https://doi.org/10.31891/2219-9365-2024-80-41
UDC 004.93.1

GUANXIANG Xia
Vinnytsia National Technical University
Hunan Mass Media Vocational and Technical College
xiaguanxiang@163.com
KOVTUN Viacheslav
Vinnytsia National Technical University
https://orcid.org/0000-0002-7624-7072
kovtun_v_v@vntu.edu.ua

# METHODS FOR IMPLEMENTATION OF STRING OBJECT RECOGNITION RESULTS IN REAL-TIME VIDEO STREAMS

*This study addresses the challenge of string object recognition in real-time video streams, particularly focusing on improving recognition accuracy under dynamic conditions with distortions such as defocusing, glare, and motion artefacts. A novel algorithm is proposed that integrates recognition results from multiple video frames using extended result models, considering alternative classification options for each object. The algorithm leverages dynamic programming and advanced metrics, such as the Generalized Levenshtein Distance, to aggregate recognition outcomes effectively. Experimental validation on the MIDV-500 dataset demonstrates the proposed method's superiority over traditional approaches, including the ROVER method, in reducing recognition errors across various text fields. The findings highlight the algorithm's robustness and scalability for applications in document digitization, automated data extraction, and mobile-based text recognition. Future research directions include optimizing computational efficiency, expanding to multilingual recognition tasks, and validating performance on diverse datasets to ensure generalizability for real-world applications.*

*Keywords: real-time video stream, string object recognition, text recognition, dynamic programming, video-based OCR, recognition result integration, distortion mitigation.*

ГУАНСЯНГ Ся
Вінницький національний технічний університет
Хунанський коледж масових медіа
КОВТУН В’ячеслав
Вінницький національний технічний університет

# МЕТОДИ РОЗПІЗНАВАННЯ РЯДКОВИХ ОБ’ЄКТІВ У ВІДЕО ПОТОКАХ РЕАЛЬНОГО ЧАСУ

*У статті досліджується проблема підвищення точності розпізнавання рядкових об’єктів у відео потоках реального часу. Досліджуваний процес характеризується динамічністю та присутністю в аналізованому відео потоці таких типів спотворень, як дефокусування, відблиски та артефакти руху. Представлено новий метод, який інтегрує результати розпізнавання з кількох відеокадрів, враховуючи альтернативні варіанти класифікації для кожного рядкового об’єкта. Методпоєднує принципи динамічного програмування та сучасні метрики, такі як узагальнена відстань Левенштейна, для ефективного агрегування результатів розпізнавання. Експериментальна перевірка на наборі даних MIDV-500 демонструє перевагу запропонованого методу над традиційними підходами, зокрема методом ROVER, у зниженні помилок розпізнавання різних типів рядкових об’єктів. Результати підкреслюють надійність і масштабованість методу для застосувань у сфері оцифровки документів, автоматизованого екстрагування даних та розпізнавання тексту мобільними пристроями. Майбутні напрями досліджень будуть спрямовані на оптимізацію обчислювальної ефективності представленого методу, його адаптацію до розпізнавання рядкових об’єктів, утворених різними мовами, та перевірку продуктивності на наборах даних, які відтворюють реалістичні сценарії застовуаання відповідних систем розпізнавання.*

*Ключові слова: відео потік реального часу, розпізнавання рядкових об’єктів, розпізнавання тексту, динамічне програмування, відео-оптичне розпізнавання символів, інтеграція результатів розпізнавання, зменшення впливу спотворень.*

## THE PROBLEM STATEMENT IN GENERAL FORM AND ITS CONNECTION WITH IMPORTANT SCIENTIFIC OR PRACTICAL TASKS

The recognition of string objects such as text paragraphs, text lines, and document fields presents significant challenges, particularly when the source images are obtained from mobile device cameras [1, 2]. These challenges arise due to various image distortions [3, 4], including defocusing, blurring, glare on reflective surfaces, and insufficient resolution. Such factors often degrade the performance of character recognition algorithms, making it difficult to achieve the desired level of accuracy.

In real-time video streams, these challenges are further exacerbated by the dynamic nature of video content, where changes in lighting, object orientation, and motion can introduce additional complexity [5]. Despite these difficulties, video streams offer a unique advantage by providing a continuous sequence of frames. This allows for the repeated recognition of the same object across multiple frames, thereby improving the final recognition accuracy through redundancy.

However, relying on a single best result from a video stream is often insufficient, as certain frames may fail

to capture the object fully or clearly [6]. This limitation necessitates the development of advanced methods for aggregating recognition results from multiple frames. By leveraging information from diverse frames, it is possible to mitigate the impact of distortions and improve the reliability of recognition outcomes.

Addressing this problem is of critical scientific and practical significance. It directly contributes to the advancement of real-time recognition systems, enabling applications in document digitization, automated data extraction, mobile-based text recognition, and other fields where efficient and accurate recognition of string objects is essential.

## THE PROBLEM STATEMENT AND REVIEW OF RECENT RESEARCH

Consider the model for the recognition result of a single object. Let the image $\mathrm{P}$ of an object $k$ be classified into one of $N$ classes from a set $K=\{k_1,k_2,\ldots,k_N\}$ using a classification module $f$. In the classical formulation, the result of classification is one of the classes $f(\mathrm{P})=k_f$, where $k_f\in K$, and the task of recognizing a single object is to maximize the posterior probability that the class $k_f$ matches the true value $k$. In a more general formulation, the classification module $\hat{f}$ assigns a set of pairs $\hat{f}(I)=\{(k_1,\phi_1),(k_2,\phi_2),\ldots,(k_N,\phi_N)\}$ to the input image, where $\phi_i$ is the membership score of the object to class $k_i$. The final recognition result is the class corresponding to the maximum membership score:

$$f(\mathrm{P})=\arg\max\left\{\hat{f}(\mathrm{P})\right\}\in\left\{k_f\left|\left((k_f,\phi_f)\in\hat{f}(\mathrm{P})\right)\wedge\left(\phi_f=\max_{(k,\phi)\in\hat{f}(\mathrm{P})}\phi\right)\right.\right\}. \quad (1)$$

In the case where there are multiple pairs $(k_{f1},\phi_f),(k_{f2},\phi_f),\ldots$ with the same maximum membership score, one of the classes is selected as the answer according to the adopted convention (for example, the class with the smallest index in the set $K$). The recognition result model for a single object (1) is a variant of the Algorithm for Calculating Scores (ACS) [7]. It is also the most widely used model in optical image recognition methods using convolutional neural networks [8, 9].

