixels into two classes: useful and background. Binarization was performed using an adaptive thresholding method, which assigns unique threshold values to the separated image fragments. Global thresholding is not suitable for scanned images of document pages, as their illumination may be unevenly distributed (Subudhi & Sahu, 2014). In such cases, a threshold value common to all pixels may lead to information loss.

To perform image binarization, the brightness of each pixel I is compared with an adaptive threshold value T, which is determined for each set of neighboring pixels within the local region of the image using the following formula:

$$T=\frac{I_{\max}+I_{\min}}{2}-C \tag{5}$$

where I_{min} is the minimum brightness value of the local image area, I_{max} is the maximum brightness value of the local image area, and C is a constant for fine-tuning the threshold value.

The result of thresholding is binarization – the conversion of a grayscale image into a binary black-and-white image:

$$I'=\begin{cases}b_0, & if\ \ I\le T\\ b_1, & if\ \ I>T\end{cases} \tag{6}$$

where $b_0=0$ is the value for black color, and $b_1=1$ is the value for white color.

If the obtained pixel brightness value is $I'=0$, the pixel is considered background. If $I'=1$, the pixel belongs to an object in the document image.

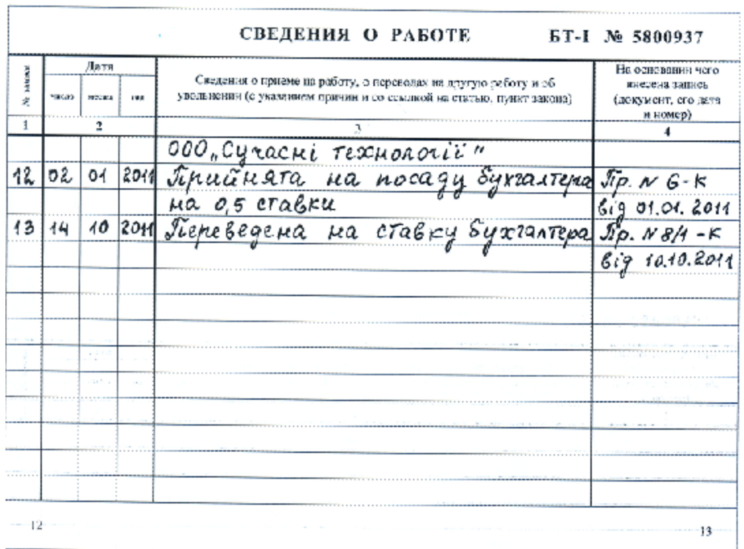
The Hough transform converts the Cartesian coordinates of image pixels (x,y) to polar coordinates (ρ,θ). Skew assessment detects deviations of the document's orientation angle from the vertical or horizontal direction. Each edge point in the image's edge map is transformed into all possible lines that can pass through it. Dominant lines create peaks in the Hough parameter space, which allows the detection of the document's skew angle θ. Then, the coordinates of each pixel (x,y) in the image can be adjusted by performing an affine transformation according to the formulas (Jipeng et al., 2011):

$$\begin{cases}x'=x\cdot\cos\theta-y\cdot\sin\theta\\ y'=x\cdot\sin\theta+y\cdot\cos\theta\end{cases} \tag{7}$$

where x' and y' are adjusted coordinates used to correct the skew.

SEGMENTATION

Segmentation is a fundamental step in recognizing handwritten digits in images of complex documents. In most cases, segmentation involves identifying homogeneous regions in the document image. Solving the task of calculating the total length of service requires identifying the table structure on the scanned pages of the document. An example of such a table is shown in Figure 3. The columns on the left of this table contain periods of employment at different companies throughout a person's working life.



СВЕДЕНИЯ О РАБОТЕ БТ-І № 5800937

№ записи	Дата: число	месяц	год	Сведения о приеме на работу, о переводах на другую работу и об увольнении (с указанием причин и со ссылкой на статью, пункт закона)	На основании чего внесена запись (документ, его дата и номер)
1	2			3	4
				ООО „Сучасні технології"	
12	02	01	2011	Прийнята на посаду бухгалтера	Пр. № 6-к
				на 0,5 ставки	від 01.01.2011
13	14	10	2011	Переведена на ставку бухгалтера	Пр. № 8/1-к
					від 10.10.2011

12 13

Figure 3. Image of a scanned document page in grayscale

To extract handwritten numerical data from the table in the document image, the boundaries of each cell needed to be identified, and the table structure updated. This was achieved by applying morphological transformations such as erosion and dilation.

In binary morphology, an image is represented as an ordered set of black and white pixels (0 or 1). The position of each pixel in the image *I* specified by two coordinates (x, y). The dilation operation consists of convolving the image with a kernel *S*, which can have any shape (most commonly a rectangle or square) and includes an anchor point – usually the center of the kernel. While scanning the kernel *S* over image *I*, the pixel value at the anchor point is replaced by the maximum pixel value within the area covered by the kernel. This process results in a new binary image $I' = I \oplus S$. The equation for the morphological dilation operator is as follows:

$$I \oplus S = \left\{ I_{\max}(x, y) \mid for\ \ every\ \ (x, y) \in S \cap I \neq 0 \right\} \tag{8}$$

The dilation operation enlarges image regions by using a kernel that defines the shape of the pixel neighborhood over which the maximum value is taken.

The erosion operation calculates the local minimum within the region covered by the kernel. As the kernel *S* scans over the image *I*, the pixel value at the anchor point is replaced with the minimum pixel value within the kernel's coverage area. This process produces a new binary image $I' = I \,\Theta\, S$. The equation for the morphological erosion operator is:

$$I \,\Theta\, S = \left\{ I_{\min}(x, y) \mid for\ \ every\ \ (x, y) \in S \subseteq I \right\} \tag{9}$$

The erosion operation shrinks regions in the image, causing small objects to disappear.

The dilation and erosion operations are often used together in binary image processing. According to set theory rules, subtracting the result of erosion from the result of dilation enables distinguishing only horizontal or only vertical lines. These lines can then be combined to form a table structure without internal content, as shown in Figure 4. This structure allows us to obtain the coordinates and dimensions of the table cells. Subsequently, digit segmentation can be performed within the cells containing dates (day, month, and year).

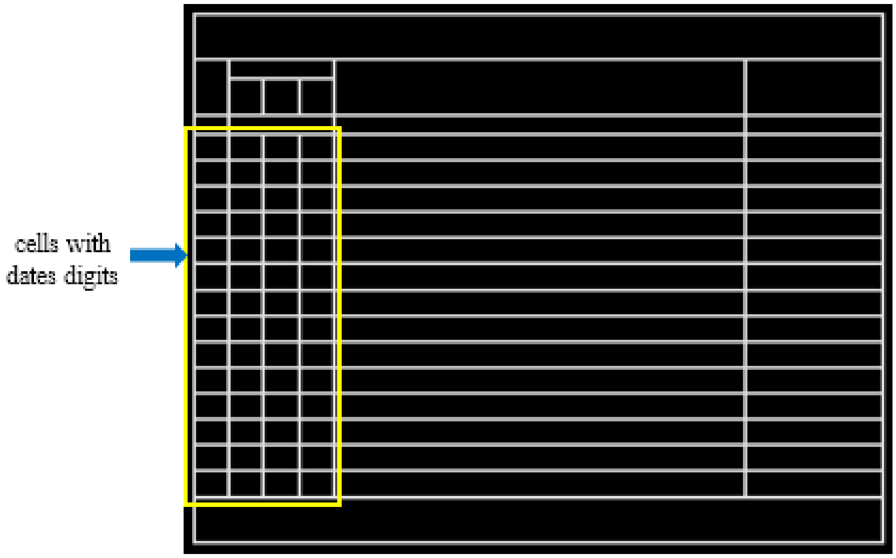


Figure 4. Positions of cells containing date digits in the reconstructed table

The size and position of the cells containing dates on the table image are known. This makes it possible to extract only those cells from the main image that contain date digits for further processing. The first cell in the row with the date contains the date number in the document. The next three consecutive cells in the table row contain the digits of the date itself, as shown in Figure 5. The first of these cells contains two digits for the day of the month, the second contains two digits for the month, and the third contains four digits for the year.

Figure 5. Table cells containing the date digits

Each cell contains digits that may be fused or slanted. The Hough transform was applied to correct the slant of the handwritten digits. The straightening was performed in a direction that minimizes the angle of inclination of the digits. The segmentation of digits within the cells was implemented using the connected component labeling method. A connected component of a binary image is the largest set of connected points such that there is a path connecting any two points within the set. Image elements are labeled so that those belonging to the same connected region are distinguished from others. Extracting connected components means assigning a unique label to each object in the image. These labels are then used as identifiers to access the objects.

The area of the image that contains the numbers in a cell can be represented as a matrix I. If $I(x, y) = 0$, then the pixel is considered background; if $I(x, y) = 1$, the pixel belongs to an object. To find the connected components of a binary discrete image, an arbitrary point where $I(x, y) = 1$ is selected. A label is assigned to this point and to its neighbors. In the next step, the neighbors of these neighbors (excluding those already labeled) are also assigned the same label. This process continues recursively. Once completed, one connected component will be fully labeled, and the process continues by selecting a new unlabeled starting point. To find it, one must move through the image until the first unmarked point is encountered. When no unmarked points remain, all objects in the image will have been labeled. Using the connected component labeling method, the background can also be marked into connected components.

Figure 6 shows how the document page image is split into distinct subimages corresponding to table cells containing date digits. First, the entire table area that may contain dates in its cells is extracted. This area includes both rows with empty cells and rows with dates corresponding to entries in the work record book. Then, areas containing rows with dates are extracted from this image. Each such area includes four cells of the same size. The first cell contains the record number in the document corresponding to the date. The next three cells contain the date digits: day, month, and year. Subimages of the regions with digits in each cell are then extracted. Finally, digit segmentation is performed on these subimages, as shown in Figure 7.

