On optimal stopping strategies for text recognition in a video stream as an application of a monotone sequential decision model

Konstantin Bulatov^{1,2} · Nikita Razumnyi³ · Vladimir V. Arlazarov^{1,2}

Received: 14 November 2018 / Revised: 27 March 2019 / Accepted: 9 July 2019 / Published online: 23 July 2019
© The Author(s) 2019

Abstract

The paper describes the problem of stopping the text field recognition process in a video stream, which is a novel problem, particularly relevant to real-time mobile document recognition systems. A decision-theoretic framework for this problem is provided, and similarities with existing stopping rule problems are explored. Following the theoretical works on monotone stopping rule problems, a strategy is proposed based on thresholding the estimation of the expected difference between consequent recognition results. The efficiency of this strategy is evaluated on an openly accessible dataset. The results show that this method outperforms the previously published methods based on identical results cluster size thresholding. Notes on future work include incorporation of recognition result confidence estimations in the proposed model and more precise evaluation of the observation cost.

Keywords Recognition in video stream · Mobile OCR · Stopping rules · Decision making · Mobile document recognition · Anytime algorithms

1 Introduction

Mobile document analysis systems have become the focus of a wide range of research in the recent years [5,7,12,22]. Modern mobile devices are equipped with high-quality cameras and decent computing power which allows them to be used for camera-based document analysis tasks. Document recognition on mobile devices presents a set of challenges such as motion blur, defocus, glares on reflective document surface, insufficient camera resolution for accurate OCR (Optical Character Recognition), and other complications [8,24]. However, the possibility to process multiple frames in real time can be used to mitigate some of these difficulties. In particular, having the ability to recognize the same object multiple times in different video frames was shown

to significantly increase the overall recognition accuracy [6]. Multiple recognition results for the same document object (e.g. text field) can be either sorted in some order of preference or confidence and then a single result with the highest expected accuracy can be selected, or the results can be integrated together to produce the result which could be better than each individual frame result. Figure 1 presents examples of text fields as appeared in video frames, and Table 1 presents the corresponding frame recognition results and the integrated results as they change over time.

While object recognition in a video stream presents some advantages and helps to solve problems related to camera-based image analysis, it creates two novel problems which have to be addressed: the problem of optimal integration of multiple results (or selection of the best result among the candidates obtained from different image frames), and the problem of stopping the video stream recognition process. The stopping problem is particularly important in relation to real-time computer vision systems working on a mobile device [5,17,24], where the time required to obtain the result is often as important as the accuracy of the result itself. Although there are some published works mentioning multiple OCR results integration [6] and the video stream OCR stopping problem [2,5], the corpus of tested and evaluated methods related to this task is hardly sufficient.

This work was partially financially supported by Russian Foundation for Basic Research, Projects 17-29-03170 and 17-29-03370.

✉ Konstantin Bulatov
kbulatov@smartengines.com

¹ Institute for Systems Analysis, FRC CSC RAS, Moscow, Russia

² Smart Engines, Moscow, Russia

³ Moscow Institute of Physics and Technology, Moscow, Russia

Fig. 1 Examples of text field frames as appeared in the a video stream. Images were taken from MIDV-500 dataset [1] (clip TS 07 field 1 and clip HA10 field 2)

Table 1 Examples of single frame text field recognition results and the integrated recognition results

#	Frame result	Integrated	Frame result	Integrated
1	{RE0CZ	{RE0CZ	–	–
2	HUD\	{RE0CZ	I	–
3	M0N	E0CZ	W	–
4	<i>FREDEZ</i>	RE0CZ	A–	–
5	`REZIZ	RE0CZ	3" J" .1	–
6	MM	RE0CZ	(MEN	–
7	<i>FREDEZ</i>	FRE0EZ	<i>SPECIMEN</i>	E–
8	<i>FREDEZ</i>	<i>FREDEZ</i>	<i>SPECIMEN</i>	MEN
9	<i>FREDEZ</i>	<i>FREDEZ</i>	<i>SPECIMEN</i>	CIMEN
10	<i>FREDEZ</i>	<i>FREDEZ</i>	SPICIUEN	SPCIMEN
11	<i>FREDEZ</i>	<i>FREDEZ</i>	<i>SPECIMEN</i>	<i>SPECIMEN</i>
12	RREDEZ	<i>FREDEZ</i>	. –	<i>SPECIMEN</i>
...	...	...	...	...

Correct results are highlighted in italic. Recognition was performed using Tesseract v3.05.01 [23], on clips from MIDV-500 dataset [1] (clip TS07 field 1 and clip HA10 field 2), see Sect. 4 for detailed dataset analysis

From the system's composition perspective, the process of video stream OCR which independently processes each frame and integrates the recognition result can be regarded as an anytime algorithm [30]. From the desired properties of anytime algorithms, one can expect *interruptibility* (i.e. the process may be stopped after any frame and currently integrated result will be available as the answer) and *monotonicity* (i.e. on average the quality of integrated results does not decrease). However, the recognition result, at least in the case studied in this paper, does not have a *recognizable quality*, i.e. the quality of the current result cannot be determined at run time. It is worth mentioning that for OCR problems there exist other classes of anytime algorithms, such as algorithms which output partial recognition results: first for characters which are easy to recognize, and progressing to the most difficult characters, recognition of which demands higher computation time [19]. Although different from the task considered in this paper, for such cases the problem of optimal stopping is also relevant and also requires attention.