To define the recognition result of a string object, it is necessary to introduce the concept of an empty class $\eta$, representing the absence of a single object. The extended result of classifying a single object is considered to be a mapping $\alpha: K\cup\{\eta\}\to[0,1]$ from the set of classes, augmented with the label of the empty class $\eta$, to the set of membership scores. Each membership score is a real number between 0 and 1, and the sum of the membership scores is equal to one. Thus, the set of all possible recognition results for a single object $K$ is defined as:

$$\hat{K}\overset{def}{=}\left\{\alpha\in[0,1]^{K\cup\{\eta\}}\left|\sum_{k\in K\cup\{\eta\}}\alpha(k)=1\right.\right\}. \quad (2)$$

On the set of all possible recognition results for a single object $K$, a metric can be defined as follows:

$$l_{\hat{K}}(\alpha,\beta)\overset{def}{=}\frac{1}{2}\sum_{k\in K\cup\{\eta\}}\left|\alpha(k)-\beta(k)\right|,\ \forall\alpha,\beta\in\hat{K}. \quad (3)$$

It is easy to verify that the function $l_{\hat{K}}(\alpha,\beta)$ possesses the properties of a valid metric:

1. $l_{\hat{K}}(\alpha,\beta)=0\Leftrightarrow\forall k\in K\cup\{\eta\}:\ \alpha(k)=\beta(k)\Leftrightarrow\alpha=\beta$, thus the identity axiom is satisfied.

2. $\forall k\in K\cup\{\eta\}:\ \left|\alpha(k)-\beta(k)\right|=\left|\beta(k)-\alpha(k)\right|\Rightarrow l_{\hat{K}}(\alpha,\beta)=l_{\hat{K}}(\beta,\alpha)$, thus the symmetry axiom is satisfied.

3. $\forall i,j\in\square:\ \left|i+j\right|\le\left|i\right|+\left|j\right|\Rightarrow\forall k\in K\cup\{\eta\}:$ $\left|\alpha(k)-\gamma(k)\right|\le\left|\alpha(k)-\gamma(k)\right|+\left|\gamma(k)-\beta(k)\right|\Rightarrow l_{\hat{K}}(\alpha,\beta)\le l_{\hat{K}}(\alpha,\gamma)+l_{\hat{K}}(\gamma,\beta)$, thus, the triangle inequality is also satisfied.

It is worth noting that the metric $l_{\hat{K}}(\alpha,\beta)$ corresponds to the Manhattan metric in the vector space over the ordered set $K \cup \{\eta\}$. Since for $\alpha$ and $\beta$, the sum of the values across all $k \in K \cup \{\eta\}$ equals one, the set of values for the function $l_{\hat{K}}(\alpha,\beta)$ forms a segment [0, 1]. We denote the "empty result" as:

$$\eta \overset{def}{=} \{(\eta,1),(k_1,0),(k_2,0),\ldots,(k_N,0)\}. \quad (4)$$

The result $\mathrm{P}$ of recognition of a string object will be called a string over the set $K \setminus \{\hat{\eta}\}$, i.e., an element $\mathrm{P} \in P$, where $P \overset{def}{=} \left(\hat{K} \setminus \{\hat{\eta}\}\right)^*$. The string $\mathrm{P}$ represents a sequence of recognition results of individual objects $\mathrm{P} = \rho_1\rho_2\ldots\rho_m$, where $\rho_i \in \hat{K} \setminus \{\hat{\eta}\}$, and the length of the string $|\mathrm{P}| = m$ is called the number of elements in this sequence. The notation $\mathrm{P}_{i\ldots j}$ refers to a substring of the string $\mathrm{P}$, including elements $\rho_i\rho_{i+1}\ldots\rho_{j-1}\rho_j$ for $1 \le i \le j \le m$. When $i > j$, the substring $\mathrm{P}_{i\ldots j}$ corresponds to an empty string $\hat{\eta}$ of zero length.

The recognition of string objects in real-time video streams has garnered considerable attention due to its broad applicability in fields such as document digitization, automated data extraction, and mobile-based text recognition [10]. Several studies have contributed to advancing this area, emphasizing the challenges and innovations in handling dynamic and diverse input conditions.

Recent advancements in deep learning have significantly improved object detection and recognition capabilities. Study [9] provided a comprehensive survey of deep learning techniques for object detection, emphasizing their potential in dynamic environments such as autonomous driving systems. Their findings highlight the robustness of convolutional neural networks (CNNs) in handling varying lighting conditions, object orientations, and motion-induced blurring.

In video-based recognition tasks, [11] proposed an objective video quality assessment method tailored for object recognition tasks. Their work underscores the importance of high-quality video frames in enhancing recognition accuracy, particularly in scenarios where video content is affected by motion artefacts and resolution inconsistencies.

For tasks specific to text recognition, [12] explored methods to increase real-time object recognition accuracy on mobile platforms. Their study emphasized the role of preprocessing techniques such as image normalization and text area segmentation in mitigating the effects of glare, defocusing, and other distortions commonly encountered in mobile-captured images.

Machine learning models incorporating Optical Character Recognition (OCR) techniques have also been a focal point of recent research. Research [13] performed a comparative analysis of Google Vision OCR and Tesseract, demonstrating the trade-offs between accuracy and computational efficiency. Similarly, [14] implemented OCR for Javanese script, showcasing the adaptability of OCR systems to diverse textual formats and languages.

The integration of dynamic programming methods into recognition pipelines has shown promising results. Study [15] introduced a normalized Levenshtein distance metric, which has been widely adopted for string-matching tasks. This metric provides a foundation for aligning recognition results across multiple frames in video streams, enabling improved aggregation of recognition outcomes.