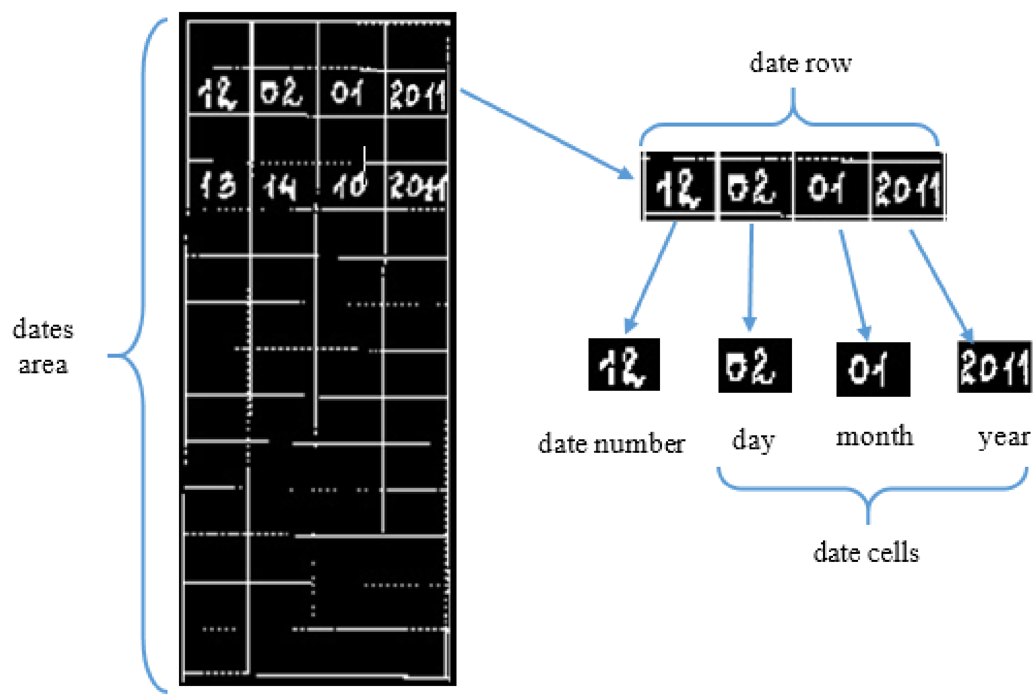


Figure 6. Splitting the image into distinct subimages containing date digits

The pages of the digitized documents used in this research have a standardized structure, which simplifies the extraction of content from cells containing dates. Unsuccessful segmentation cases mainly occur when extracting handwritten digits from images of individual cells. Figures 7 and 8 show examples of both successful and unsuccessful segmentations. Typical errors include merging digits into a single region, splitting a digit into parts, segmenting only a portion of a digit, and including parts of adjacent digits in the segmentation. To reduce the number of unsuccessful cases, digit skew correction, scaling, and morphological transforms were applied before segmentation. However, due to the high variability of handwriting, the merging of characters, and digits extending beyond cell boundaries, some errors remain uncorrected. As a result, the use of a CNN classifier under real-world conditions led to a nearly 14% drop in the recognition accuracy of individual handwritten digits. This highlights the necessity of human involvement in the validation process of the recognized dates.

Figure 7. Segmenting digits in subimages containing date digits

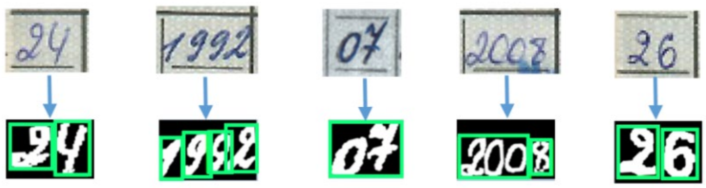


Figure 8. Original and segmented images of digits

The final step is to save each segmented digit and scale it to a size of 28×28 pixels. The actual size of the digits in the image is 20×20 pixels. This size is necessary because the MNIST database, which was used to train the convolutional neural network for handwritten digit recognition, stores images in this format. Figure 9 illustrates the scaling of digits segmented from a table cell.

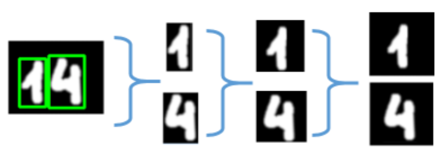


Figure 9. Scaling of the segmented digits

Segmented and scaled digit images are provided as input to the CNN classifier for character-by-character digit recognition.

Development of a CNN Model

Data Set

The neural network for handwritten digit recognition was trained on the MNIST dataset, which contains 70,000 images of handwritten digits from 0 to 9 in white on a black background. All images are 28x28 pixels in size; the size of the digits in the image is 20x20 pixels (Memon et al., 2020). Each digit is centered in the image so that its center of mass aligns with the image center, as shown in Figure 10. The training set contains 60,000 images of handwritten digits, and the test set contains 10,000.

Figure 10. Sample handwritten digit images from the MNIST dataset

Typical CNN Architecture for Handwritten Digit Recognition

A typical architecture of a convolutional neural network for recognizing handwritten digits is shown in Figure 11. The input layer of the CNN receives the image to be recognized. This image is a data array of size $n \times n$, where each pixel holds a value representing its intensity. In the context of the task at hand, this is a binarized image of a handwritten digit with a size of 28×28 pixels.

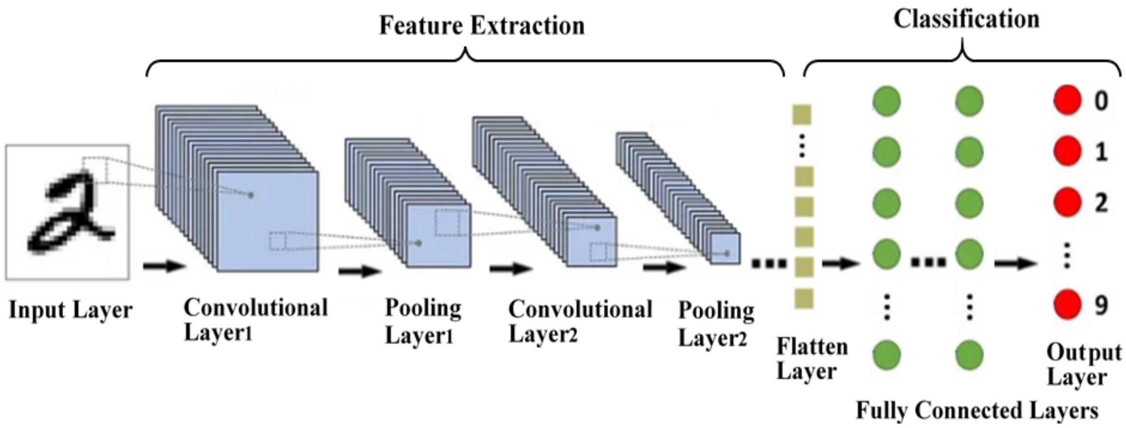


Figure 11. Typical CNN architecture for recognizing images of handwritten digits

Next, the CNN architecture contains convolutional layers and pooling layers. Convolutional layers are the main building blocks of CNNs, where the convolutional operation is applied to the outputs of the neurons of the previous layer. It is performed using kernel slides, which are small-sized weight matrices of size $m \times m$ (usually from 3×3 to 7×7). To recognize binarized images of handwritten digits, two-dimensional convolutional layers Conv2D (input shape ($28 \times 28 \times 1$)) are used.

In convolutional layers, each neuron is connected to a limited number of neurons of the previous layer. The essence of convolution is that the kernel performs element-wise scalar multiplication with that part of the previous layer's outputs over which it is currently positioned. The result is summed and is the convolution value. The kernels apply convolution to all outputs of the previous layer, sliding with a given stride k (most often equal to 1), thereby forming feature maps.

Different kernels extract various image features (such as lines at different angles, edges, etc.). Kernels are the basis of local connections that have similar parameters for generating feature maps. The initial convolutional layers are responsible for extracting low-level features (simple lines, edges, boundaries), while the deeper layers combine these features into more complex shapes. During neural network training, this progression through successive convolutional layers enables the network to recognize increasingly complex patterns.

The first convolutional layer performs convolution with a stride of $k = 1$ over an input image of size $n \times n$ using a kernel of size $m \times m$, producing a feature map of size $(n - m + 1) \times (n - m + 1)$ at the output. When the step size increases, a feature map of smaller sizes is obtained:

$$(\frac{n-m}{k}+1) \times (\frac{n-m}{k}+1) \tag{10}$$

It is advisable to choose the kernel size so that the number of feature maps is even. This helps avoid losing information when reducing dimensionality in the next pooling layer. In this work, an input image size of 28×28 pixels, a kernel size of 3×3, and a convolution stride k equal to 1 were used. Therefore, the size of the feature maps in the first convolutional layer is 26×26 pixels. To preserve the original image size, it is necessary to pad the borders of the image. The amount of padding is controlled by the parameter P. Applying zero padding $P = 1$ to the original 28×28 image increases its spatial dimensions 30×30. Then, after applying convolution with a 3×3 kernel and stride $k = 1$, the size of the feature maps in the first convolutional layer will equal 28×28. Padding in subsequent convolutional layers is chosen so that spatial resolution is preserved after convolution.

The number of kernels and feature maps in a convolutional layer is a hyper-parameter, chosen as a power of two, with the number of feature maps increasing in subsequent convolutional layers. The number of feature maps most often increases by a factor of 2. The weights of the convolutional kernels are parameters determined during network training.

