Neural Computing and Applications (2022) 34:2473–2493
https://doi.org/10.1007/s00521-022-06887-1

ORIGINAL ARTICLE

Automatic and intelligent content visualization system based on deep learning and genetic algorithm

Murat İnce[1]

Received: 12 August 2020 / Accepted: 4 January 2022 / Published online: 15 January 2022

Abstract

Increasing demand in distance education, e-learning, web-based learning, and other digital sectors (e.g., entertainment) has led to excessive amounts of e-content. Learning objects (LOs) are among the most important components of electronic content (e-content) and are preserved in learning object repositories (LORs). LORs produce different types of electronic content. In producing e-content, several visualization techniques are employed to attract users and ensure a better understanding of the provided information. Many of these visualization systems match images with corresponding text using methods such as semantic web, ontologies, natural language processing, statistical techniques, neural networks, and deep neural networks. Unlike these methods, in this study, an automatic and intelligent content visualization system is developed using deep learning and popular artificial intelligence techniques. The proposed system includes subsystems that segment images to panoptic image instances and use these image instances to generate new images using a genetic algorithm, an evolution-based technique that is one of the best-known artificial intelligence methods. This large-scale proposed system was used to test different amounts of LOs for various science fields. The results show that the developed system can be efficiently used to create visually enhanced content for digital use.

Keywords Panoptic segmentation · Deep learning · Genetic algorithm · e-content visualization

1 Introduction

Distance education provides learners with the opportunity to train themselves individually in a flexible environment [1]. When there are obstacles to traditional education—such as the Covid-19 pandemic—the use of distance education increases, and educational data grow exponentially [2, 3]. Moreover, several studies have demonstrated the importance and popularity of distance education [4]. E-learning, web-based education, and web-based training are some example applications of distance education [5]. These systems contain enormous amounts of unstructured content [6]. Given the increasing demand for these types of educational activities [7], the need for qualified e-content has emerged as a problem [8]. These educational systems involve huge amounts of e-content containing plaintext, images, and video. More specifically, e-content is produced with learning objects (LOs) [9]: content, educational, and information objects and reusable learning resources [10]. LOs can contain images, text, video, sound, animations, simulations, graphs, and tables that have educational significance [11]. LO is supported by the IEEE LOM, IMS, and DCMI standards, which require certain properties, such as accessibility, interoperability, compatibility, and reusability [12]. In order to meet these criteria, LOs are combined with descriptive information called metadata [13]. LOs and their identifier metadata are stored in learning object repositories (LORs) [14]. Thanks to LORs, LOs are reusable, shareable, and interoperable with other systems and are also readable by computers [10]. Producing high-quality e-content generally requires a great deal of time and financial resources, but such content can be produced easily and quickly using LOs from LORs. LOs can be segmented into smaller objects, which can in turn be combined to produce different large-scale LOs [15]. Text-based content can be visualized during this process [16].

✉ Murat İnce
muratince@isparta.edu.tr

[1] Vocational School of Technical Sciences, Isparta University of Applied Sciences, Cunur West Campus, 32200 Isparta, Turkey

Visualization of learning content is important because it simplifies topics and increases learners' comprehension of the information [17]. Nowadays, visualization systems are needed because they aid in clarifying meaning for users with learning disabilities [18]. Images, video, animations, and simulations are some visualization techniques used to attract users, improve concentration, and make sense of the given information within the content [19, 20]. Moreover, visualization helps ensure that information is stored in the learner's long-term memory [21, 22]. Studies show that people can have a higher rate of remembering when they are exposed to a visualized text than when they are not [23, 24]. Another important effect of visualization is that it improves user retention with regard to color, size, shape, orientation, position, organization, and relations in content design [25]. Moreover, visualization is usable not only in education but also in daily life, such as business, entertainment, and other digital contexts [26, 27].