The ROVER (Recognizer Output Voting Error Reduction) method, as described in [16], offers a robust framework for combining recognition results from multiple classifiers. This approach has been successfully adapted to optical recognition tasks, including printed and handwritten text, by leveraging a voting mechanism to resolve ambiguities in recognition outputs.

In addressing the challenges of string object recognition in real-time video streams, study [17] proposed an anytime algorithm with an adaptive variable-step-size mechanism. Their method, designed for the path planning of UAVs, demonstrates the broader applicability of such algorithms in scenarios requiring iterative optimization under dynamic conditions.

Despite these advancements, existing methods often fall short of effectively integrating recognition results across frames to leverage redundancy in video streams. The present study builds upon these foundations, proposing an extended model of string object recognition that considers alternative classification options for individual objects. This approach seeks to bridge gaps in current methodologies, providing a more robust and accurate framework for real-time video-based text recognition.

## FORMULATION OF THE ARTICLE'S OBJECTIVES

The purpose of this article is to develop and validate an advanced algorithm for integrating string object recognition results in real-time video streams, leveraging alternative classification options and extended result models to enhance accuracy and reliability in dynamic conditions. To design a recognition result model for string

objects that incorporates alternative classification options for individual elements, ensuring robustness in handling video stream distortions such as defocusing, glare, and motion artefacts. To create an algorithm for integrating recognition results of string objects based on the proposed model, ensuring compatibility with dynamic programming methods and existing metrics such as the Generalized Levenshtein Distance. To conduct experimental studies on open datasets like MIDV-500, comparing the proposed algorithm's performance with traditional methods such as ROVER, and evaluating its effectiveness in reducing recognition errors across various text field types and video sequence lengths.

**PRESENTATION OF THE MAIN MATERIAL**

Let us introduce the concept of an elementary edit operation $D$ as a pair $(\alpha,\beta)\neq(\hat{\eta},\hat{\eta})$, where $\alpha,\beta\in\hat{K}$. An edit operation $D=(\alpha,\beta)$, applied to the string $\mathrm{P}$, corresponds to:

1. If $\beta\neq\eta$: replacing element $\rho_i=\alpha$ in the string $\mathrm{P}$ with element $\beta$, if $\beta\neq\eta$;

2. If $\beta=\hat{\eta}$: deleting element $\rho_i=\alpha$ from the string $\mathrm{P}$;

3. If $\alpha=\hat{\eta}$: inserting an element $\beta$ into the string $\mathrm{P}$.

Let us consider two arbitrary strings $\mathrm{P},\Theta\in P$ of finite length. An editorial prescription is defined as a sequence of elementary editorial changes $D_{\mathrm{P},\Theta}=D_1D_2\ldots D_L$ that transforms a string $\mathrm{P}$ into a string $\Theta$. The weight of an editorial prescription is considered to be the sum of the distances (in terms of metric $l_{\hat{K}}$) between pairs of objects involved in the elementary editorial changes $D_i=(\alpha_i,\beta_i)$ of the prescription $D_{\mathrm{P},\Theta}$:

$$\omega\left(D_{\mathrm{P},\Theta}\right)\overset{def}{=}\sum_{i=1}^{L}l_{\hat{K}}\left(\alpha_i,\beta_i\right). \tag{5}$$

A metric on the set of recognition results for string objects $P$ is defined as the minimal weight of an editorial prescription that transforms one string into another:

$$l_P\left(\mathrm{P},\Theta\right)=\min\left\{\omega\left(D_{\mathrm{P},\Theta}\right)\right\}. \tag{6}$$

The metric (6) can be considered as one of the implementations of the Generalized Levenshtein Distance [13] and possesses the properties of a true metric, provided that the metric (3) also satisfies these properties [13].

To calculate the distance between two recognition results of string objects $l_P\left(\mathrm{P},\Theta\right)$, the following recursive scheme can be used. Let $a(i,j)\overset{def}{=}l_P\left(\mathrm{P}_{1\ldots i},\Theta_{1\ldots j}\right)$ represent the distance between the prefixes of strings $\mathrm{P}$ and $\Theta$, which have lengths $i$ and $j$, respectively. Then

$$a(0,0)=0,\ a(i,0)=\sum_{h=1}^{i}l_{\hat{K}}\left(\rho_h,\hat{\eta}\right),\ a(0,j)=\sum_{h=1}^{j}l_{\hat{K}}\left(\hat{\eta},\theta_h\right),$$
$$a(i,j)=\min\begin{Bmatrix}l_{\hat{K}}\left(\rho_i,\hat{\eta}\right)+a(i-1,j),\\ l_{\hat{K}}\left(\hat{\eta},\theta_j\right)+a(i,j-1),\\ l_{\hat{K}}\left(\rho_i,\theta_j\right)+a(i-1,j-1)\end{Bmatrix}, \tag{7}$$

and the sought value of the metric $l_P\left(\mathrm{P},\Theta\right)$ corresponds to the value $a\left(|\mathrm{P}|,|\Theta|\right)$.

It is worth noting that the maximum possible value of the metric $l_P\left(\mathrm{P},\Theta\right)$ is the maximum length of the strings $\mathrm{P}$ and $\Theta$ (when using (3) as the metric on the set of recognition results for single objects). Since $l_P$ it is a special case of the Generalized Levenshtein Distance, it is possible to construct a normalized version of this metric while preserving the axioms of identity, symmetry, and the triangle inequality [13]:

$$\tilde{l}_P\left(\mathrm{P},\Theta\right)def\ \frac{2l_P\left(\mathrm{P},\Theta\right)}{\sigma\left(|\mathrm{P}|+|\Theta|\right)+l_P\left(\mathrm{P},\Theta\right)}, \tag{8}$$

where $\sigma$ is the maximum possible weight of an elementary insertion or deletion. For the case of the metric (3): $\sigma=\max\left\{l_{\hat{K}}\left(\alpha,\hat{\eta}\right),l_{\hat{K}}\left(\hat{\eta},\beta\right);\alpha,\beta\in K\right\}=1$.