Pooling layers are placed after the convolutional layers. Each feature map from the previous convolutional layer is divided into regions of size $w \times w$, from which either the maximum or the average value is selected. The pooling operation can be performed with a stride size, which is most often set equal to the pooling window size $k = w$. In this work, a pool 2×2 with a stride size $k = 2$ is applied using the maximum value selection method. The resulting values form the MaxPool2D pooling layer. This reduces the dimensionality of the feature maps from the previous convolutional layer by half. Feature map compression is achieved by discarding less important information while preserving the main features.

The output feature maps of the last pooling layer are typically converted into a one-dimensional (1D) array of numbers in the flatten layer. The output vector from the flatten layer is fed into the first fully connected layer of the CNN. Fully connected layers are placed after the convolutional and pooling layers in the CNN architecture. Each neuron in a fully connected layer is connected to all neurons in the previous layer. Each connection has its own weight, which is determined during network training.

The recognition of digits from their handwritten images involves classifying them into ten predefined classes, each corresponding to the image of a specific digit (0–9). Therefore, the output layer of a convolutional neural network, which is the last fully connected layer, contains 10 neurons.

Choosing Activation Functions

The type of activation function determines the functionality of the neural network and the training method. The ReLU activation function was used in the convolutional layers of the CNN, as it is currently considered the simplest and most efficient in terms of computational complexity. The scalar result of each convolution *x* is fed into the ReLU activation function:

$$f(x)_{\mathrm{Re}LU} = \max(0, x) \tag{11}$$

ReLU returns the value of *x* if *x* is positive and 0 otherwise. In the hidden, fully connected layers of the CNN, ReLU is also applied as an activation function.

The output layer of the convolutional neural network has 10 neurons for recognizing handwritten digits. The Softmax activation function is used for this layer. Softmax calculates the probabilities that the input image belongs to each of the possible classes. The output of the i-th neuron of the output layer is determined by the Softmax formula:

$$\hat{y}_i = \frac{\exp(y_i)}{\sum_{j=1}^{m} \exp(y_i)} \tag{12}$$

where $\hat{y}_i$ is the generated (predicted) output value of the *i*-th neuron ($\sum_{j=1}^{m} \exp(y_i)$), y_i is the actual output value of the *i*-th neuron, and *m* is the number of neurons in the output layer ($m = 10$).

When recognizing an image of a handwritten digit, the output of the CNN is a vector $Y = (\hat{y}_1, ..., \hat{y}_i, ..., \hat{y}_m)$, whose components $\hat{y}_i$ represent the probabilities that the digit belongs to the i-th class (i.e., that it is the i-th digit from 0 to 9).

Training and Evaluation of a CNN Model

The choice of optimizer is crucial when training a CNN, as it significantly affects the performance of the final architecture. Since no universal optimization algorithm exists, this work explores several optimizers to develop a high-quality and accurate CNN model for handwritten digit recognition.

The basic method for parameter optimization during neural network training is stochastic gradient descent (SGD) with backward propagation of errors. However, SGD has several drawbacks. These include a fixed learning rate parameter, the attenuation of gradients with increasing network depth, and a high chance of falling into a local minimum.

An improved version of SGD is used to address these issues. It incorporates the Nesterov Accelerated Gradient and the momentum concept. The use of the Nesterov Accelerated Gradient takes into account the direction of the gradient in the next step, which allows for more efficient parameter adjustment. Using the momentum concept reduces fluctuations by adding the fraction of the previous update step to the current gradient. This helps accelerate SGD in the right direction and get out of local lows.

Adaptive methods were also used to train the neural network during the hyperparameter selection stage: RMSprop (Root Mean Square Propagation) and Adam (Adaptive Moment Estimation). These methods adapt the learning rate individually for each parameter, which helps optimize complex parameters more effectively. The RMSprop optimizer adaptively adjusts the network's learning rate by stimulating gradients that change in the same direction for several epochs and suppressing those that frequently change signs. The Adam optimizer is also an effective algorithm for training convolutional neural networks, as it combines an adaptive learning rate and an optimization method to find optimal values for model parameters efficiently.

In the MNIST dataset, the classes of handwritten digits are balanced. Therefore, the categorical cross-entropy loss function was used to measure the network's performance, and the accuracy metric was used to evaluate its quality. Categorical cross-entropy is used for multiclass classification. During training, the CNN parameters are iteratively updated to minimize the loss function:

$$loss = -\sum_{j=1}^{n} \sum_{i=1}^{m} (y_i^j \cdot \log(\hat{y}_i^j)) \tag{13}$$

where m is the number of classes, each corresponding to a separate digit (m=10); n is the total number of objects (images of handwritten digits); y_i^j is the actual belonging of the j-th image of the handwritten digit to the i-th class (1 if it belongs, otherwise 0); $\hat{y}_i^j$ is the predicted probability of assigning the j-th image of a handwritten digit to the i-th class.

By iteratively applying backward propagation of errors and parameter updates, the network gradually adjusts its parameters to minimize the loss function and improve predictive performance. To optimize the training process, dropout layers were included in the neural network structure, and Batch Normalization was applied. Batch Normalization improves the performance and stability of CNNs by re-centering and scaling the data as it passes through the network (Ioffe & Szegedy, 2015; Santurkar et al., 2019). It helps address the problem of internal covariance shift, which consists of changing the distribution of connection weights of each layer during training. This complicates the learning process since changing the parameters of previous layers during training affects the output values of the neurons of the current layer. In convolutional layers, Batch Normalization is applied to the outputs of each feature map while preserving the spatial structure of the data.

Using dropout layers is one of the ways to combat overfitting during the training of a neural network using the stochastic gradient descent method. Dropout regularization refers to a specific level of neurons and consists of excluding, with a certain probability, a proportion of neurons of this network. Training is performed on this sparser network: a gradient step is taken, after which the excluded neurons are reintroduced. Thus, at each step of stochastic gradient descent, one of the possible 2^n network architectures is configured, where n is the total number of neurons. During the testing of a neural network, neurons are no longer excluded, but the output of each neuron is multiplied by $(1-p)$. In CNNs, dropout layers are most commonly used with p values ranging from 0.1 to 0.3.

The effectiveness of the CNN classifier model was evaluated using the accuracy metric, which describes the overall accuracy of the model's prediction across all classes. Accuracy is calculated as the ratio of correctly predicted classes to the total number of predictions:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{14}$$

where TP and TN are the numbers of true positive and true negative results, respectively, while FP and FN are the numbers of false positive and false negative results.

THE PROPOSED CNN ARCHITECTURE

The convolutional neural network model proposed in this paper for handwritten digit recognition is presented in Table 1 as a sequence of different layer types. The CNN architecture contains four convolutional layers with ReLU activation functions and a kernel size of 3×3, which apply convolution operations with a stride of $k=1$, forming feature maps. The model also includes two pooling layers with a pooling window size of 2×2 and a stride of 2.

Table 1. Architecture of the proposed CNN model for the recognition of handwritten digit images

Layer (type)	Activation shape	Activation size	Number of parameters
Input image	(28,28,1)	784	–
Conv2D Layer (kernel 3x3, k=1)	(28,28,32)	25,088	320
Conv2D Layer (kernel 3x3, k=1)	(28,28,32)	25,088	9,248
MaxPooling2D Layer (pool 2x2, k=2)	(14,14,32)	6,272	0
Dropout Layer (0.25)	(14,14,32)	6,272	0
Conv2D Layer (kernel 3x3, k=1)	(14,14,64)	12,544	18,496
Conv2D Layer (kernel 3x3, k=1)	(14,14,64)	12,544	36,928
MaxPooling2D Layer (pool 2x2, k=2)	(7,7,64)	3,136	0
Flatten Layer	(3136,1)	3,136	0
Dense Layer	(128,1)	128	40,1536
Dropout Layer (0.5)	(128,1)	128	0
Dense Layer (output)	(10,1)	10	1,290

A binarized image of a handwritten digit with a size of 28×28 is fed into the input layer of the network. Zero-padding with $P=1$ is applied to the input image. Next, the CNN contains two Conv2D convolutional layers, forming 32 feature maps of size 28×28. The second convolutional layer is followed by a MaxPool2D pooling layer, which reduces the size of the feature maps to 14×14.

The next CNN block contains two Conv2D convolutional layers, each preceded by zero-padding with $P=1$. Each of these layers produces 64 feature maps of size 14×14. This is followed by a pooling layer, which reduces the feature map size to 7×7.

Then, a Flatten layer is applied, which converts the array of two-dimensional matrices into a one-dimensional vector. The model includes two dropout layers configured to randomly exclude a fraction of the neurons in the layer with rates of 0.25 and 0.5 to prevent overfitting.

The CNN architecture also contains two fully connected (Dense) layers. The first fully connected layer has 128 neurons and uses the ReLU activation function. The second fully connected layer is the output layer, which has 10 neurons and uses the Softmax activation function to output probabilistic predictions for each class.

The optimal configuration of neural network hyperparameters and regularization techniques was selected during the development of the classifier model. This study examined CNN architectures with different configurations of layers and tuning parameters, which are presented in Table 2.