Nowadays, digital documents are extensively used for educational purposes and also used excessively in every aspect of our daily lives. Digitalization has thus created a need for interactive and effective text visualization techniques. Several studies have automatically converted natural language texts to images that represent the meaning of the corresponding texts [28]. Van Wierst et al. [29] developed BolVis to help researchers in philosophy, which largely depends on a plethora of reading texts. Researchers can filter, compare, and explore the meanings of the most relevant part of the large-scale reading texts. Another study developed the Text Variation Explorer software to effectively visualize changes in sociolinguistics studies and provide a generic structure for use in other linguistics studies [30]. Singh et al. [31] combined discriminative feature selection and latent topic analysis to visualize representative data for a large-scale corpus obtained from conference proceedings, movie summaries, and newsgroup postings. They were inspired by the wheel of emotion to visualize data in the form of a flower. In Sui's study [32], the timely topic scoring technique was used to show topic trends in Twitter text over the studied time period. Another usage of text visualization is clinical studies [33]. Patients' documents are in unstructured text format, and given the number of these documents, clinicians often cannot deal with all of them in a short time. Thus, MedStory was developed to handle long texts using text visualization. Sprint is a system developed by Yamada et al. [34] to generate 3D models from text descriptions of a scene. Similarly, Joshi et al. [35] developed a story picture engine to represent text with some pictures. Mihalcea et al. [36] developed a similar system to generate pictures to automatically represent simple sentences. In this system, a WordNet structure is used as a lexical source. Utkus is another system based on an ontology for the Russian language that was used to create representative pictures for object behavior [37]. Bui et al. [38] developed another medical text-to-picture system to visualize patient instructions. In a similar study, Ruan et al. [39] summarized patients' medical data. Jiang et al. [40] developed an instant messaging software to obtain images of queried keywords. Vishit is another software that was used for the speakers of the Hindu language to ease communication between different cultures using semantic-based text-to-image visualization [41]. Another use of a text visualization system is broadcasting news with enhanced representation of emotions [42]. Although there are many text-to-image systems, Hassani and Lee [43] proposed that they are not at the desired level. They wanted to create scenes, rather than showing a representative image for a given text. NALIG [44], WordsEye [45], VizStory [46], and mobile Arabic story scene visualizer [47] are some examples of text-to-scene systems. Manufacturing [48], mapping [49], and education [50] are some of the other application areas of text visualization systems. These studies can also be used for educational e-content. Moreover, there are studies more specific to educational purposes. For example, Gunarathne et al. [51] developed a web-based LO visualization system that works on the well-known MERLOT II LOR. In this system, when users search a keyword, results are visualized using data extraction, transformation, and clustering methods. In another study, the DLNotes2 tool, which is based on semantic networks, was developed to visualize learning activity results in digital libraries [52]. In these studies, semantic web, ontologies, natural language processing, statistical methods, neural networks, and deep neural networks have been used to represent text with images.

The objective of this study is to provide an alternative, hybrid method for an automatic and intelligent e-content visualization system that differs from the aforementioned studies. By employing both panoptic image segmentation and genetic algorithms, the proposed study is intended to contribute to the associated literature. In this system, stored and newly added LOs that have plaintext content are visualized with images in an intelligent and novel method that combines deep learning and genetic algorithms. Using panoptic image segmentation (PIS), natural language processing (NLP), convolutional neural networks (CNNs), long short-term memory (LSTM) networks, and genetic algorithm (GAs), e-content can be efficiently visualized and enhanced. E-content generation has high costs and can lead to problems related to wasted time. Using panoptic-segmented image instances to create new images with a GA provides reusability. It is important to note that developing hybrid systems that combine deep learning and GAs provides an effective and interesting perspective on the image visualization problem. Thus, the educational

benefits of the e-content are increased. The proposed system contains different modules that use different artificial intelligence methods and differs from other studies in its GA-based image-creating module. This proposed system based on multi-subsystems (modules) is easy to design and offers an alternative approach to the content visualization problem.