Let us consider the task of recognizing a string object in a video sequence. The input to the system is a sequence of images $\mathrm{P}_1, \mathrm{P}_2, \ldots, \mathrm{P}_M$ of the string object $w \in K^*$. Using the module $\hat{F}$ for recognizing the string object in a single image, each image is assigned a recognition result $\hat{F}\left(\mathrm{P}_i\right) \in P$. Within the framework of the considered model, we assume that in the initial recognition result of the string object, the membership estimates corresponding to the empty class $\eta$ are equal to zero:

$$\hat{F}\left(\mathrm{P}_i\right) = P_i,\ P_i \in P,\ \mathrm{P}_i = \rho_1^i \rho_2^i \ldots \rho_{m_i}^i,\ \rho_j^i(\eta) = 0\ \ \forall j \in \{1, \ldots, m_i\}. \tag{9}$$

The task consists of combining the results $\mathrm{P}_1, \mathrm{P}_2, \ldots, \mathrm{P}_M$ with certain weights $\omega_1, \omega_2, \ldots, \omega_M$ into a unified result $\mathrm{P} \in P$, minimizing the distance to the true value $w$ according to a given metric. Since $\mathrm{P} \in P$ it is a string over the set $\hat{K} \setminus \{\eta\}$, and $w$ is a string over the set of classes $K$, additional conversion is required to determine the distance between them. The most natural approach is to transform the true value $w$ into a string $\hat{w} \in P$:

$$\begin{gathered} w = w_1 w_2 \ldots w_{m_w},\ w_j \in K,\ \hat{w} = \hat{w}_1 \hat{w}_2 \ldots \hat{w}_{m_w},\ \hat{w}_j \in \hat{K} \setminus \{\hat{\eta}\}, \\ \hat{w}_j \overset{def}{=} \{(\eta, 0), (k_1, 0), (k_2, 0), \ldots, (w_j, 1), \ldots, (k_N, 0)\}, \end{gathered} \tag{10}$$

and use the distance (6) or its normalized variant (8) as the distance from the integrated result $\mathrm{P}$ to the true value $w$.

However, from a practical application perspective, it is also important to obtain the final recognition result of the string object (analogous to the final result (1) for a single object). The following two-step procedure can be used to obtain the final result:

1. At the first step, each component $\rho_j \in \hat{K} \setminus \{\hat{\eta}\}$ of the integrated result $\mathrm{P} = \rho_1 \rho_2 \ldots \rho_{m_\mathrm{P}}$ is assigned either the class $k_{\rho_j} \in K$ with the maximum membership estimate $\rho_j\left(k_{\rho_j}\right)$, or the empty class $\eta$, if its estimate $\rho_j(\eta)$ exceeds a certain threshold $\lambda$:

$$\bar{\rho}_j = \begin{cases} \arg\max \rho_j(k)\, \forall \rho_j(\eta) < \lambda, \\ \eta \forall \rho_j(\eta) \geq \lambda. \end{cases} \tag{11}$$

2. In the second step, all components $\bar{\rho}_j = \eta$ are removed from the resulting string $\bar{\rho}_1 \bar{\rho}_2 \ldots \bar{\rho}_{m_\mathrm{P}}$. The resulting string $\overline{\mathrm{P}}_\lambda \in K^*$ can be used as the final recognition result of the string object.

As the distance from the integrated result $\mathrm{P}$ to the true value $w$, the Levenshtein distance $levenshtein\left(\overline{\mathrm{P}}_\lambda, w\right)$ [13] or its normalized variant:

$$l_L\left(\overline{\mathrm{P}}_\lambda, w\right) = \frac{2 levenshtein\left(\overline{\mathrm{P}}_\lambda, w\right)}{\left|\overline{\mathrm{P}}_\lambda\right| + |w| + levenshtein\left(\overline{\mathrm{P}}_\lambda, w\right)}, \tag{12}$$

can now be used.

The approach presented in [16] has also been applied to combining multiple classifiers in optical recognition tasks for printed and handwritten texts. The approach described in [16], known as Recognizer Output Voting Error Reduction (ROVER), employs a two-module scheme:

In the first stage, the alignment module converts all input string objects into strings of equal length by optimally inserting the empty class $\eta$.

In the second stage, the voting module selects a class for each component of the resulting string based on a linear combination of occurrence frequency and confidence scores generated by the recognition module.

The model of a single-string recognition result in the ROVER approach [16] is represented as a pair consisting of a string over the set of recognition classes for individual objects and the confidence score of the recognition module, i.e., an object from the set $K^* \times$ □ .

To construct an algorithm for integrating string object recognition results with an extended model of a single recognition result, let us consider the problem formulation of string alignment of the form (9).

Let $\mathrm{P}_1, \mathrm{P}_2, \ldots, P_M$ be the set of $M$ strings, where $\mathrm{P}_i \in P$ and $\left|\mathrm{P}_i\right| = m_i > 0$ are given:

$$\mathrm{P}_1 = \rho_1^1\rho_2^1\ldots\rho_{m_1}^1,\ \mathrm{P}_2 = \rho_1^2\rho_2^2\ldots\rho_{m_2}^2,\ \mathrm{P}_M = \rho_1^M\rho_2^M\ldots\rho_{m_M}^M. \quad (13)$$

The alignment of a given set of strings will be understood as an $align$ function: $\{1,\ldots,M\}\times\left\{1,\ldots,\max_{i=1}^{M} m_i\right\}\to\left\{1,\ldots,\sum_{i=1}^{M} m_i\right\}$. The function $align(i,j)$ specifies the index of the output "integrated" string component, into which the components $\rho_j^i$ contribute. For each input string, the values of the $align$ function for individual string components are different and maintain their order: $\forall i\in\{1,\ldots,M\}$, $\forall j\in\{1,\ldots,m_i-1\}: align(i,j)<align(i,j+1)$.