Table 2. CNN models investigated during the hyperparameter selection stage

Layer (Type)	Different CNN architectures						
	1	2	3	4	5	6	7
Input image, Activ_Shape: (28,28,1)	+	+	+	+	+	+	+
Conv2D, Activ_Shape: (28,28,32)	+	+	+	+	+	+	+
BatchNormalization, Activ_Shape: (28,28,32)	–	–	+	–	–	–	+
Conv2D, Activ_Shape: (28,28,32)	–	–	–	+	+	+	+
BatchNormalization, Activ_Shape: (28,28,32)	–	–	–	–	–	–	+
MaxPooling2D, Activ_Shape: (14,14,32)	+	+	+	+	+	+	+
Dropout (0.25), Activ_Shape: (14,14,32)	–	+	+	+	+	+	+
Conv2D, Activ_Shape: (14,14,64)	–	–	–	–	+	+	+
BatchNormalization, Activ_Shape: (14,14,64)	–	–	–	–	–	–	+
Conv2D, Activ_Shape: (14,14,64)	–	–	–	–	–	+	+
BatchNormalization, Activ_Shape: (14,14,64)	–	–	–	–	–	–	+
MaxPooling2D, Activ_Shape: (7,7,64)	–	–	–	–	+	+	+
Flatten, Activ_Shape:(6272,1)	+	+	+	+	–	–	–
Flatten, Activ_Shape:(3136,1)	–	–	–	–	+	+	+
Dense, Activ_Shape: (128,1)	+	+	+	+	+	+	+
Dropout (0.5), Activ_Shape: (128,1)	–	+	+	+	+	+	+
Dense (output), Activ_Shape: (10,1)	+	+	+	+	+	+	+

During the training of CNN models, 10% of the training dataset was allocated for validation. A comparative analysis was performed on the models, which were evaluated based on their accuracy, loss, and computational complexity.

The results of the quality assessment of models with different configurations are presented in Table 3. The analysis of the results shows that the model's accuracy increases with the number of convolutional layers in the CNN structure. The inclusion of dropout layers in the architecture helped prevent overfitting of the neural network model. The addition of Batch Normalization layers was accompanied by a decrease in the accuracy of handwritten digit recognition. In models without Batch Normalization, smoother training was observed. Therefore, Batch Normalization layers were not included in the final CNN model.

Figures 12 and 13 present the training and validation accuracy graphs of the CNN model over 25 epochs using different optimizers. The results show that including the parameter $momentum = 0.9$ for the SGD optimizer improves the accuracy of handwritten digit recognition. However, significantly higher accuracy is achieved with the Adam and RMSprop optimizers.

Figures 14 and 15 show the training and validation loss graphs of the CNN model over the number of epochs using different optimizers. It should be noted that using the SGD optimizer with the parameter $momentum = 0.9$ improves performance and minimizes the loss function. However, the Adam and RMSprop optimizers reduce the loss function much faster, achieving significantly lower loss values.

Table 3. Quality assessment of CNN models with different configurations

CNN architecture	Accuracy			Loss		
	Training	Validation	Test	Training	Validation	Test
1	0.9999	0.9887	0.9878	0.0004	0.0746	0.0546
2	0.9915	0.9913	0.9899	0.0237	0.0382	0.0360
3	0.9912	0.9915	0.9897	0.0260	0.0402	0.0446
4	0.9947	0.9927	0.9927	0.0153	0.0407	0.0301
5	0.9955	0.9942	0.9939	0.0137	0.0320	0.0235
6	0.9968	0.9938	0.9955	0.0098	0.0363	0.0257
7	0.9960	0.9945	0.9939	0.0123	0.0333	0.0280

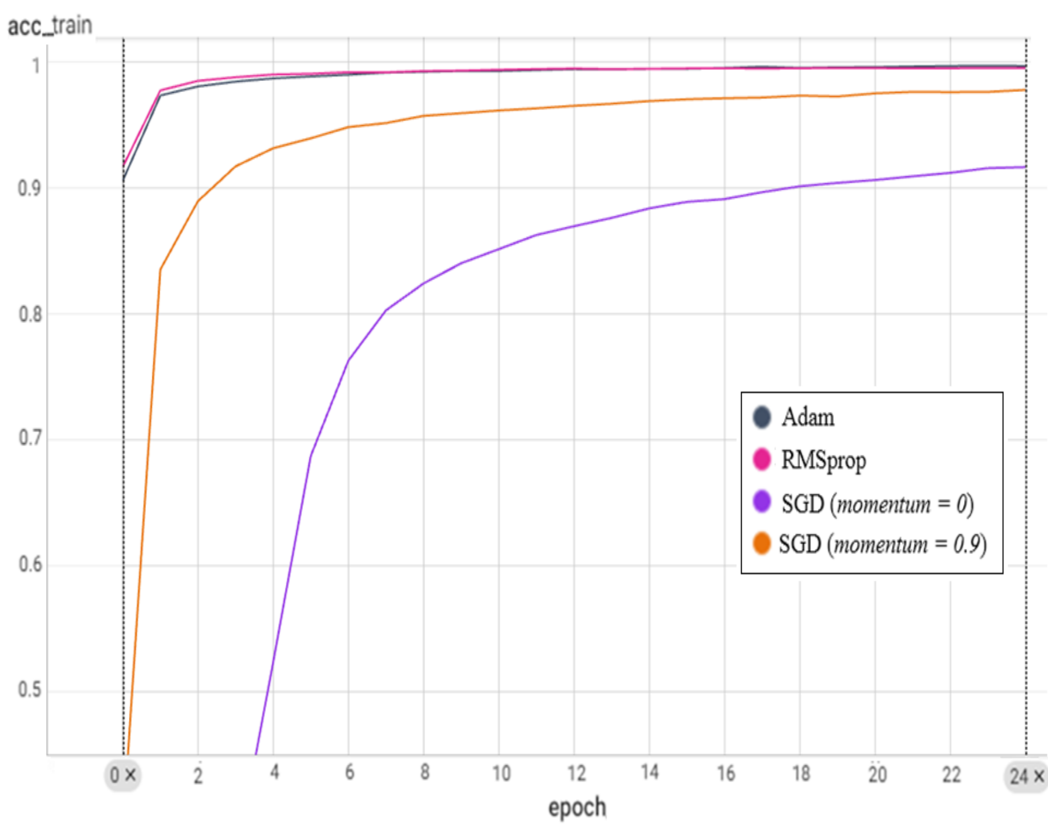


Figure 12. Accuracy of the CNN model with different optimizers for the training data

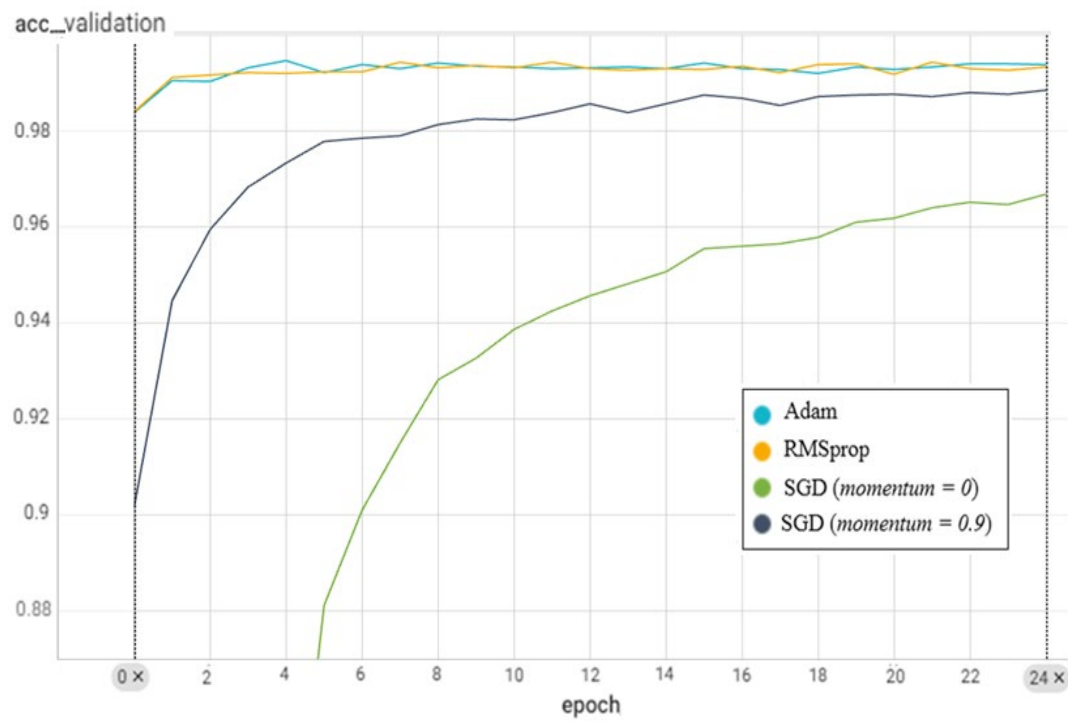


Figure 13. Accuracy of the CNN model with different optimizers for the validation data