At the same time, there exists a large amount of works dedicated to more general stopping rule problems in mathematical statistics and decision theory [4,9,13,26]. This includes such well-researched problems as the secretary problem [20], house-selling problem [15], S_n/n^1 problem or the problem of maximizing the average [18] and others. The proofreading problem [13,14] can be regarded as particularly relevant to the text field recognition in a video stream. This problem goes as follows: a manuscript has been digitized with some number of errors M . A proofreading can be performed for the digitized version, each i th proofreading corrects X_i mistakes and costs a fixed amount c . Each mistake present in the final version of the text also has an inherent cost. The task is to make a decision after i th proofreading, whether the current version of the digitized manuscript should be published, or the review process should continue, in order to minimize the total expected cost. A number of solutions are proposed for this problem [14,28], based on different assumptions about the distribution of M and X_i . This problem has several variations and other applications, e.g. to software testing [11].

A certain similarity can be observed between proofreading stopping rule problem and the problem of stopping the video stream recognition process: the text field is recognized on multiple frames such that to obtain the recognition result for next frame some cost must be paid (which could represent the time required to pre-process the next frame and perform another text recognition iteration). Assuming that each time a single accumulated recognition result is defined, after each frame a decision must be made either to pay the additional observation cost and continue the recognition process in hope that the result would improve, or to stop the process and output the currently accumulated result. In this case the expected loss can be represented as a linear combination of expected

number of processed frames and a distance between expected recognition result to the correct text field value (in terms of some pre-defined metric).

There are important differences between a video stream recognition stopping rule problem and the proofreading problem, which have to be considered:

1. In most formulations of the proofreading problem, the value of X_i is taken to be non-negative, i.e. it is assumed that each proofreading either corrects some mistakes or at least does not introduce any new ones. In the text field recognition problem, there is no guarantee that the recognition result on the next frame, or the integrated result after processing the next frame, will always be closer to the correct value. However, it can be assumed that the integration algorithms are usually designed in such a way that the recognition of multiple versions of the object generally have higher accuracy than the recognition of a single image of an object.
2. The solutions for the proofreading problem or its variations typically rely on the decision maker having the ability to assess either the loss which will be suffered if the stopping decision is made, or the difference in losses between stages, as the observed value of X_i directly contributes to the loss function. In the case of text recognition in a video stream, however, the distance from the currently obtained result to the correct result could only be assessed using some recognition confidence estimations (which could suffer from overconfidence [21]), or could even be completely unavailable.

The proofreading problem is explored in [14], and it is shown that the optimal stopping rule can be constructed for such problem using a notion of *monotone stopping problems*. The theory of monotone stopping problems is given in detail in [10,13] and is used in this paper in order to construct the method for stopping the recognition process.

In summary, the goal of this paper is to explore and describe a decision-theoretic framework for finding the optimal stopping rule for the video stream text field recognition process, assuming that there are no text recognition confidence estimations available to the decision maker. Section 2 provides a formal problem statement and gives a theoretical background for the notion of monotone stopping problems. Section 3 describes an approach to stopping the video stream recognition process based on estimating the expected distance from the currently obtained integrated recognition result to the corresponding result on the next stage. Section 4 presents an experimental evaluation of the proposed approach on a publicly available identity documents video dataset with comparison to some other stopping rules. Section 5 provides some notes on a generalization of this model which could be a subject for a future work.

2 Theoretical framework

2.1 Problem statement

Consider the task of text field recognition in a video stream. Let $\mathbb{X}$ be the set of all text field recognition results (i.e. strings over some fixed alphabet), with a defined metric function $\rho : \mathbb{X} \times \mathbb{X} \rightarrow [0, +\infty)$. A text field with correct value $X^* \in \mathbb{X}$ is recognized such that a sequence of random recognition results $\mathbf{X} = (X_1, X_2, \dots)$ is observed one result at a time, each observation $x_i \in \mathbb{X}$ is a realization of X_i . We assume that $X_1, X_2, \dots$ have identical joint distribution with X^* . Within the scope of this paper, we also assume that there are no confidence estimations available for the text field recognition results.

Let us now define recognition results integrator as a function of several recognition results which produces a single integrated result $R : \mathbb{X}^+ \rightarrow \mathbb{X}$ (here by $\mathbb{X}^+$ we mean the set of all non-empty sequences of elements from $\mathbb{X}$). At any moment n the observations $X_1 = x_1, \dots, X_n = x_n$ are obtained and the integrated result $R_n = R(x_1, \dots, x_n)$ can be produced. The process may be stopped at any time $n > 0$ with a cost:

$$c_e \cdot \rho(R_n, X^*) + c_f \cdot n, \quad (1)$$

where $c_e > 0$ is the cost of recognition error in terms of distance from the ideal result, and $c_f > 0$ is the cost of each observation, representing, for example, the computational time required for the recognition of a text field in a single image.

Since both c_e and c_f are positive constants, the loss function can be simplified without changing the optimization problem:

$$L_n \stackrel{\text{def}}{=} \rho(R_n, X^*) + c \cdot n, \quad c = c_f/c_e. \quad (2)$$

The loss associated with stopping at time $n = 0$ (i.e. not taking any observations at all) may be regarded as infinite.

The task is to choose a time to stop the observations in order to minimize the expected loss. Let us formalize the problem using notation used in [13]. A stopping rule can be defined as a sequence of functions:

$$\Phi \stackrel{\text{def}}{=} (\phi_0, \phi_1(x_1), \phi_2(x_1, x_2), \phi_3(x_1, x_2, x_3), \dots), \quad (3)$$

where $\forall n : 0 \leq \phi_n(x_1, \dots, x_n) \leq 1$. The function $\phi_n(x_1, \dots, x_n)$ represents a conditional probability of stopping at the stage n given that stage n has been reached (i.e. given that the values $X_1 = x_1, \dots, X_n = x_n$ have been observed).