We also introduce a $match$ function: $\{1,\ldots,M\}\times\left\{1,\ldots,\sum_{i=1}^{M} m_i\right\}\to K$, defined as follows:

$$match(i,c)\overset{def}{=}\begin{cases}\rho_j^i\,\forall align(i,j)=c,\\ \eta\,\forall\exists j: align(i,j)=c.\end{cases} \quad (14)$$

The alignment problem consists of finding the $align$ function such that the penalty functional

$$\sum_c\sum_{i_1<i_2} l_{\hat{K}}\left(match(i_1,c),\left(match(i_2,c),\right)\right)\to\min \quad (15)$$

is minimized, reflecting the total pairwise distance between the recognition results of individual objects that contribute to the same components of the integrated result.

To generalize the voting module, which selects a class for each component of the resulting string, we introduce a family of functions for combining the recognition results of individual objects $y^{(M)}$:

$$y^{(M)}:\hat{K}^M\times\left(\square_0^+\right)^M\to\hat{K}\setminus\{\hat{\eta}\}. \quad (16)$$

The function $y^{(M)}$ takes as input $M$ recognition results of individual objects $\alpha_1,\alpha_2,\ldots,\alpha_M$ such that $\exists i:\alpha_i\neq\hat{\eta}$, and a set of associated non-negative weights $\omega_1,\omega_2,\ldots,\omega_M$, reflecting the significance of the result, such that $\sum_{i=1}^{M}\omega_i>0$.

Then, the function $Y^{(M)}$ for integrating the recognition results of string objects takes the form

$$Y^{(M)}\left(\mathrm{P}_1,\mathrm{P}_2,\ldots,\mathrm{P}_m,\omega_1,\omega_2,\ldots,\omega_M\right)=y_1^{(M)},y_2^{(M)},\ldots,y_{m_Y}^{(M)}. \quad (17)$$

where $m_Y=\max align(i,j)$, and each component of the resulting string is calculated using the combination function (16) and in accordance with the alignment result (14):

$$y_j^{(M)}=y^{(M)}\left(match(1,j),match(2,j),\ldots,match(M,j),\omega_1,\omega_2,\ldots,\omega_M\right). \quad (18)$$

In the general case, the exact solution to the problem (15) involves calculating a dynamic programming scheme (analogous to the scheme for calculating the Generalized Levenshtein Distance (7) with a computational cost that exponentially depends on the number of input strings $M$ (since the results of the string alignment subproblems $\mathrm{P}_{11\ldots i_1},\mathrm{P}_{21\ldots i_2},\ldots,\mathrm{P}_{M1\ldots i_M}$ must be used for all tuples $(i_1,i_2,\ldots,i_M)\in\{1,\ldots,m_1\}\times\{1,\ldots,m_2\}\times\ldots\times\{1,\ldots,m_M\}$. Heuristic algorithms for shortest path search, such as $A^*$-search [17], can also be applied when calculating this scheme. The next section will present an algorithm for integrating the recognition results of string objects, with the alignment functional approximated using the method employed in the ROVER approach [16].

When calculating the integrated result of the recognition of a string object, a set of intermediate integrated results $Y^{(1)}(\mathrm{P}_1,\omega_1),\ldots,Y^{(i-1)}(\mathrm{P}_1,\ldots,\mathrm{P}_{i-1},\omega_1,\ldots,\omega_{i-1})$ is generated, where the result $Y^{(i-1)}$ is used to solve the alignment task at step $i$. In the first step of the algorithm:

$$Y^{(1)}\left(\mathrm{P}_1,\omega_1\right)=\mathrm{P}_1. \quad (19)$$

At each subsequent $i$-th step of the algorithm, the optimal alignment of the strings $\mathrm{P}_i$ and $Y^{(i-1)}\left(\mathrm{P}_1,\ldots,\mathrm{P}_{i-1},\omega_1,\ldots,\omega_{i-1}\right)$ is built using a dynamic programming scheme, similar to (7). Let $\gamma\left(d,e\right)\overset{def}{=}l_P\left(\mathrm{P}_{i1\ldots d},Y^{(i-1)}\left(\mathrm{P}_1,\ldots,\mathrm{P}_{i-1},\omega_1,\ldots,\omega_{i-1}\right)_{1\ldots e}\right)$ and $R_r\left(d,e\right)$ are the auxiliary functions for $r\in\left\{1,2,3\right\}$. The calculation of $\gamma\left(d,e\right)$ and $R_r\left(d,e\right)$ is performed according to the following procedure:

$$\gamma\left(0,)\right)=0,\ \gamma\left(d,0\right)=\sum_{c=1}^{d}l_{\hat{K}}\left(\rho_c^i,\hat{\eta}\right),\ \gamma\left(0,e\right)=\sum_{c=1}^{e}l_{\hat{K}}\left(\hat{\eta},y_c^{(i-1)}\right),$$
$$R_1\left(d,e\right)=l_{\hat{K}}\left(\rho_d^i,\hat{\eta}\right)+\gamma\left(d-1,e\right),\ R_2\left(d,e\right)=l_{\hat{K}}\left(\hat{\eta},y_e^{(i-1)}\right)+\gamma\left(d,e-1\right), \tag{20}$$
$$R_3\left(d,e\right)=l_{\hat{K}}\left(\rho_d^i,y_e^{(i-1)}\right)+\gamma\left(d-1,e-1\right),\ \gamma\left(d,e\right)=\min\left\{R_1\left(d,e\right),R_2\left(d,e\right),R_3\left(d,e\right)\right\}.$$

To calculate the integration result $Y^{(i)}\left(\mathrm{P}_1,\ldots,\mathrm{P}_i,\omega_1,\ldots,\omega_i\right)$ at the $i$-th step we introduce two auxiliary functions $\varphi_{\mathrm{P}}:\left\{0,\ldots,m_i+m_{Y_{i-1}}\right\}\to\left\{1,\ldots,m_i\right\}$ and $\varphi_Y:\left\{0,\ldots,m_i+m_{Y_{i-1}}\right\}\to\left\{1,\ldots,m_{Y_{i-1}}\right\}$, whose calculation is performed according to the following recursive procedure:

$$\varphi_{\mathrm{P}}\left(0\right)=m_i,$$
$$\varphi_Y\left(0\right)=m_{Y_{i-1}},\varphi_{\mathrm{P}}\left(c+1\right)=\begin{cases}\varphi_{\mathrm{P}}\left(c\right)\forall R_2\left(\varphi_{\mathrm{P}}\left(c\right),\varphi_Y\left(c\right)\right)\wedge R_1\left(\varphi_{\mathrm{P}}\left(c\right),\varphi_Y\left(c\right)\right)\neq\gamma\left(\varphi_{\mathrm{P}}\left(c\right),\varphi_Y\left(c\right)\right),\\ \varphi_{\mathrm{P}}\left(c\right)+1\forall else,\end{cases} \tag{21}$$
$$\varphi_Y\left(c+1\right)=\begin{cases}\varphi_Y\left(c\right)\forall R_1\left(\varphi_{\mathrm{P}}\left(c\right),\varphi_Y\left(c\right)\right)=\gamma\left(\varphi_{\mathrm{P}}\left(c\right),\varphi_Y\left(c\right)\right),\\ \varphi_Y\left(c\right)+1\forall else.\end{cases}$$