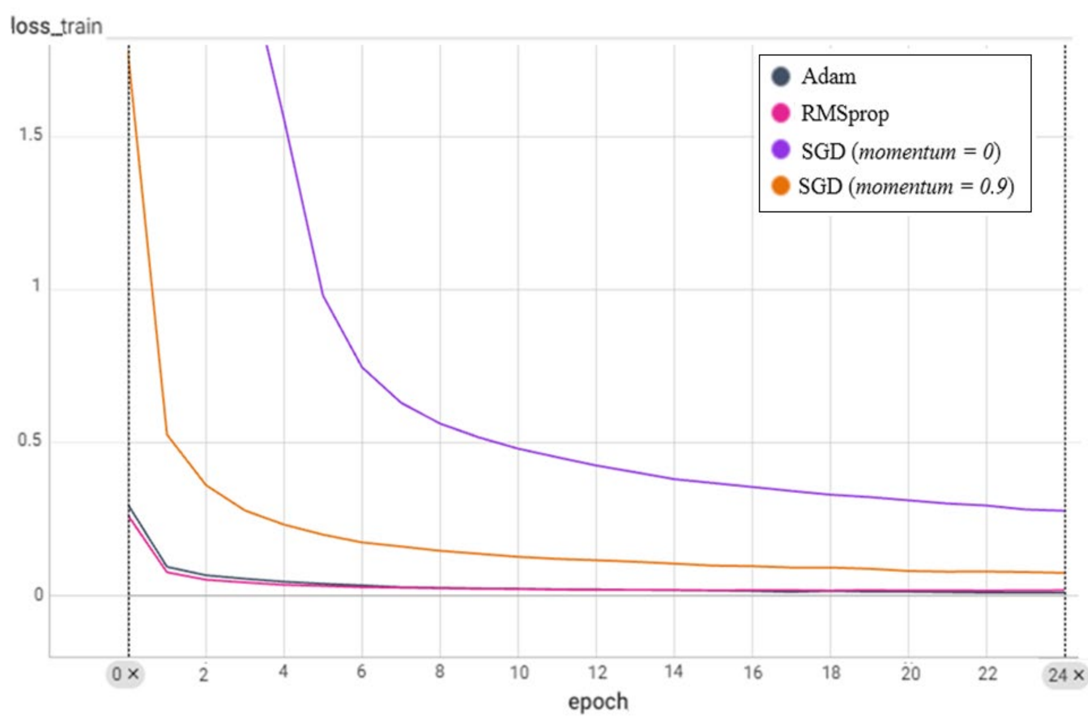


Figure 14. Loss values of the CNN model with different optimizers for the training data

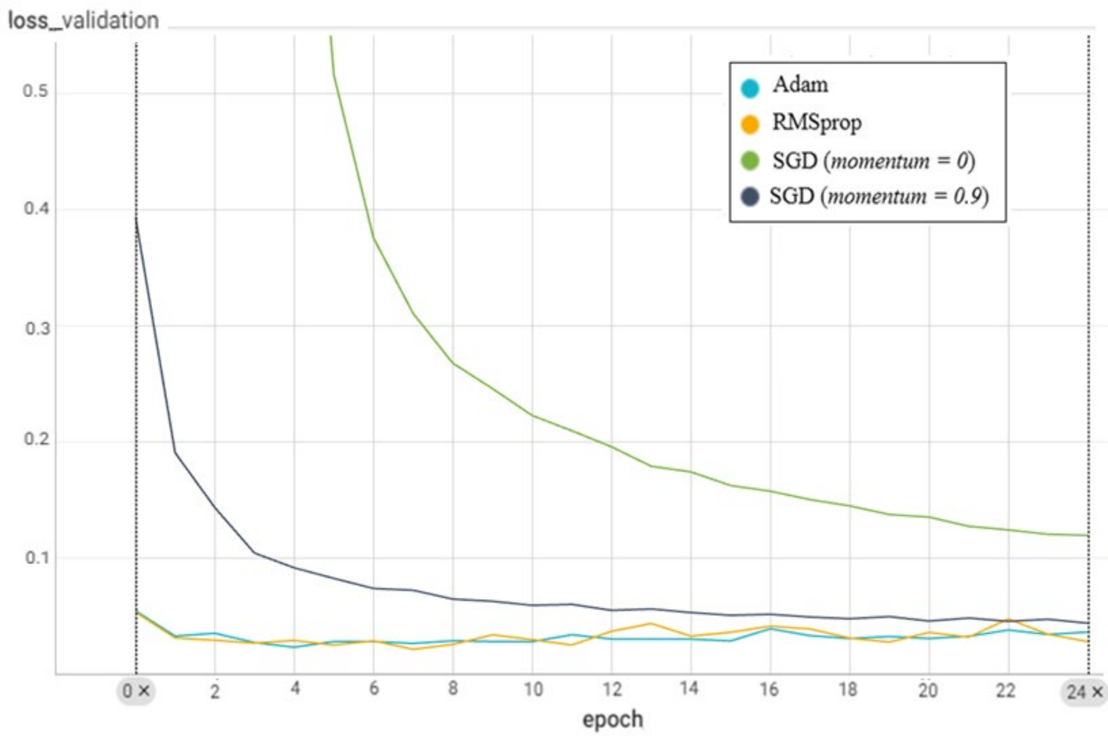


Figure 15. Loss values of the CNN model with different optimizers for the validation data

Table 4 shows the quality evaluation of CNN models using different optimizers. The study demonstrated that the CNN model has better performance metrics when using the Adam optimizer. Therefore, a CNN model trained with the Adam optimizer was selected for handwritten digit recognition.

Table 4. Accuracy and loss values of the CNN models using different optimizers

Optimizer	Training		Validation		Test	
	Accuracy	Loss	Accuracy	Loss	Accuracy	Loss
SGD (*momentum* = *0*)	0.9165	0.2775	0.9668	0.1196	0.9555	0.1382
SGD (*momentum* = *0.9*)	0.9778	0.0743	0.9885	0.0440	0.9865	0.0389
RMSprop	0.9951	0.0176	0.9933	0.0281	0.9951	0.0221
Adam	0.9968	0.0098	0.9938	0.0363	0.9955	0.0257

Figure 16 shows the accuracy and loss graphs of the CNN model during the training and validation phases. The quality of the CNN model in recognizing handwritten digit images is shown on the confusion matrix in Figure 17. Analysis of the confusion matrix shows that the model performs well in classifying handwritten digits. The most common errors are misclassifying the digit 5 as 3 and the digit 9 as 4.

The accuracy and loss values for the built CNN architecture are 0.9968 and 0.0098 for the training data, and 0.9938 and 0.0281 for the validation data. The accuracy on the test data was 0.9955, with a 95% confidence interval of [0.9953; 0.9977], and the loss value was 0.0257.

These results indicate that the model provides high accuracy in recognizing handwritten digit images.

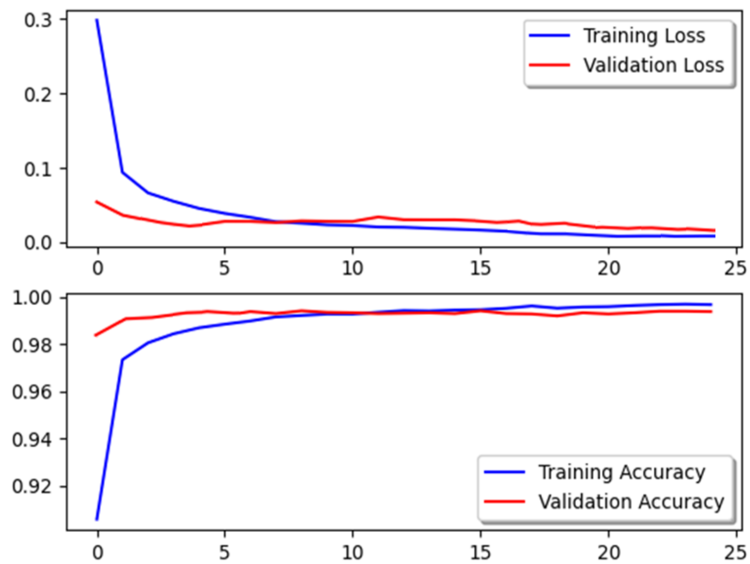


Figure 16. Accuracy and loss curves of the CNN model during the training and validation phases

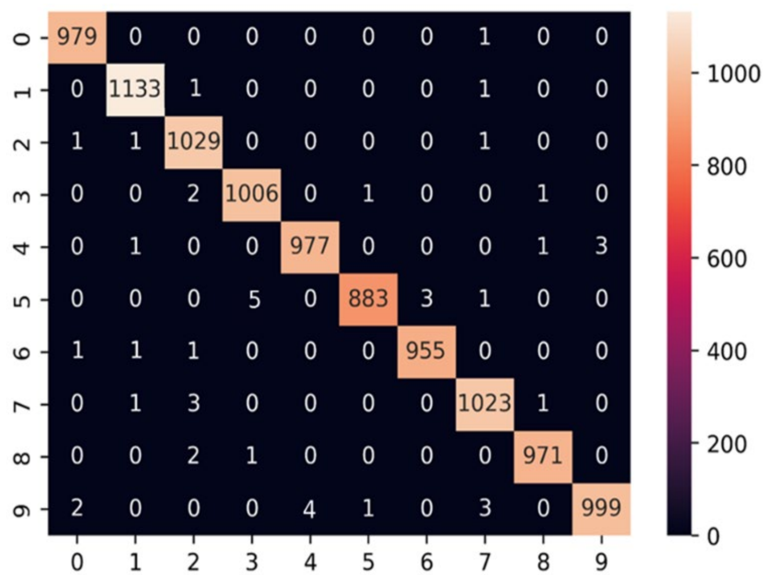


Figure 17. Confusion matrix for handwritten digit recognition on the test data

Calculation of Total Work Experience

Application for Processing Digitized Documents

To automate the extraction and processing of numerical information from scanned pages of the employment record book, a software application was developed using the proposed CNN model and implemented in Python. The graphical interface was created using the cross-platform Kivy library. The TensorFlow library and the Keras framework were used to develop the CNN model for handwritten digit recognition. During the hyperparameter tuning stage, the training process was monitored using TensorBoard. Image preprocessing, binarization, and segmentation were implemented using functions from the OpenCV library. The NumPy and Matplotlib libraries were used to manage data and visualize results.

The developed system calculates a person's length of service by recognizing handwritten digits from scanned images of pages from their employment record book. These pages contain dates corresponding to periods of employment at different companies throughout the person's working life. Before using the application, it is necessary to download files containing the scanned images of employment record book pages. These images are stored in the database of the Integrated Comprehensive Information System of the Pension Fund of Ukraine (ICIS PFU) for individuals applying for a pension. The main window of the application is shown in Figure 18.