Considering the general topic of the study, some noteworthy studies from the literature should be explained briefly as follows. Panoptic segmentation is the combination of instance segmentation and semantic segmentation. Its aim is to detect countable foreground objects ("things"; e.g., car, animal, person) and uncountable amorphous background regions ("stuff"; e.g., sky, grass) [53]. Many studies have explored the task of panoptic segmentation. Kirillov et al. [54] combined Mask R-CNN (instance segmentation method) with a feature pyramid network (FPN) (semantic segmentation method). They started with an FPN backbone used to extract rich multi-scale features. In Mask R-CNN, they used a region-based branch on top of the FPN, for instance segmentation. Then, a lightweight dense prediction branch was added on top of the same FPN for semantic segmentation. Thus, Mask R-CNN extended with FPN provides a fast and accurate baseline for both instance and semantic segmentation in a single network. In another study, Cai et al. [55] developed a panoptic segmentation system based on attention over the segmentation regions. They used image features based on the shape of the segmentation regions and generated captions based on the attention-weighted features to independently process things and stuff classes. A similar study that segmented both stuff and things created a semi-supervised panoptic segmentation system that uses a mixture of weak and fully labeled annotations [56]. Attention-guided unified network (AUNET) is another approach to panoptic segmentation that uses a region proposal network (RPN) and pixel-level attention [57]. In order to achieve more correct segmentation, object-level and pixel-level attention-based modules were used in AUNET. A proposal attention module (PAM) was used to detect foreground objects while a mask attention module (MAM) was used for background objects. Both thing and stuff segmentation accuracies were considerably improved by using different attention modules. AUNET was tested on the Mscoco and Cityscapes benchmark datasets, and the results showed that it was effective. In a study by De Geus et al. [58], a single-network method was used with the ResNet-50 feature extraction model for panoptic segmentation. They used Mask R-CNN and a pyramid pooling module to generate regions for potential objects to predict correct pixel classes. Occlusion-aware network (OANET) was introduced by Liu et al. [59] as a novel method that uses a heuristic approach to combine instance and semantic segmentation models to overcome the problem of overlapping between objects. They also used a spatial ranking module for the occlusion problem between the predicted instances. De Geus et al. [60] developed the end-to-end fast panoptic segmentation network (FPSNET). FPSNET assigns an instance ID to each pixel, rather than mask predictions and rule-based classification. The speed of this method was effectively tested on the Cityscapes and Pascal Voc datasets. Ocfusion is another system for solving overlapping object region problems by modeling instance masks with binary relations and testing them with the ground truth relations derived from the existing benchmark datasets [61]. Mohan and Valada [62] developed an efficient panoptic segmentation (EfficientPS) system for autonomous robot operations. EfficientPS used two-way FPN for panoptic fusion of a task-specific instance and semantic segmentation modules without any parameters.

NLP—one of the popular applications of artificial intelligence—is the analysis, understanding, and reconstruction of language according to certain rules [63]. NLP operations mainly include text preprocessing, morphological analysis, syntactic analysis, and semantic analysis [64]. Morphological analysis includes the processes of finding roots by separating the suffixes of words. In this way, what part of speech a word is can be determined [65]. In syntactic analysis, the usage purposes of the words in a sentence—such as subject, object, and adverb—are determined. The use of these words in different places and numbers in sentences may have different meanings. Semantic analysis, on the other hand, examines the association of discrete words with appropriate objects [65]. After these operations, text normalization is performed. To do so, NLP tools such as NLTK [66] can be used to convert capital letters to lowercase, convert numbers to text, remove punctuation and markup, remove spaces, open abbreviations, and remove unnecessary words [67].

There are many artificial intelligence methods in computer science for evolutionary optimization problems [68]. Evolution strategies, GAs, and programming are well-known artificial intelligence methods based on evolution. GAs seek to solve difficult and complex problems in many subject areas using defined mathematical models and functions. They use encoded parameters to search for and optimize solutions for a related problem [69]. Using a GA enables the possible solutions to a problem in a solution space to be obtained without being restricted to a local maximum or minimum. If a problem is difficult and complex and the solution space is very big, a classic search approach increases search time at the cost of performance. Fitness function, selection, reproduction, and mutation are the main processes of the GA. The GA searches for the best solutions in a defined population for the problem area with the following basic steps [70]: defining the initial

population in the solution space, calculating each chromosome as a solution candidate by fitness function, selecting the best chromosomes according to the fitness function, crossing over and mutating chromosomes, and (if the best solution is found or the terminal condition is provided) stopping the algorithm.