The integrated result at the $i$-th step is calculated as follows:

$$m_{Y_i}=\min\left\{c:\varphi_{\mathrm{P}}\left(c\right)=\varphi_Y\left(c\right)=0\right\},\ Y^{(i)}\left(\mathrm{P}_1,\ldots,\mathrm{P}_i,\omega_1,\ldots,\omega_i\right)=y_1^{(i)}y_2^{(i)}\ldots y_{m_{Y_i}}^{(i)},$$
$$y_c^{(i)}=\begin{cases}y^{(2)}\left(y_{\varphi_Y(\varphi(c))+1}^{(i-1)},\hat{\eta},Z_{i-1},\omega_i\right)\forall\varphi_{\mathrm{P}}\left(\varphi\left(c\right)\right)=\varphi_{\mathrm{P}}\left(\varphi\left(c\right)-1\right),\\ y^{(2)}\left(\hat{\eta},\rho_{\varphi_{\mathrm{P}}(\varphi(c))+1}^{i},Z_{i-1},\omega_i\right)\forall\varphi_Y\left(\varphi\left(c\right)\right)=\varphi_Y\left(\varphi\left(c\right)-1\right),\\ y^{(2)}\left(y_{\varphi_Y(\varphi(c))+1}^{(i-1)},\rho_{\varphi_{\mathrm{P}}(\varphi(c))+1}^{i},Z_{i-1},\omega_i\right)\forall else,\end{cases} \tag{22}$$

where $Z_i\overset{def}{=}\sum_{c=1}^{i}\omega_c$; $\varphi\left(c\right)\overset{def}{=}m_{Y_i}-c+1$ is the auxiliary function, and $y^{(2)}$ is the function for integrating the two recognition results of single objects (16).

It should be noted that within the proposed algorithm, the integration function (16) requires the following property:

$$y^{(M)}\left(\alpha_1,\ldots,\alpha_M,\omega_1,\ldots,\omega_M\right)=$$
$$=y^{(2)}\left(y^{(M-1)}\left(\alpha_1,\ldots,\alpha_{M-1},\omega_1,\ldots,\omega_{M-1}\right),\alpha_M,\omega_1+\ldots+\omega_{M-1},\omega_M\right). \tag{23}$$

In the case where the used function $y$ does not possess the property (23), the alignment procedure remains unchanged, and the integrated result at step $i$ must be calculated for each component of the resulting string using formula (18), after explicitly restoring the functions $align$ and $match$.

Within the framework of this dissertation, it is proposed to use the weighted average as the function $y$, which possesses the property (23):

$$y^{(M)}\left(\alpha_1,\ldots,\alpha_M,\omega_1,\ldots,\omega_M\right)\left(k\right)=\frac{1}{Z_M}\sum_{i=1}^{M}\alpha_i\left(k\right)\omega_i,\ \forall k\in K\cup\left\{\eta\right\}. \tag{24}$$

The experimental study was conducted on the open MIDV-500 dataset [18], which holds 50 different types of identity document videos (with 10 video clips for each certificate; 30 frames per video) with explained ideal locations and content of text areas. Four groups of fields were analyzed: dates recorded with numbers and punctuation marks, Machine-Readable Zone (MRZ) strings, certificate number,and elements of the certificate holder's name written in the Latin alphabet.

Only frames in which the entire document is visible were considered (thus, the video sequences in the examined subset of a dataset had varying sizes, from 1 to 30 frames). To decrease the effects of normalization and offer a clearer representation of the outcomes, every clip was extended in duration to 30 frames by repeating the clip from the beginning (thus, all analyzed clips had an identical duration of 30 frames).

Each area was extracted from the source image using a projective transformation, according to the joint annotation of the ideal boundaries of the document and the coordinates of the text area, with margins increased to 30% of the shortest side of the text area. The size of the extracted text area images corresponded to a resolution of 300 dots per inch. Each extracted text area was recognized using the component of the Tesseract system [13], which is responsible for recognizing a single text string with an extended result model (9).

The normalized Levenshtein distance (12) between the true value and the text string obtained using the procedure (11), was used as the distance between the integrated recognition result of the text area and its true value. All character comparisons were performed regardless of case, and the Latin letter "O" was considered identical to the digit "0".

In the framework of this experimental study, the proposed algorithm, operating within the extended model of the string object recognition result, was compared with an analogue operating within the classical model. For each group of text areas and each video sequence, integration was performed using the ROVER method, where simple text strings formed by the procedure (11), applied to the frame-by-frame recognition results, were used as input data. The threshold $\lambda$ for the empty symbol score (11) was set to 0.6 for both the control ROVER method and the proposed algorithm.

Fig. 1 presents the results of the compared algorithms for the four groups of text areas in the MIDV-500 dataset. It can be noted that for each group of fields, the integration using the proposed algorithm of the full recognition results (i.e., considering alternative recognition options for each character) achieves a lower error value than integration using the ROVER method (which only considers the first alternatives for the recognition of each character), regardless of the length of the sequence of integrated results.

The achieved average values of the distance between the integrated recognition result of the text area and its true value for different lengths of the integrated video sequence prefix are presented in Table 1.

Table 1.