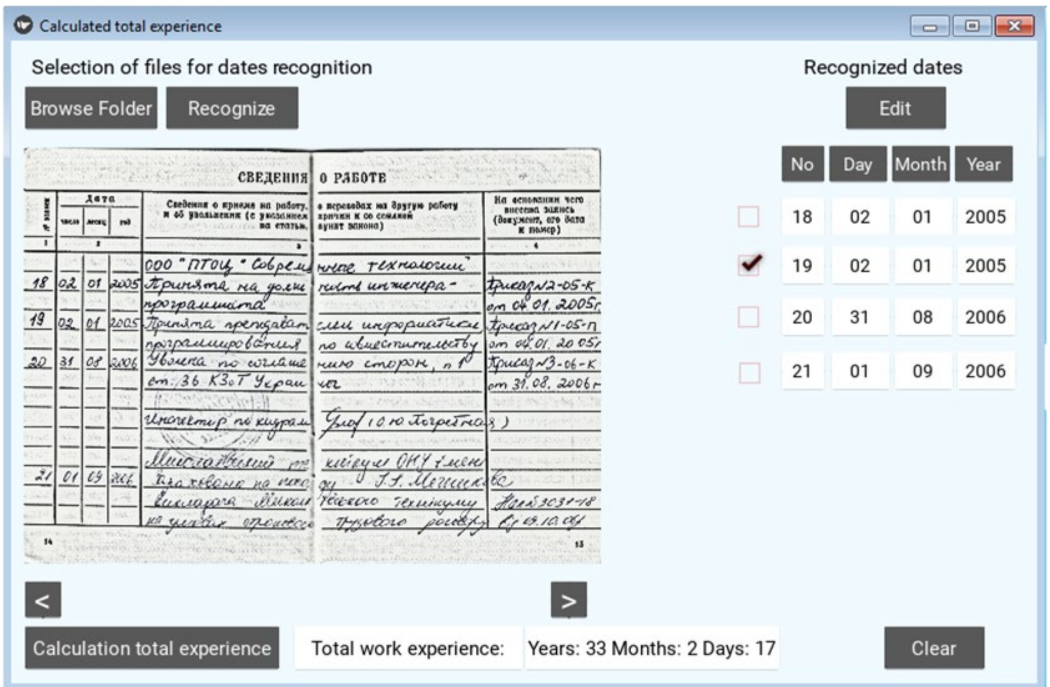


Figure 18. The application window for automated calculation of a person's work experience

By clicking the *Browse Folder* button, a folder window opens, displaying the uploaded scanned copies of the document pages. These pages are sorted in the order in which they appear in the employment record book. To proceed, one needs to select the required files by highlighting them and clicking the *Open* button. The scanned images of the document pages will then be ready for further processing.

Clicking the *Recognize* button initiates preliminary image processing of the uploaded pages, including segmentation and the extraction of handwritten digits from fields containing dates. The segmented digit images are then passed to the input of a neural network for recognition. Based on the recognized digits, a sequence of dates is formed, taking into account the format used in the table cells of the document.

In the application window, the image of the first page of the employment record book with entries is displayed on the left. Using the buttons at the bottom of the window, it is possible to sequentially browse through all the pages of the employment record book. The right side of the application window displays the entries with dates that were recognized on the page shown in the left part of the window.

POST-PROCESSING AND CALCULATION OF WORK SENIORITY

The post-processing stage includes verifying the correctness of handwritten digit recognition, forming a chronological sequence of dates, and correcting any errors detected after processing the scanned document images. This stage cannot be fully automated. To obtain accurate results, the extracted numerical data must match the original data in the document exactly. Therefore, human involvement in the verification and correction process is critically important. During verification, incorrectly recognized digits are corrected. Additionally, dates unrelated to the beginning or end of employment at a specific company are removed. These may include entries such as transfers to another position within the same organization, as well as information about awards and incentives.

The developed web application enables the verification, correction, and confirmation of recognized data. This includes the ability to view document page images sequentially and compare the dates on them with the recognized dates. To edit the generated sequence of dates, select the checkbox next to the entry that requires changes and click the *Edit* button. A drop-down list will open, offering the following options:

- *Change* – to edit the digits of the selected date.
- *Delete* – to remove the entry with that date.
- *Add* – to create a new entry if needed.

After verification and correction, the extracted dates are processed according to their intended purpose. To initiate this process, click the *Calculation Total Experience* button located at the bottom of the application window. The calculation of the exact number of years, months, and days that make up a person's total work experience will begin. The calculated work experience will be displayed in the application window, as shown in Figure 18. Clicking the *Clear* button allows you to quickly clear the fields and begin working with another document. The system calculates work experience accurately, as confirmed by testing on a sample of employment periods containing dates from various time ranges.

EXPERIMENTAL RESULTS

The experimental validation of the proposed methodology was carried out in one of the branches of the Pension Fund of Ukraine. The developed application automates the work of employees during the registration of a person for retirement. This process involves handling a large number of digitized and paper documents. This process is time-consuming, as both the extraction of information from digitized documents and the calculation of total seniority are performed manually. To evaluate the efficiency and accuracy of handwritten digit recognition using the application, 211 images of employment record book pages were processed. Detailed information is presented in Table 5.

It was found that the accuracy of handwritten digit recognition was 85%, which is lower than the accuracy achieved when recognizing isolated handwritten digits using the developed CNN model. The decrease in accuracy occurs at the image segmentation stage of digitized documents. However, this is still a good result, considering that the handwritten digits were extracted from complex document pages containing both printed and handwritten text, tables, and non-textual information. Moreover, the obtained accuracy is higher than the results obtained by other researchers for data extraction from complex documents (Uchida et al., 2022).

Productivity was evaluated by comparing the time required for manual extraction of dates and calculation of the total length of service with the time required to perform the same operations using the developed application. On average, manually extracting records containing dates from a single employment record book and calculating the total work experience took 42 minutes. In contrast, using the application reduced this time to 6 minutes. The results show that replacing the manual calculation of total work experience with an automated approach significantly reduces document processing time

– by a factor of 7.7. The conducted study demonstrates that implementing the proposed methodology in the work of public administration bodies contributes to increased operational efficiency.

Table 5. Experimental validation of the developed methodology

Employment record books	Pages (*quantity*)	Dates		Time to calculate seniority (*min*)		Recognition errors	
		Records (*quantity*)	Digits (*quantity*)	Manual calculating	With the application	Digits (*quantity*)	%
1	10	30	240	40	5,5	36	15
2	12	32	256	43	4,5	35	14
3	8	26	208	35	6	34	16
4	14	34	272	45	4,4	33	12
5	9	27	216	36	5,5	32	15
6	11	36	288	48	6	46	16
7	8	29	232	38	6,5	39	17
8	9	24	192	32	5	26	14
9	11	33	264	44	5,5	40	15
10	13	39	312	52	5,5	47	15
11	9	30	240	40	6	40	17
12	12	43	344	58	6,6	58	17
13	10	27	216	36	4,5	29	13
14	11	26	208	35	4,5	25	12
15	9	27	216	36	5,3	32	15
16	12	37	296	48	5,5	44	15
17	13	49	392	62	6,6	71	18
18	10	39	312	52	7,2	56	18
19	9	24	192	32	5	26	14
20	11	26	208	35	4,4	25	12
Total	211	628	5104	847	110	774	
Average							15

FINDINGS & DISCUSSION

This study explored approaches to reducing bureaucratic delays in processing digitized documents in Ukraine's government sector. The focus was on the hybrid use of OpenCV segmentation methods and CNN models. The effectiveness of this approach is supported by Saritha et al. (2020), who argue that combining OpenCV with CNNs enables efficient processing of documents with non-standard layouts and improves symbol recognition accuracy. Unlike template-based approaches, which are limited to predefined patterns, CNNs can learn from a wide range of writing styles, making them more adaptable and robust.

This research builds on previous findings by using a hybrid approach. It combines OpenCV techniques for efficient preprocessing and segmentation of digitized document page images with a CNN model for recognizing the extracted digits. Various optimizers and CNN architectures were tested on the MNIST dataset to develop the classifier model. The goal was to determine the most effective hyperparameter configuration. This approach is consistent with the studies by Ahlawat et al. (2020) and Cui and Bai (2019), which highlight the importance of selecting optimal hyperparameters to achieve high recognition accuracy and improve neural network performance. The model achieved its best performance when trained with the Adam optimizer. Its accuracy also improved as the number of convolutional layers in the CNN architecture increased.

The developed CNN classifier model for handwritten digit recognition demonstrated high recognition accuracy, achieving 99.68% on the training data and 99.55% on the test data. These results exceed those obtained by Saqib et al. (2022) on the MNIST dataset. However, the highest recognition accuracy can be achieved using ensemble methods, transformer-based models, or hybrid approaches combining transformers and CNNs, as confirmed in studies by Aparna and Rajchandar (2024), Mahadevkar et al. (2024), and Tüselmann and Fink (2024). Nevertheless, these methods are associated with high computational complexity. Saqib et al. (2022) emphasize the importance of maintaining an optimal balance between high model accuracy and computational complexity during training. This approach was also followed in the present study. The constructed classifier model achieves high accuracy in recognizing handwritten digits with low computational complexity. However, the model can be further improved through the use of hybrid deep learning AI-based techniques.

In this study, significant attention was given to the preprocessing and segmentation of digitized document images using OpenCV methods. This aligns with the findings of Boiangiu et al. (2020), Eken et al. (2019), Johnson et al. (2018), Xue et al. (2020), and Q. Ye and Doermann (2015), who emphasized the effectiveness of OpenCV techniques at these stages. It also supports the conclusions of Baviskar et al. (2021), who highlighted the importance of preprocessing and segmentation for improving recognition accuracy in digitized documents.