Based on the stopping rule Φ and the sequence of observations $\mathbf{X}$ a random stopping time N can be defined. Let

$P(N = n | \mathbf{X} = (x_1, x_2, \dots))$ denote a probability mass function for the stopping time N , i.e. the probability for stopping the process at stage n given the sequence of observations $\mathbf{X}$. This probability mass function relates to the stopping rule Φ as follows:

$$\begin{aligned} P(N = 0 | \mathbf{X} = (x_1, x_2, \dots)) &= \phi_0, \\ P(N = n | \mathbf{X} = (x_1, x_2, \dots)) &= \phi_n(x_1, \dots, x_n) \\ &\times \prod_{j=1}^{n-1} (1 - \phi_j(x_1, \dots, x_j)) \quad \forall n \in \{1, 2, \dots\}, \\ P(N = \infty | \mathbf{X} = (x_1, x_2, \dots)) \\ &= 1 - \sum_{j=0}^{\infty} P(N = j | \mathbf{X} = (x_1, x_2, \dots)). \end{aligned} \quad (4)$$

Conversely, given the random stopping time N , the stopping rule for $n \in \{0, 1, \dots\}$ can be expressed as a conditional probability of stopping at stage n given the sequence of observations $\mathbf{X}$ and given that the process did not stop at an earlier stage:

$$\phi_n(X_1, \dots, X_n) = P(N = n | N \geq n, \mathbf{X} = (x_1, x_2, \dots)), \quad (5)$$

thus both the sequence of functions Φ and the corresponding random variable N can be used to denote a stopping rule. We will use N from now on.

The problem is to choose a stopping rule N to minimize the expected loss $V(N)$, which can be expressed as follows:

$$V(N) = E(L_N(X_1, \dots, X_N)) \quad (6)$$

2.2 Principle of optimality

Since ρ cannot take negative values, $\forall n : L_n \geq c \cdot n$, and since c is a positive constant, we can assume the following:

$$\begin{aligned} E(\inf_n L_n) &> -\infty, \\ \liminf_{n \rightarrow \infty} L_n &\geq L_\infty \end{aligned} \quad (7)$$

Under assumptions (7), it can be shown [9,13] that the optimal stopping rule exists and it follows the principle of optimality.

Let V_n^* denote the minimal expected loss which can be obtained using a stopping rule N such that $P(N \geq n) = 1$, i.e. using a stopping rule that reaches stage n :

$$V_n^*(x_1, \dots, x_n) = \operatorname{ess\,inf}_{N \geq n} E_n(L_N), \quad (8)$$

where by $E_n(\cdot)$ for the sake of clarity we denote a conditional expectation $E(\cdot | X_1 = x_1, \dots, X_n = x_n)$, and $\operatorname{ess\,inf}_{N \geq n}$

denotes an essential infimum over a set of stopping rules which reach stage n . The essential infimum is used here instead of a regular infimum, because in general there are more than a countable number of stopping rules $N \geq n$ and an infimum over an uncountable set of random variables may not be measurable [13]. Thus, (8) effectively means that $P(V_n^*(x_1, \dots, x_n) \leq E_n(L_N)) = 1$ for all $N \geq n$ and if Z is any other random variable such that $\forall N \geq n : P(Z \leq E_n(L_N)) = 1$ then $P(Z \leq V_n^*(x_1, \dots, x_n)) = 1$.

The principle of optimality states that it is optimal to stop at stage n if and only if the loss suffered in such case is equal to the minimum expected loss obtainable for any stopping rule which reaches stage n . The connection between the minimum expected loss for any stopping rule $N \geq n$ and for any stopping rule which does not stop at stage n (and thus reaches stage $n+1$) can be expressed in form of an optimality equation:

$$V_n^* = \min\{L_n, E_n(V_{n+1}^*)\}. \quad (9)$$

Using assumptions (7) it can be proven [13] that equation (9) holds and that the following stopping rule is optimal:

$$N^* = \min\{n \geq 0 : L_n \leq E_n(V_{n+1}^*)\}. \quad (10)$$

In other words, the principle of optimality defines the optimal rule (10) as a rule that calls for stopping at the earliest stage n on which the loss is not higher than the minimal expected loss among all stopping rules which reach stage n .

2.3 Monotone problems

Monotone stopping rule problems [10,13] are a subset of stopping rule problems defined as follows. Let A_n denote the event $\{L_n \leq E_n(L_{n+1})\}$. The stopping rule problem is said to be monotone if

$$A_0 \subset A_1 \subset A_2 \subset \dots \quad (11)$$

The condition (11) means that if at some stage n the loss function is not higher than expected loss function at the next stage, then this will be true for all future stages as well.

Consider a stopping rule called one-stage look-ahead rule (also called a myopic rule):

$$N_A = \min\{n \geq 0 : L_n \leq E_n(L_{n+1})\}. \quad (12)$$

Rule N_A calls for stopping at stage n if the current loss is not higher than the loss which will be suffered if the process stops at stage $n+1$. It can be shown [10,13] that if the stopping rule problem is monotone and it has a finite horizon (i.e. for some fixed $T < \infty$ any stopping rule must stop when it

reaches stage T), then the one-stage look-ahead rule (12) is optimal, i.e. $V(N_A) = V(N^*)$.

In order to construct the method for stopping the video stream recognition process, in the next section we will formulate the conditions under which the problem could be considered monotone, at least from a certain stage, and then propose a stopping rule which approximates the behaviour of the one-stage look-ahead rule (12).