**Achieved distance between the integrated recognition result and the true value without integration using the ROVER method and the proposed algorithm**

| Integration method | Frame number (length of the sequence of integrable results) | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | **3** | **6** | **9** | **12** | **15** | **18** | **21** | **24** | **27** |
| Without integration | 0.136 | 0.154 | 0.160 | 0.157 | 0.168 | 0.159 | 0.165 | 0.166 | 0.150 |
| Integration using the proposed algorithm | 0.115 | 0.089 | 0.078 | 0.071 | 0.066 | 0.065 | 0.066 | 0.066 | 0.064 |
| Integration using the ROVER method | 0.125 | 0.096 | 0.083 | 0.075 | 0.070 | 0.069 | 0.069 | 0.069 | 0.067 |

The experimental results, summarized in Fig. 1 and Table 1, highlight the superior performance of the proposed algorithm in integrating recognition results of string objects compared to both the ROVER method and the absence of integration. Across all four groups of text fields analyzed from the MIDV-500 dataset, the proposed algorithm consistently achieved lower error values, indicating its robustness in improving recognition accuracy. Fig. 1 illustrates the diminishing returns property, where the error reduction becomes less pronounced as the number of frames increases. This characteristic suggests that the proposed method efficiently utilizes the redundancy in video sequences while maintaining computational efficiency. Table 1 quantifies the error distances for various sequence lengths, showing that the proposed algorithm outperforms the ROVER method across all frame counts. For shorter sequences (e.g., three frames), the difference in error reduction is modest. Still, it becomes increasingly significant as the sequence length grows, reinforcing the algorithm's scalability and effectiveness in handling extended video streams.

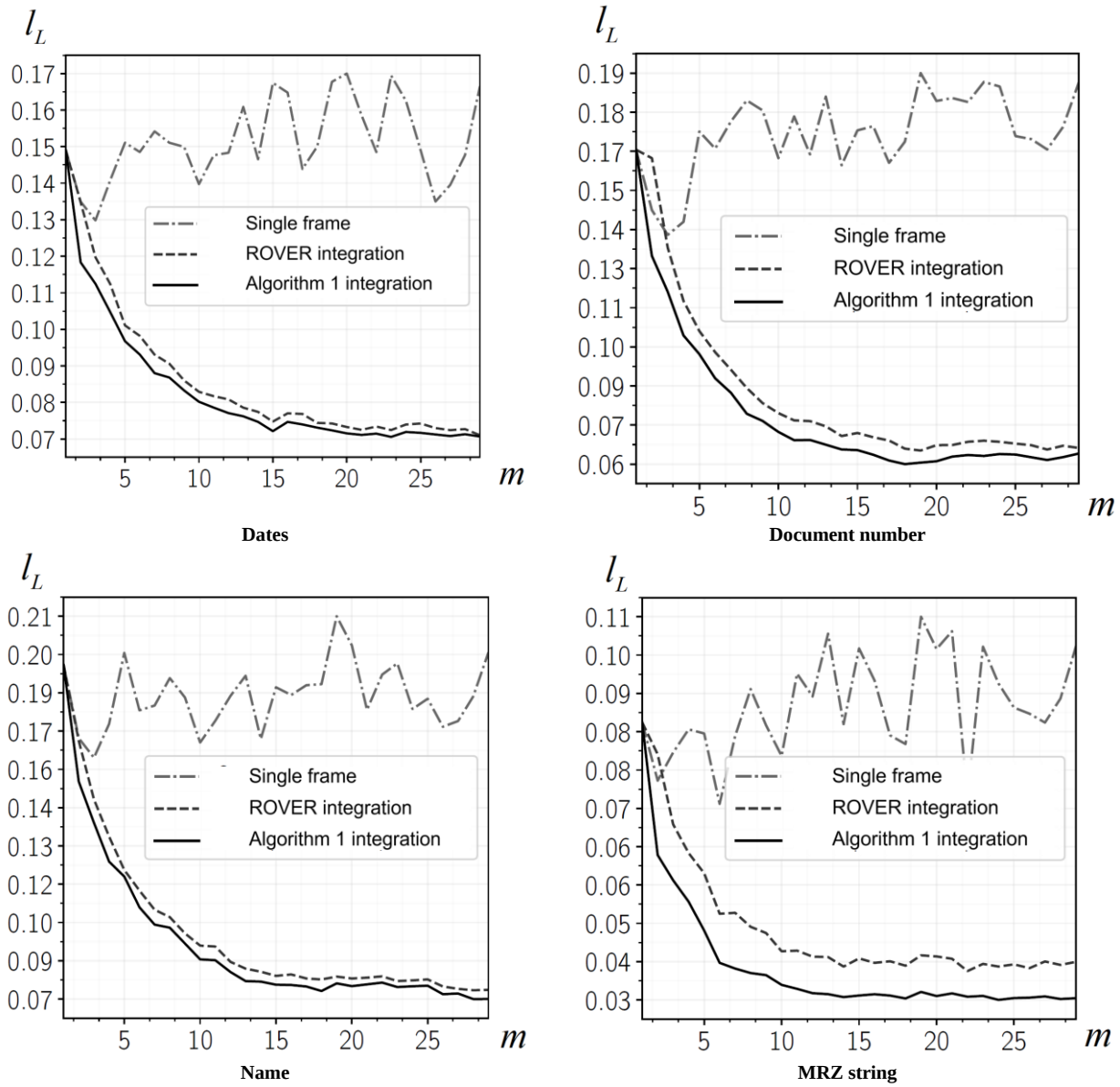


**Fig. 1. Results of the integration algorithms for four groups of text areas in the MIDV-500 dataset**

## CONCLUSIONS FROM THE PRESENT STUDY AND PROSPECTS FOR FURTHER RESEARCH IN THIS AREA

This study highlights the significant advancements achieved by the proposed algorithm for integrating string object recognition results in real-time video streams. Experimental findings validate its superiority over traditional methods, such as the ROVER approach, by demonstrating enhanced accuracy and robustness in addressing video stream distortions like defocusing, glare, and motion artefacts. The incorporation of extended result models, which consider alternative classification options, has proven essential in achieving these improvements, particularly in scenarios involving dynamic and challenging conditions. This work underscores the potential of advanced aggregation methods in reducing recognition errors and optimizing video-based text recognition processes.