Segmentation methods based on morphological operations (erosion and dilation) and the connected components method were applied to extract handwritten digits from table cells. This approach helped overcome the limitations of handwritten character extraction in complex documents, as indicated by Liu et al. (2016) in their study.

Experimental validation of the proposed methodology showed that the accuracy of recognizing handwritten digits from images of complex documents was 85%. This represents a 14% decrease in the CNN classifier's accuracy due to segmentation challenges in individual handwritten digits extracted from table cell subimages. However, this result is higher than the accuracy typically achieved by traditional OCR methods for recognizing handwritten symbols in complex documents, which usually does not exceed 80%. This value also surpasses the result obtained by Reul et al. (2019) in recognizing historical printed publications using ensemble methods.

As stated by Preethi et al. (2021) and Tüselmann and Fink (2024), deep learning methods that use pixel-level image segmentation show promising results in recognizing documents with non-standard layouts. Combining convolutional segmentation models with transformer-based models enables even greater accuracy through context recognition. This has been confirmed in the studies by Mahadevkar et al. (2024) and Tüselmann and Fink (2024), and it represents a promising direction for further improvement of the proposed methodology. However, these methods require significant computational resources and processing time, as highlighted by Saqib et al. (2022). Moreover, the recognition accuracy for handwritten symbols in complex documents still remains within the range of 85–92%.

The 85% accuracy achieved in this study highlights the need for human involvement in the validation of recognized digits. Therefore, at the post-processing stage, the methodology includes manual verification and correction of recognized digits before calculations are performed. Accurate results can only be obtained if the extracted numerical data fully match their original values in the document. This approach is consistent with the findings of Reul et al. (2019) and Boliubash (2024), who emphasize the importance of manual validation during post-processing when dealing with complex and important information.

The application developed to implement the proposed methodology automates the processing of scanned images of Ukrainian citizens' workbook pages. It focuses on extracting handwritten digits representing dates and calculating the total length of service. This indicator is used in the formula for calculating pensions in the branches of the Pension Fund of Ukraine.

Information on employment periods over the past 20 years is available to PFU employees through the Electronic Register of Insured Persons within the State Register of Compulsory Social Insurance. However, up to 80% of entries in the workbooks of individuals currently retiring were made before the introduction of compulsory state pension insurance in the country. As a result, PFU employees still process these records manually.

This study achieved a 7.7-fold reduction in the processing time of digitized documents, which is impressive as it compares manual data extraction with processing using modern OCR methods. This result confirms the conclusion of Reul et al. (2019), who claim that software applications for the automated processing of digitized documents significantly accelerate their processing.

However, the results reported by Reul et al. (2019) do not provide a comprehensive assessment, as they do not include the time required for manual validation. According to their data, the processing time per page was 1-2 minutes. In contrast, this study achieved a better result: the average processing time per page was 0.5 minutes, including manual validation during post-processing.

Expanding the application of the developed methodology within Ukrainian public administration bodies to automate the processing of digitized documents will help improve their operational efficiency. This aligns with the conclusion of Yevtushenko (2024), who states that the digitalization of public administration requires fundamentally new innovations and IT solutions to increase labor productivity and enhance communication between state structures and the population. The empirical results obtained in this study provide valuable insights for policymakers and practitioners working towards the development of e-government in the country.

CONCLUSIONS

The digital transformation of public administration necessitates identifying methods to enhance operational efficiency by automating document processing and analysis. Manual entry and processing of data extracted from scanned document images remain significant bottlenecks. To address this challenge, this article proposes a methodology for the automated calculation of employment seniority for individuals applying for pensions at the Pension Fund of Ukraine.

Convolutional neural network models with different architectures were investigated to identify the optimal combination of hyperparameters for handwritten digit recognition. The models were trained on the MNIST dataset, which contains 70,000 images of handwritten digits, using various optimizers. Model accuracy improved with an increasing number of convolutional layers in the CNN architecture and when the Adam optimizer was employed. The addition of dropout layers further enhanced the training process by reducing overfitting. The trained CNN model demonstrated high accuracy in digit recognition: 99.68% on the training set and 99.55% on the test set.

An application was developed to improve the recognition of digits in images of employment record book pages containing printed, handwritten, and non-textual information. This application implemented methods for segmenting handwritten digits from table cells using morphological transformations such as erosion and dilation, as well as the connected component labeling method. Image preprocessing included converting images to grayscale, reducing noise using a Gaussian filter, binarization, and applying the Hough transform to correct skew.

The accuracy of digit recognition from digitized documents was lower than that of recognizing individual handwritten digits, reaching 85%. However, this result is higher than the accuracy of recognizing handwritten symbols from complex documents using traditional OCR methods, which ranges from 65% to 80%. Therefore, a hybrid approach combining OpenCV segmentation methods with the developed CNN classifier model is a good solution for improving the efficiency of digitized document processing.

The results of experimental validation showed that integrating the trained CNN classifier model into the developed application enables the proposed methodology to automate the processing of digitized documents containing handwritten digits. This integration resulted in a significant time reduction – by a factor of 7.7. The acceleration was achieved by automating previously slow workflows that required manual data extraction and processing. Successful implementation of this system can significantly reduce the impact of the human factor in routine data extraction tasks, accelerate document processing, and improve the efficiency of government institutions.

However, the developed system addresses only one component of the work of Pension Fund employees – automating the processing of digitized documents to calculate the length of service. The calculated length of service determines the formula used to compute the pension. This highlights the need for further development to expand the areas of automation in the processing of digitized documents within public administration bodies.

Recognition accuracy can be improved by integrating convolutional segmentation models with transformer-based models. This approach achieves handwritten symbol recognition rates of 85–92% in complex documents and highlights promising directions for further enhancement of the proposed methodology. Additionally, automating document workflows using hybrid deep learning techniques in other public sector departments is another promising avenue for future research and development.

REFERENCES

Adnan, K., & Akbar, R. (2019a). An analytical study of information extraction from unstructured and multidimensional big data. *Journal of Big Data*, *6,* Article 91. https://doi.org/10.1186/s40537-019-0254-8

Adnan, K., & Akbar, R. (2019b). Limitations of information extraction methods and techniques for heterogeneous unstructured big data. *International Journal of Engineering Business Management*, *11*. https://doi.org/10.1177/1847979019890771

Ahlawat, S., Choudhary, A., Nayyar, A., Singh, S., & Yoon, B. (2020). Improved handwritten digit recognition using Convolutional Neural Networks (CNN). *Sensors*, *20*(12), 3344. https://doi.org/10.3390/s20123344

Alhamad, H. A., Shehab, M., Shambour, M. K. Y., Abu-Hashem, M. A., Abuthawabeh, A., Al-Aqrabi, H., Daoud, M. S., & Shannaq, F. B. (2024). Handwritten recognition techniques: A comprehensive review. *Symmetry*, *16(6)*, 681. https://doi.org/10.3390/sym16060681

Ali, S., Shaukat, Z., Azeem, M., Sakhawat, Z., Mahmood, T., & Rehman, K. (2019). An efficient and improved scheme for handwritten digit recognition based on convolutional neural network. *SN Applied Sciences*, *1*, Article 1125. https://doi.org/10.1007/s42452-019-1161-5

Alzubaidi, L., Zhang, J., Humaidi, A. J., Al-Dujaili, A., Duan, Y., Al-Shamma, O., Santamaria, J., Fadhel, M. A., Al-Amidie, M., & Farhan, L. (2021). Review of deep learning: Concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data*, *8*, Article 53. https://doi.org/10.1186/s40537-021-00444-8

Aparna, C., & Rajchandar, K. (2024). A robust solution for recognizing accurate handwritten text extraction using quantum convolutional neural network and transformer models. *Computers and Electrical Engineering, 120*, 109794. https://doi.org/10.1016/j.compeleceng.2024.109794

Baviskar, D., Ahirrao, S., Potdar, V., & Kotecha, K. (2021). Efficient automated processing of the unstructured documents using artificial intelligence: A systematic literature review and future directions. *IEEE Access*, *9*, 72894-72936. https://doi.org/10.1109/ACCESS.2021.3072900

Boiangiu, C. A., Dinu, O. A., Popescu, C., Constantin, N., & Petrescu, C. (2020). Voting-based document image skew detection. *Applied Science*, *10*(7), 2236. https://doi.org/10.3390/app10072236

Boliubash, N. (2024). Avtomatyzatsiia obrobky otsyfrovanykh dokumentiv iz rukopysnymy symvolamy u derzhavnomu sektori [Automation of processing of digitized documents in the public sector with handwritten characters]. *Proceedings of the XXI International Scientific Conference: Olviy Forum – 2024: Strategies of the Black Sea Region Countries in the Geopolitical Space, Technical Sciences and Engineering, Mykolaiv, Ukrainian*, 36-40. https://dspace.chmnu.edu.ua/jspui/bitstream/123456789/2349/1/ОФ-2024.%20Технічні%20науки%20та%20інженерія.pdf

Butenko, E. A., Zadorozhnyj, A. M., Lyubovinkina, N. J., & Potemkina, S. V. (2023). Sistema iskusstvennogo intelekta dlya klassifikacii dokumentov slozhnoj struktury [Artificial intelligence system for classifying complex structure documents]. *System Analysis in Science and Education, 1,* 7-12. https://sanse.ru/index.php/sanse/article/view/567/517

Clausner, C., Antonacopoulos, A., & Pletschacher, S. (2020). Efficient and effective OCR engine training. *International Journal Document Analysis and Recognition*, *23*(1), 73-88. https://doi.org/10.1007/s10032-019-00347-8

Cui, H., & Bai, J. (2019). A new hyperparameters optimization method for convolutional neural networks. *Pattern Recognition Letters*, *125*, 828-834. https://doi.org/10.1016/j.patrec.2019.02.009