3 Proposed approach

3.1 Approximation of the optimal stopping rule

Let us make the following requirement for the integrator function R : the expected distances between two consecutive integrated recognition results decrease over time.

$$E(\rho(R_n, R_{n+1})) \geq E(\rho(R_{n+1}, R_{n+2})) \quad \forall n > 0. \quad (13)$$

In the terminology of anytime algorithms [30] assumption (13) means that the problem has a property of *diminishing returns*. With such assumption about the integration function R we can show that the stopping rule problem (6) with the loss function (2) becomes monotone starting from a certain stage.

Indeed, let B_n denote the event $\{E_n(\rho(R_n, R_{n+1})) \leq c\}$ and consider the stopping rule problem (6) starting from stage n on which B_n occurred for the first time. The events considered in condition (11) take the following form:

$$\begin{aligned} A_n : \{ & \rho(R_n, X^*) + cn \leq E_n(\rho(R_{n+1}, X^*)) + cn + c \} \\ & = \{ \rho(R_n, X^*) - E_n(\rho(R_{n+1}, X^*)) \leq c \}. \end{aligned} \quad (14)$$

With a fixed X^* at stage n using triangle inequality, we can obtain the relationship between the distance from the current result to the ideal, the expected distance to the result on next stage and the expected distance from the next result to the ideal:

$$\begin{aligned} \rho(R_n, X^*) & \leq E_n(\rho(R_n, R_{n+1})) + E_n(\rho(R_{n+1}, X^*)) \\ \Rightarrow \rho(R_n, X^*) - E_n(\rho(R_{n+1}, X^*)) & \leq E_n(\rho(R_n, R_{n+1})). \end{aligned} \quad (15)$$

If the right side of the inequality obtained in (15) is less than or equal to the constant c , then the left side must be as well, and hence if the event B_n occurs, the event A_n (14) must also occur. Moreover, using (13) we can see that if B_n occurs, then B_{n+1} will also occur, thus obtaining:

$$\forall n > 0 : B_n \subset A_n, \quad B_n \subset B_{n+1}. \quad (16)$$

Fig. 2 Relationship between the constructed rule N_B (based on estimating the expected distance from current integrated recognition result to the next) and the optimal stopping rule N^*

Hence, starting from stage n on which B_n occurred for the first time, events $A_n, A_{n+1}, A_{n+2}, \dots$ will occur as well, so the problem can be considered monotone starting from this stage, which means that among all stopping rules which reach stage n the one-stage look-ahead rule (12) is optimal, if the problem has a finite horizon.

Let us now consider stopping rule which tells the decision maker to stop on the earliest stage on which B_n occurs:

$$N_B = \min\{n > 0 : E_n(\rho(R_n, R_{n+1})) \leq c\}. \quad (17)$$

If N_B tells the decision maker to stop at stage n , then N_A also stops at this stage, and if the problem becomes monotone starting from stage n , decision of N_A is optimal, so the optimal rule N^* also stops at this stage. Moreover, if $\rho(R_n, X^*) - E_n(\rho(R_{n+1}, X^*)) > c$ the rule N_B does not stop, and neither does N^* following the principle of optimality. Thus, assuming (13), N_B will not stop preliminary, and if it tells the decision maker to stop, then it is optimal to stop. Figure 2 graphically represents the differences between rules N_B and N^* under possible relationships between A_n and B_n .

The set of situations in which N_B does not stop and N^* may stop is conditioned by two main drawbacks of N_B : it relies on the estimation of difference in loss function values using triangle inequality and thus is inefficient if integrated results degrade (i.e. if $\rho(R_n, X^*) - E_n(\rho(R_{n+1}, X^*)) < 0$), and it performs by thresholding the expected metric which could be without any upper bound. Thus, in our proposed approach we will assume that the metric on the set of all text field recognition results has an upper bound (that is, $\exists G : \forall x, y \in \mathbb{X} : 0 \leq \rho(x, y) \leq G$) and that the integrator function R yields better results over time:

$$E(\rho(R_n, X^*)) \geq E(\rho(R_{n+1}, X^*)) \quad \forall n > 0. \quad (18)$$

In summary, the proposed approach for stopping rule is, firstly, to estimate the expected distance between the current

integrated recognition result R_n (which is known at stage n) and the unknown next result R_{n+1} , and secondly, to perform thresholding of this estimation, thus approximating the behaviour of rule N_B .

3.2 Estimation of the expected distance

As required by the stopping rule N_B , on n th stage of the process an estimation has to be provided for the expected distance between integrated results $\Delta_n \stackrel{\text{def}}{=} E(\rho(R_n, R_{n+1}))$ given current observations $X_1 = x_1, \dots, X_n = x_n$. We will assume that the integrator function R is available and can be used to compute such estimation. In the scope of this paper, we propose to estimate Δ_n by modelling the integrated recognition result on the next stage assuming that the new observation will be close to the ones already obtained:

$$\hat{\Delta}_n \stackrel{\text{def}}{=} \frac{1}{n+1} \left(\delta + \sum_{i=1}^n \rho(R_n, R(x_1, x_2, \dots, x_n, x_i)) \right), \quad (19)$$

where δ is an external parameter. In general the selection of the method for estimating the next integrated recognition result (or the expected distance between it and the current result) might depend on the nature of the integrator function R and other specifics of the problem. Other versions of such estimation for the case of text field recognition could be considered in future works.