Future research should aim to enhance the computational efficiency of the algorithm to ensure seamless real-time application on resource-constrained platforms, such as mobile devices, while also exploring its applicability to multilingual and complex textual formats. Additional efforts could focus on integrating preprocessing techniques like adaptive filtering and dynamic region selection to improve resilience against distortions further. Moreover, leveraging recent advancements in artificial intelligence and machine learning could help scale the algorithm's capabilities, ensuring adaptability to various data types and application contexts. Finally, extensive validation across diverse datasets is necessary to ensure the generalizability and reliability of the proposed method for a broader range of practical applications, including document digitization, automated data extraction, and real-time mobile-based text recognition systems.

## References

1. Mahajan S., Rani R. Text detection and localization in scene images: a broad review // Artificial Intelligence Review. – 2021. – Vol. 54, Issue 6. – P. 4317–4377. – Springer Science and Business Media LLC. – DOI: 10.1007/s10462-021-10000-8.
2. Liu H., Wang H., Bai J., Lu Y., Long S. DeepSSR: a deep learning system for structured recognition of text images from unstructured paper-based medical reports // Annals of Translational Medicine. – 2022. – Vol. 10, Issue 13. – P. 740. – AME Publishing Company. – DOI: 10.21037/atm-21-6672.
3. Zhang M., Joshi A., Kadmawala R., Dantu K., Poduri S., Sukhatme G. S. OCRdroid: A Framework to Digitize Text Using Mobile Phones // Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering. – 2010. – P. 273–292. – Springer Berlin Heidelberg. – DOI: 10.1007/978-3-642-12607-9_18.
4. Koceski S., Koceska N., Krstev A. Object Recognition Based on Local Features Using Camera-Equipped Mobile Phone // Communications in Computer and Information Science. – 2011. – P. 296–305. – Springer Berlin Heidelberg. – DOI: 10.1007/978-3-642-19325-5_30.
5. M M., Rajender U., A T., Rumale A. S. Real-time object detection in videos using deep learning models // ICTACT Journal on Image and Video Processing. – 2023. – Vol. 14, Issue 2. – P. 3103–3109. – ICT Academy. – DOI: 10.21917/ijivp.2023.0441.
6. Shi Y., Zhang T., Guo X. Practical Video Object Detection via Feature Selection and Aggregation // arXiv. – 2024. – Version 1. – DOI: 10.48550/ARXIV.2407.19650.
7. Zhao Y. Research and Design of Automatic Scoring Algorithm for English Composition Based on Machine Learning // Scientific Programming. – 2021. – Vol. 2021. – P. 1–10. – Hindawi Limited. – DOI: 10.1155/2021/3429463.
8. Tian Y. Artificial Intelligence Image Recognition Method Based on Convolutional Neural Network Algorithm // IEEE Access. – 2020. – Vol. 8. – P. 125731–125744. – Institute of Electrical and Electronics Engineers (IEEE). – DOI: 10.1109/access.2020.3006097.
9. Zhao X., Wang L., Zhang Y., Han X., Deveci M., Parmar M. A review of convolutional neural networks in computer vision // Artificial Intelligence Review. – 2024. – Vol. 57, Issue 4. – Springer Science and Business Media LLC. – DOI: 10.1007/s10462-024-10721-6.
10. AlKendi W., Gechter F., Heyberger L., Guyeux C. Advancements and Challenges in Handwritten Text Recognition: A Comprehensive Survey // Journal of Imaging. – 2024. – Vol. 10, Issue 1. – P. 18. – MDPI AG. – DOI: 10.3390/jimaging10010018.
11. Leszczuk M., Janowski L., Nawała J., Boev A. Objective Video Quality Assessment Method for Object Recognition Tasks // Electronics. – 2024. – Vol. 13, Issue 9. – P. 1750. – MDPI AG. – DOI: 10.3390/electronics13091750.
12. Dergachov K., Krasnov L., Bilozerskyi V., Zymovin A. Data pre-processing to increase the quality of optical text recognition systems // Radioelectronic and Computer Systems. – 2021. – Issue 4. – P. 183–198. – National Aerospace University - Kharkiv Aviation Institute. – DOI: 10.32620/reks.2021.4.15.
13. Abida K., Karray F., Abida W. A Novel Voting Scheme for ROVER Using Automatic Error Detection // Lecture Notes in Computer Science. – 2012. – P. 175–183. – Springer Berlin Heidelberg. – DOI: 10.1007/978-3-642-31368-4_21.
14. Thammarak K., Kongkla P., Sirisathitkul Y., Intakosum S. Comparative analysis of Tesseract and Google Cloud Vision for Thai vehicle registration certificate // International Journal of Electrical and Computer Engineering (IJECE). – 2022. – Vol. 12, Issue 2. – P. 1849–1858. – Institute of Advanced Engineering and Science. – DOI: 10.11591/ijece.v12i2.pp1849-1858.
15. Yujian L., Bo L. A Normalized Levenshtein Distance Metric // IEEE Transactions on Pattern Analysis and Machine Intelligence. – 2007. – Vol. 29, Issue 6. – P. 1091–1095. – Institute of Electrical and Electronics Engineers (IEEE). – DOI: 10.1109/tpami.2007.1078.
16. Fiscus J. G. A post-processing system to yield reduced word error rates: Recognizer Output Voting Error Reduction (ROVER) // Proceedings of the 1997 IEEE Workshop on Automatic Speech Recognition and Understanding. – 1997. – IEEE. – DOI: 10.1109/asru.1997.659110.
17. Gao H., Jia Y., Xu L., Pan F., Li S., Zhou Y. Anytime algorithm based on adaptive variable-step-size mechanism for path planning of UAVs // Chinese Journal of Aeronautics. – 2024. – Elsevier BV. – DOI: 10.1016/j.cja.2024.09.007.
18. Iglovikov V. ternaus/midv-500-models v0.0.2 (Version v0.0.2) // Zenodo. – 2020. – DOI: 10.5281/zenodo.4263532.