Eken, S., Menhour, H., & Köksal, K. (2019). DoCA: A content-based automatic classification system over digital documents. *IEEE Access*, *7*, 97996-98004. https://doi.org/10.1109/ACCESS.2019.2930339

Gari, A., Khaissidi, G., Mrabti, M., Chenouni, D., & Yacoubi, M. E. (2017). Skew detection and correction based on Hough transform and Harris corners. *Proceedings of the International Conference on Wireless Technologies, Embedded and Intelligent Systems, Fez, Morocco*, 1-4. https://doi.org/10.1109/WITS.2017.7934619

Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *Proceedings of the 32nd International Conference on International Conference on Machine Learning*, *37*, 448-456. http://proceedings.mlr.press/v37/ioffe15.pdf

Jipeng, T., Kumar, G. H., & Chethan, H. K. (2011). Skew correction for Chinese character using Hough transform. *International Journal of Advanced Computer Science and Applications*, *Special Issue on Image Processing and Analysis*, 45-48. https://doi.org/10.14569/SpecialIssue.2011.010106

Johnson, J., Gupta, A., & Fei-Fei, L. (2018). Image generation from scene graphs. *Proceedings of the Conference on Computer Vision and Pattern Recognition, Salt Lake, UT, USA,* 1219-1228. https://doi.org/10.1109/CVPR.2018.00133

Kanya, M., & Ravi, T. (2016). Named entity recognition from biomedical text – An information extraction task. *ICTACT Journal on Soft Computing*, *6*(4), 1303–1307.

Kaur, A., Baghla, S., & Kumar, S. (2015). Study of various character segmentation techniques for handwritten off-line cursive words: A review. *International Journal of Advances in Science Engineering and Technology*, *3*(3), 154-158. http://www.iraj.in/journal/journal_file/journal_pdf/6-162-1440573382154-158.pdf

Khanday, O. M., Dadvandipour, S., & Lone, M. A. (2021). Effect of filter sizes on image classification in CNN: a case study on CFIR10 and Fashion-MNIST datasets. *IAES International Journal of Artificial Intelligence. 10*(4), 872-878. https://doi.org/10.11591/ijai.v10.i4.pp872-878

Laubrock, J., & Dunst, A. (2020). Computational approaches to comics analysis. *Topics in Cognitive Science*, *12*(1), 274-310. https://doi.org/10.1111/tops.12476

Liu, W., Zhang, Y., & Wan, B. (2016). Unstructured document recognition on business invoice. *Machine Learning, Stanford iTunes University, Stanford, CA, USA, Technical Report.* https://cs229.stanford.edu/proj2016/report/LiuWanZhang-UnstructuredDocumentRecognitionOnBusinessInvoice-report.pdf

Mahadevkar, S. V., Patil, S., Kotecha, K., Soong, L. W., & Choudhury, T. (2024). Exploring AI-driven approaches for unstructured document analysis and future horizons. *Journal of Big Data*, *11*, Article 92. https://doi.org/10.1186/s40537-024-00948-z

Mehul, G., Ankita, P., Namrata, D., Rahul, G., & Sheth, S. (2014). Text-based image segmentation methodology. *Procedia Technology*, *14*, 465-472. https://doi.org/10.1016/j.protcy.2014.08.059

Memon, J., Sami, M., Khan, R. A., & Uddin, M. (2020). Handwritten Optical Character Recognition (OCR): A comprehensive systematic literature review (SLR). *IEEE Access*, *8*, 142642-142668. https://doi.org/10.1109/ACCESS.2020.3012542

Preethi, P., Mamatha, H. R., & Viswanath, H. (2021). Study of using hybrid deep neural networks in character extraction from images containing text. *Trends in Computer Science and Information Technology, 6*(2), 045-052. https://doi.org/10.17352/tcsit.000039

Prum, S. (2017). Text-zone detection and rectification in document images captured by smartphone. *Proceedings of the First EAI International Conference on Computer Science and Engineering, Penang, Malaysia.* https://doi.org/10.4108/eai.27-2-2017.152342

Raja, H., Gupta, A., & Miri, R. (2021). Recognition of automated handwritten digits on document images making use of machine learning techniques. *European Journal of Engineering and Technology Research*, *6*(4), 37-44. https://doi.org/10.24018/ejeng.2021.6.4.2460

Reul, C., Christ, D., Hartelt, A., Balbach, N., Wehner, M., Springmann, U., Wick, C., Grundig, C., Büttner, A., & Puppe, F. (2019). OCR4all – An open-source tool providing a (semi-)automatic OCR workflow for historical printings. *Applied Sciences*, *9*(22), 4853. https://doi.org/10.3390/app9224853

Sahare, P., & Dhok, S. B. (2018). Multilingual character segmentation and recognition schemes for Indian document images. *IEEE Access*, *6*, 10603-10617. https://doi.org/10.1109/ACCESS.2018.2795104

Sakila, A., & Vijayarani, D. S. (2017). Skew detection and correction in the document image. *International Journal of Innovative Research in Science, Engineering and Technology*, *6*(8), 17457-17465. https://doi.org/10.15680/IJIRSET.2017.0608274

Santurkar, S., Tsipras, D., Ilyas, A., & Madry, A. (2019). *How does batch normalization help optimization*. PsyArXiv. https://doi.org/10.48550/arXiv.1805.11604

Saqib, N., Haque, K. F., Yanambaka, V. P., & Abdelgawad, A. (2022). Convolutional-neural-network-based handwritten character recognition: An approach with massive multisource data. *Algorithms*, *15*(4), 129. https://doi.org/10.3390/a15040129

Saritha, S. J., Teja, K. R. G. D., Kumar, G. H., & Sharief, S. J. (2020). Handwritten text detection using OpenCV and CNN. *International Journal of Engineering Research & Technology*, *9*(4), 749-753. https://doi.org/10.17577/IJERTV9IS040641

Singh, P., Varadarajan, S., Singh, A. N., & Srivastava, M. M. (2020). Multi-domain document layout understanding using few-shot object detection. In A. Campilho, F. Karray, & Z. Wang (Eds.), *Image analysis and recognition* (pp. 89-99) Springer.

Proceedings of the 17th International Conference, ICIAR, Póvoa de Varzim, Portugal. Image Analysis and Recognition, Springer, 89-99. https://doi.org/10.1007/978-3-030-50516-5_8

Subudhi, R. K., & Sahu, B. (2014). A novel noise reduction method for OCR system. *International Journal of Computer Science and Technology*, *5*(2), 82-86. https://ijcst.com/vol5/spl2/ec1145.pdf

Sueiras, J., Ruiz, V., Sanchez, A., & Velez, J. F. (2018). Offline continuous handwriting recognition using sequence to sequence neural networks. *Neurocomputing*, *289*, 119-128. https://doi.org/10.1016/j.neucom.2018.02.008

Tabik, S., Alvear-Sandoval, R. F., Ruiz, M. M., Sancho-Gómez, J. L., Figueiras-Vidal, A. R., & Herrera, F. (2020). MNIST-NET10: A heterogeneous deep networks fusion based on the degree of certainty to reach 0.1% error rate. Ensembles overview and proposal. *Information Fusion*, *62*, 73-80. https://doi.org/10.1016/j.inffus.2020.04.002

Tüselmann, O., & Fink, G. A. (2024). Neural models for semantic analysis of handwritten document images. *International Journal on Document Analysis and Recognition, 27*, 245-263. https://doi.org/10.1007/s10032-024-00477-8

Uchida, S., Barney, E., & Eglin, V. (Eds.). (2022). Document analysis systems. *Proceedings of the 15th IAPR International Workshop, La Rochelle, France.* Springer. https://doi.org/10.1007/978-3-031-06555-2

Xue, W., Li, Q., & Xue, Q. (2020). Text detection and recognition for images of medical laboratory reports with a deep learning approach. *IEEE Access*, *8*, 407-416. https://doi.org/10.1109/ACCESS.2019.2961964

Ye, Q., & Doermann, D. (2015). Text detection and recognition in imagery: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *37*(7), 1480-1500. https://doi.org/10.1109/TPAMI.2014.2366765

Ye, Y., Zhu, S., Wang, J., Du, Q., Yang, Y., Tu, D., Wang, L., & Luo, J. (2018). A unified scheme of text localization and structured data extraction for joint OCR and data mining. *Proceedings of the IEEE International Conference on Big Data, Seattle, WA, USA,* 2373-2382. https://doi.org/10.1109/BigData.2018.8622129

Yevtushenko, O. (2024). Digitalization: A tool for modernization of public administration in Ukraine. *Public Administration and Regional Development, 26*, 1158-1176. https://doi.org/10.34132/pard2024.26.03

Zhao, X., Wang, L., Zhang, Y., Han, X., Deveci, M., & Parmar, M. (2024). A review of convolutional neural networks in computer vision. *Artificial Intelligence Review*, *57*, Article 99. https://doi.org/10.1007/s10462-024-10721-6

AUTHORS

Nadiia Boliubash holds a PhD in Education and is an Associate Professor at the Department of Intelligent Information Systems, Petro Mohyla Black Sea National University, Ukraine. She is the author of more than 60 research publications that reflect her scientific interests, including data mining, neural networks, artificial intelligence, information technology in education, and learning analytics.

Oleksandr Yevtushenko is a Doctor of Political Science and Professor who serves as the Head of the Department of Public Administration and Management at Petro Mohyla Black Sea National University, Ukraine. His research interests include public authority, civil society, territorial institutions of state power and local self-government, the digital transformation of public administration, the implementation of state policy in the socio-humanitarian sphere, and the psychological dimensions of leadership and governance. He is the author of more than 220 publications.