4 Experimental evaluation

4.1 Evaluated integrator and distance metrics

In order to apply the model presented in Sect. 3, we have to define a metric ρ and an integrator function R for the set of possible text field recognition results $\mathbb{X}$, i.e. on the set of strings. As metrics we considered strict string equality ρ_E and a Normalized Levenshtein Distance ρ_L [29]:

$$\rho_E(x, y) \stackrel{\text{def}}{=} \begin{cases} 0 & \text{if } x = y, \\ 1 & \text{if } x \neq y \end{cases} \quad (20)$$

$$\rho_L(x, y) \stackrel{\text{def}}{=} \frac{2 \cdot \text{levenshtein}(x, y)}{|x| + |y| + \text{levenshtein}(x, y)},$$

where $|x|$ is the length of the string x and $\text{levenshtein}(x, y)$ is the Levenshtein distance between strings x and y . The triangle inequality holds for both metrics, and the maximum value of both metrics is 1.

As an integrator function R we evaluated a version of the ROVER (Recognizer Output Voting Error Reduction) method [16], which is used for merging text field recognition results produced with different recognition algorithms [25,27] and for accumulating text recognition result in a video stream [6]. The algorithm consists of two modules: the alignment module performs an optimal wrapping of each incoming string to a word transition network, then the voting module selects the best result by traversing the net-

Table 2 Average text field recognition result metrics for Tesseract [23] on MIDV-500 dataset [1]

	Numeric dates	Doc. numbers	MRZ lines	Names (Latin)	All fields
Unique fields	91	48	30	79	248
Clips in total	824	436	260	719	2239
Images in total	17,735	9329	5096	15,587	47,747
Aver. clip size	21.523	21.397	19.600	21.679	21.325
<i>Tesseract v3.05.01</i>					
$E(\rho_E(X_i, X^*))$	0.792	0.791	0.912	0.759	0.794
$E(\rho_E(R_{\text{last}}, X^*))$	0.660	0.702	0.842	0.629	0.679
$E(\rho_E(R_{30}, X^*))$	0.664	0.697	0.850	0.629	0.681
$E(\rho_L(X_i, X^*))$	0.360	0.422	0.258	0.443	0.388
$E(\rho_L(R_{\text{last}}, X^*))$	0.244	0.326	0.162	0.338	0.281
$E(\rho_L(R_{30}, X^*))$	0.246	0.323	0.164	0.336	0.280
<i>Tesseract v4.0.0</i>					
$E(\rho_E(X_i, X^*))$	0.570	0.586	0.940	0.475	0.581
$E(\rho_E(R_{\text{last}}, X^*))$	0.424	0.440	0.904	0.293	0.441
$E(\rho_E(R_{30}, X^*))$	0.426	0.447	0.904	0.295	0.444
$E(\rho_L(X_i, X^*))$	0.238	0.287	0.339	0.250	0.262
$E(\rho_L(R_{\text{last}}, X^*))$	0.123	0.160	0.277	0.125	0.149
$E(\rho_L(R_{30}, X^*))$	0.125	0.163	0.279	0.127	0.151

X_i is the result from any frame, R_{last} is the full clip result produced using ROVER, R_{30} is the extended clip result produced using ROVER

work and choosing the best result on each stage. In case of strings integration the transition network can be built by a Levenshtein-based optimal wrapping of the incoming string, and the voting can be frequency-based (as we assume $\mathbb{X}$ to be a set of strings so there are no character recognition confidence estimation available). To implement such method, an empty symbol should be defined with a fixed voting weight relative to the weight of each symbol of the alphabet. In the performed experiments, we assumed the weight of empty symbol to be 0.6.

4.2 Dataset

The evaluation was performed on a publicly available dataset MIDV-500 [1] which contains video clips of 50 identity documents (10 clips per document, 30 frames per clip) with annotated positions and values of text fields. To follow the original publication [1] we analysed four field groups: numeric dates, document number, MRZ (machine-readable zone) lines and Latin name components.

Only the frames on which the document is fully visible were considered; thus, the clips in the analysed dataset had different lengths (from 1 to 30 frames). In order to minimize the extra normalization effects and provide a more clear presentation of the results, each evaluated clip was extended up to the length of 30 using repetitions from the beginning of the clip (thus, all evaluated clips for the performed experiments had the same length).

Each field was cropped from the original image according to the combined ground truth of document boundaries and template text field coordinates, with added margins equal to 10% of the smallest text field dimension. Each text field cropping size corresponded to 300 DPI resolution and was recognized using open-source text recognition engine Tesseract [23] (versions v3.05.01 and v4.0.0) using default parameters for text line recognition in the English language. All character comparison was case-insensitive, and Latin letter *O* was treated as equal to the digit 0.

Table 2 for each field group lists the number of unique fields in MIDV-500 dataset, the total number of field images (across all frames with fully visible document) and average field image sequence size. The table presents average distances from the frame result X_i to the correct result X^* , and from the integrated result for the whole clip to the correct result X^* before extension (R_{last}) and after extension (R_{30}), for distance metrics ρ_E and ρ_L .

Figure 3 illustrates the average distances from the integrated recognition results to the correct result for all evaluated text fields recognized using Tesseract v3.05.01. The figure shows the significant error decrease over time in both metrics, which can be seen as a practical justification of assumption (18).

Fig. 3 Average distances from the frame results and from the integrated results to the correct values, for all evaluated text fields, in terms of distance metrics ρ_E (top) and ρ_L (bottom), text recognition performed using Tesseract v3.05.01

Figure 4 illustrates the average decrease in the distance from the integrated recognition result to the correct result $E(\rho(R_n, X^*)) - E(\rho(R_{n+1}, X^*))$ over time, the expected distance Δ_n between consequent integrated results, and its estimation $\hat{\Delta}_n$. In the performed experiments the estimation parameter δ (19) with value 0.2 was used for both metrics. The figure shows that although the stopping rule N_B is a very rough approximation of N_A , there is a practical justification of assumption (13) and that $\hat{\Delta}_n$ (19) provides a decent real-time estimation of Δ_n starting from $n = 2$. The approximation of the stopping rule N_B can now be implemented by thresholding the estimation value $\hat{\Delta}_n$.

4.3 Evaluation of stopping rules

In order to evaluate the efficiency of stopping rules, a performance profile can be constructed which would graphically show the change of average number of integrated observations and the corresponding average distance from the obtained result (at stopping time) to the correct one, as

Fig. 4 Average decrease in the distance between consequent integrated results and its proposed estimation, in terms of distance metrics ρ_E (top) and ρ_L (bottom), for all evaluated fields, $\delta = 0.2$, text recognition performed using Tesseract v3.05.01

the observation cost c is varied. Such performance profile represents a trade-off between the computational time and accuracy of the integrated recognition results and allows to compare different stopping strategies visually.

As a baseline a simple counting stopping rule N_K was evaluated, which stops at a fixed stage K . Additionally, two variations of stopping rule explored in [2] were evaluated. Since the original paper relies on recognition result confidence estimations, which are unavailable in the scope of this paper, the stopping rule described in [2] is reduced to thresholding at stage n the size of the largest cluster of identical recognition results accumulated up to stage n . Thus we constructed the stopping rule N_{CX} , which thresholds the largest cluster of identical frame recognition results among $x_1, \dots, x_n$, and N_{CR} , which does the same for integrated results $R_1, \dots, R_n$. Finally, the stopping rule N_B , constructed in this paper, estimates at stage n the expected distance Δ_n to the next integrated result and stops when the estimation becomes lower or equal to a constant threshold. In the performed experiments the stopping rule N_B only became

Fig. 5 Stopping rule performance profiles: plot of the average distance from the obtained result to the correct one against the average number of processed frames as the stopping rule threshold is varied, for distance metrics ρ_E (top) and ρ_L (bottom), for all evaluated fields, recognized using Tesseract v3.05.01

active starting from stage $n = 2$ (i.e. when the estimation $\hat{\Delta}_n$ becomes more justified), and the estimation parameter δ (19) with value 0.2 was used for both metrics.

Figure 5 illustrates the stopping rules efficiency for all fields when recognized in a video stream using Tesseract recognition engine v3.05.01. The lower position of the curve signifies the better stopping rule performance. It can be observed that in general the proposed stopping rule N_B outperforms the other evaluated stopping rules, especially in ρ_L metric.

Figure 6 shows the performance profiles for the same stopping rules and with the same algorithm parameters, but using Tesseract v4.0.0. It should be noted that the method performs well without modification for two separate versions of the Tesseract recognition engine, which employ different text recognition algorithms.

Table 3 shows the average distance from the integrated result to the correct result at stopping time, which could be achieved using the evaluated stopping rules, with text fields

Fig. 6 Stopping rule performance profiles: plot of the average distance from the obtained result to the correct one against the average number of processed frames as the stopping rule threshold is varied, for distance metrics ρ_E (top) and ρ_L (bottom), for all evaluated fields, recognized using Tesseract v4.0.0

recognized using Tesseract v3.05.01. Columns of the Table 3 represent target intervals for the average number of processed frames, rows represent the evaluated stopping rules, and each cell contain the data point with the smallest average number

of observations within the target interval and the corresponding average distance to the correct result in terms of metric ρ_L . Some cells of the table do not contain any data—this signifies that the corresponding stopping rule could not achieve the average number of observations in the target interval on the evaluated dataset (due to more discrete nature of the thresholded parameter). It can be seen that almost in all target intervals the stopping rule N_B proposed in this paper outperforms the other evaluated stopping rules. The similar result can be observed for fields recognized using Tesseract v4.0.0 (results are presented in Table 4).

Stopping rule performance profiles for separate text field groups are presented in Figs. 7 (for Tesseract v3.05.010 and 8 (for Tesseract v4.0.0).

5 Notes on a generalization

In the problem statement explored in Sect. 2, it was assumed that each new observation has a constant cost c_f (1). In the real text field recognition system, this cost depends on the time required to produce the next recognition result, which may increase or decrease from stage to stage. Moreover, the observation cost may depend on the stopping rule itself (e.g. include the time required to compute the estimation $\hat{\Delta}_n$) and on the stopping rules for other objects (e.g. the time required to obtain the next recognition result for a field may depend on whether the recognition process for other fields have stopped and thus on the next frame those fields will not be recognized).

It should be noted that assumption (13) is sufficient but not essential for obtaining $\forall n > 0 : B_n \subset B_{n+1}$, as it would also be sufficient to directly require that if starting from some stage n the expected difference $E_n(\rho(R_n, R_{n+1}))$ becomes bounded by c then it would be true for all later stages. For the purposes of this paper, we assumed (13), but a more general approach may be required in future generalizations of the model.

Table 3 Achieved average distances from the integrated result at stopping time to the correct result, in terms of metric ρ_L , for all evaluated fields, recognized performed using Tesseract v3.05.01

Stopping method	Measured parameter	Target interval for the average number of observations $E(N)$								
		3 ± 0.5	4 ± 0.5	5 ± 0.5	6 ± 0.5	7 ± 0.5	8 ± 0.5	9 ± 0.5	10 ± 0.5	11 ± 0.5
N_{CX} (adapted [2])	$E(N)$	No data	No data	No data	No data	No data	7.727	No data	No data	No data
	$E(\rho_L(R_N, X^*))$						0.298			
N_{CR} (adapted [2])	$E(N)$	3.230	No data	No data	5.909	No data	8.375	No data	No data	10.560
	$E(\rho_L(R_N, X^*))$	0.329			0.306		0.292			0.286
N_K (fixed stage K)	$E(N)$	3.000	4.000	5.000	6.000	7.000	8.000	9.000	10.000	11.000
	$E(\rho_L(R_N, X^*))$	0.347	0.329	0.315	0.308	0.300	0.295	0.294	0.288	0.287
N_B (proposed)	$E(N)$	2.509	3.558	4.527	5.521	6.531	7.691	8.554	9.715	10.830
	$E(\rho_L(R_N, X^*))$	0.351	0.322	0.302	0.292	0.285	0.282	0.280	0.278	0.278

Bold values indicate the column-wise best values for the two measured parameters

Table 4 Achieved average distances from the integrated result at stopping time to the correct result, in terms of metric ρ_L , for all evaluated fields, recognized performed using Tesseract v4.0.0

Stopping method	Measured parameter	Target interval for the average number of observations $E(N)$								
		3 ± 0.5	4 ± 0.5	5 ± 0.5	6 ± 0.5	7 ± 0.5	8 ± 0.5	9 ± 0.5	10 ± 0.5	11 ± 0.5
N_{CX} (adapted [2])	$E(N)$	No data	No data	5.332	No data	No data	8.471	No data	No data	10.901
	$E(\rho_L(R_N, X^*))$			0.195			0.170			0.162
N_{CR} (adapted [2])	$E(N)$	2.936	No data	5.099	No data	6.920	No data	8.594	10.103	No data
	$E(\rho_L(R_N, X^*))$	0.227		0.201		0.180		0.167	0.164	
N_K (fixed stage K)	$E(N)$	3.000	4.000	5.000	6.000	7.000	8.000	9.000	10.000	11.000
	$E(\rho_L(R_N, X^*))$	0.237	0.213	0.197	0.191	0.185	0.180	0.178	0.171	0.171
N_B (proposed)	$E(N)$	2.580	3.551	4.571	5.539	6.683	7.742	8.771	9.779	10.726
	$E(\rho_L(R_N, X^*))$	0.224	0.188	0.174	0.165	0.161	0.161	0.158	0.158	0.157

Bold values indicate the column-wise best values for the two measured parameters

Fig. 7 Stopping rule performance profiles: plot of the average distance from the obtained result to the correct one against the average number of processed frames as the stopping rule threshold is varied, for distance metrics ρ_E (left) and ρ_L (right), for separate field groups, recognized using Tesseract v3.05.01 **Fig. 8** Stopping rule performance profiles: plot of the average distance from the obtained result to the correct one against the average number of processed frames as the stopping rule threshold is varied, for distance metrics ρ_E (left) and ρ_L (right), for separate field groups, recognized using Tesseract v4.0.0

The model of a recognition result presented in Sect. 2 is rather simplified as it does not include any kind of confidence estimations for the overall field recognition result or for separate characters (and thus it is assumed that the decision maker does not have any grounds for estimating the distances $\rho(x_n, X^*)$ or $\rho(R_n, X^*)$). If the confidence estimations are available, the described model and the stopping rule N_B are still valid; however, there could exist a better approximation of the one-stage look-ahead rule N_A (12) which relies on text field recognition result confidence and on modelling the behaviour of this confidence in a video stream [3]. The generalization of the decision-theoretic stopping rule problem framework proposed in this paper by incorporating the recognition result confidence information is a subject of a future work.

6 Conclusion

The paper described the task of stopping a video stream recognition process, which is an important and novel problem which has to be addressed in mobile document recognition systems design. We presented a problem statement following the classic formulation of a corresponding problem in decision theory and proposed an original stopping method which treats the text field recognition in a video stream as a monotone stopping rule problem and which relies on approximating the optimal stopping rule by estimating the distance to the next integrated recognition result. The method was evaluated on an openly accessible dataset MIDV-500 with Tesseract as a recognition engine. It can be seen that the proposed stopping rule performs better than thresholding of the number of processed frames or the size of the maximal cluster of identical preliminary results, considering that no text field recognition confidence estimations were available to the model.

In a future work, this model can be extended to incorporate confidence estimations which could allow for better approximation of the one-stage look-ahead stopping rule, on which the proposed method is based. Additionally the model could be generalized as to account for variable observations cost and dependence on the stopping rules for other objects.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

1. Arlazarov, V.V., Bulatov, K., Chernov, T., Arlazarov, V.L.: A dataset for identity documents analysis and recognition on mobile devices in video stream (2018). ArXiv e-prints [arXiv:1807.05786](https://arxiv.org/abs/1807.05786)
2. Arlazarov, V.V., Bulatov, K., Manzhikov, T., Slavin, O., Janiszewski, I.: Method of determining the necessary number of observations for video stream documents recognition. In: Proceedings of the SPIE, vol. 10696, pp. 1–6 (2018). <https://doi.org/10.1117/12.2310132>
3. Arlazarov, V.V., Slavin, O., Uskov, A., Janiszewski, I.: Modelling the flow of character recognition results in video stream. Bull. South Ural State Univ. **11**, 14–28 (2018). <https://doi.org/10.14529/mmp180202>
4. Berezovsky, B.A., Gnedin, A.V.: Problems of Best Selection. Academia Nauk USSR, Moscow (1984). (in Russian)
5. Bulatov, K., Arlazarov, V.V., Chernov, T., Slavin, O., Nikolaev, D.: Smart IDReader: document recognition in video stream. In: 14th International Conference on Document Analysis and Recognition (ICDAR), vol. 6, pp. 39–44 (2017)
6. Bulatov, K.B., Kirsanov, V.Y., Arlazarov, V.V., Nikolaev, D.P., Polevoy, D.V.: Methods for integration the results of the documents text fields recognition in the video stream of a mobile device. RFBR J. **92**(4), 109–115 (2016). <https://doi.org/10.22204/2410-4639-2016-092-04-109-115>. (in Russian)
7. Burie, J.C., Chazalon, J., Coustaty, M., Eskenazi, S., Luqman, M.M., Mehri, M., Nayef, N., Ogier, J.M., Prum, S., Rusiñol, M.: ICDAR2015 competition on smartphone document capture and OCR (SmartDoc). In: International Conference on Document Analysis and Recognition (ICDAR), pp. 1161–1165 (2015)
8. Chernov, T., Kolmakov, S., Nikolaev, D.: An algorithm for detection and phase estimation of protective elements periodic lattice on document image. Pattern Recognit. Image Anal. **27**(1), 53–65 (2017). <https://doi.org/10.1134/S1054661817010023>
9. Chow, Y., Robbins, H., Siegmund, D.: Great Expectations: The Theory of Optimal Stopping. Houghton Mifflin, Boston (1971)
10. Chow, Y.S., Robbins, H.: A martingale system theorem and applications. In: Proceedings of the 4th Berkeley Symposium on Mathematics, Statistics and Probability, vol. 1, pp. 93–104. University of California Press, Berkeley, CA (1961)
11. Dalal, S.R., Mallows, C.L.: When should one stop testing software? J. Am. Stat. Assoc. **83**(403), 872–879 (1988)
12. Esser, D., Muthmann, K., Schuster, D.: Information extraction efficiency of business documents captured with smartphones and tablets. In: Proceedings of the 2013 ACM Symposium on Document Engineering, ACM, DocEng'13, pp. 111–114 (2013). <https://doi.org/10.1145/2494266.2494302>
13. Ferguson, T.S.: Optimal stopping and applications (2008). <https://www.math.ucla.edu/~tom/Stopping/Contents.html>. Accessed 13 Nov. 2018
14. Ferguson, T.S., Hardwick, J.P.: Stopping rules for proofreading. J. Appl. Probab. **26**(2), 304–313 (1989)
15. Ferguson, T.S., Klass, M.J.: House-hunting without second moments. Seq. Anal.: Des Methods Appl. **29**, 236–244 (2010)
16. Fiscus, J.G.: A post-processing system to yield reduced word error rates: recognizer output voting error reduction (ROVER). In: IEEE Automatic Speech Recognition and Understanding Workshop, pp. 347–354 (1997). <https://doi.org/10.1109/ASRU.1997.659110>
17. Hartl, A., Arth, C., Schmalstieg, D.: Real-time detection and recognition of machine-readable zones with mobile devices. In: Proceedings of the 10th International Conference on Computer Vision Theory and Applications, vol. 3, pp. 79–87 (2015). <https://doi.org/10.5220/0005294700790087>
18. Klass, M.J.: Properties of optimal extended-valued stopping rules for s_n/n^1 . Ann Probab **1**(5), 719–757 (1973)
19. Kobayashi, T., Iwamura, M., Matsuda, T., Kise, K.: An anytime algorithm for camera-based character recognition. In: Proceedings of the 12th International Conference on Document Analysis and Recognition, pp. 1140–1144 (2013). <https://doi.org/10.1109/ICDAR.2013.231>

20. Mucci, A.G.: On a class of secretary problems. *Ann. Probab.* **1**(3), 417–427 (1973)
21. Pereyra, G., Tucker, G., Chorowski, J., Kaiser, L., Hinton, G.: Regularizing neural networks by penalizing confident output distributions (2017). ArXiv e-prints [arXiv:1701.06548](https://arxiv.org/abs/1701.06548)
22. Puybureau, E., Geraud, T.: Real-time document detection in smartphone videos. In: 25th International Conference on Image Processing (ICIP), pp. 1498–1502 (2018). <https://doi.org/10.1109/ICIP.2018.8451533>
23. Smith, R.: An overview of the Tesseract OCR engine. In: Proceedings of the 9th International Conference on Document Analysis and Recognition. IEEE Computer Society, ICDAR'07, vol. 2, pp. 629–633 (2007)
24. Sourvanos, N., Tsatiris, G.: Challenges in input preprocessing for mobile OCR applications: a realistic testing scenario. In: 9th International Conference on Information, Intelligence, Systems and Applications, pp. 1–5 (2018). <https://doi.org/10.1109/IISA.2018.8633688>
25. Stuner, B., Chatelain, C., Paquet, T.: LV-ROVER: lexicon verified recognizer output voting error reduction (2017). ArXiv e-prints [arXiv:1707.07432](https://arxiv.org/abs/1707.07432)
26. Tamaki, M.: On the optimal stopping problems with monotone thresholds. *J. Appl. Probab.* **52**(4), 926–940 (2015). <https://doi.org/10.1239/jap/1450802744>
27. Wemhoener, D., Yalniz, I.Z., Manmatha, R.: Creating an improved version using noisy OCR from multiple editions. In: Proceedings of the 12th International Conference on Document Analysis and Recognition. IEEE Computer Society, ICDAR'13, pp. 160–164 (2013). <https://doi.org/10.1109/ICDAR.2013.39>
28. Yang, M.C.K., Wackerly, D.D., Rosalsky, A.: Optimal stopping rules in proofreading. *J. Appl. Probab.* **19**(3), 723–729 (1982)
29. Yujian, L., Bo, L.: A normalized levenshtein distance metric. *IEEE Trans. Pattern Anal. Mach. Intell.* **29**(6), 1091–1095 (2007). <https://doi.org/10.1109/TPAMI.2007.1078>
30. Zilberstein, S.: Using anytime algorithms in intelligent systems. *AI Mag.* **17**, 73–83 (1996)

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.