Deep learning algorithms are used frequently in many areas, including image processing, classification, and NLP. These methods—so-called deep learning networks—differ from classical artificial neural networks in several ways, including the application of layer numbers [71]. CNNs are one of the most well-known deep learning algorithms. Although a CNN is a feedforward neural network, it also contains convolutional, pooling, and fully connected layers. CNN is mostly used for image processing to reduce image size and extract and classify image attributes [72]. Another frequently used deep learning algorithm is recurrent neural networks (RNNs). RNNs relate the value calculated in the previous output to the current input values [73]. RNNs can process input sequences of arbitrary length and time series problems. In the training phase of long-sequence RNN problems, the gradient vector component may grow or decay [74]. This can cause gradient vanishing problems as well as learning problems with respect to finding the correct relations in the sequences of the RNN model. To overcome this issue, LSTM networks—a specific version of a classical RNN—were developed.

The rest of the paper is structured as follows. Section 2 describes the methods used in the proposed system. Section 3 details the application and evaluation of the proposed approach for content visualization. The final section presents the conclusion of the paper.

2 Material and methods

This study's methods were selected based on the objective, motivations, background, and research effort. In particular, deep learning was used for PIS and a GA was used to generate images from segmented images. Figure 1 represents the stages within the flow of the introduced approach. When there are many plaintext LOs in LORs or other content repositories, content visualization is required to ensure the best comprehension of the given information. The proposed method consists of many subsystems (modules). In our repository, there are many kinds of LOs, some of which are based on plaintext and some of which contain images. As described in the literature, LOs can be split into smaller LOs or vice versa. Segmented image instances are used to compose images that represent a corresponding plaintext in the LO. Thus, the LO can be visualized by an intelligent and automatic method.

2.1 Mscoco dataset

Mscoco is a large-scale dataset used for object detection, image segmentation, and image-captioning tasks. This dataset contains 330,000 images, 1.5 million object instances, 80 object categories, 91 stuff categories, and five captions per image [75]. Given the novelty of panoptic segmentation, there are few datasets with detailed panoptic annotations and public evaluation metrics. Mscoco [76] proved the most suitable and challenging for the new panoptic segmentation task, given its detailed annotations and high data complexity. It consists of 115,000 training images, 5,000 validation images, and 20,000 testing images [75] for the panoptic segmentation task. Mscoco's panoptic annotations include 80 thing categories and 53 stuff categories. In this study, 100,000 training images, 5,000 validation images, and 10,000 testing images from the Mscoco dataset were used. In addition, 10 extra data per thing category (a total of 800 images and annotations) were added to the dataset.

2.2 Evaluation metrics

The Panoptic Quality (PQ) metric is used to evaluate the performance of the panoptic model [57]. PQ was calculated for each class independently and average over classes to match unique splits which the predicted and ground truth segments into three sets: Matched pairs of segments were true positives (TP), unmatched predicted segments were false positives (FP), and unmatched ground truth segments were false negatives (FN) [53]. By using these three sets, PQ was defined as [53]:

$$PQ = \frac{\sum_{(p,g)\in TP} IoU(p,g)}{|TP| + \frac{1}{2}|FP| + \frac{1}{2}|FN|} \quad (1)$$

PQ was intuitive after inspection: $\frac{1}{|TP|}\sum_{(p,g)\in TP} IoU(p,g)$ was simply the average IoU of matched segments, while $\frac{1}{2}|FP| + \frac{1}{2}|FN|$ was added to the denominator to penalize segments without matches. Note that all segments receive equal importance regardless of their area. IoU(p, g) represents the intersection over union between predicted object p and ground truth g, true positive (TP) the matched pairs of segments (IoU(p, g) > 0.5), false positive (FP) unmatched predicted segments, and false negative (FN) unmatched ground truth segments [57]. Segmentation Quality (SQ) evaluates how closely matched segments are with their ground truths. When this value approaches 1, it means that the TP predicted segments are more closely matched with their ground truths. However, it does not take into account any of the bad predictions, which is the point at which the Recognition Quality (RQ) becomes relevant. This metric is a combination of

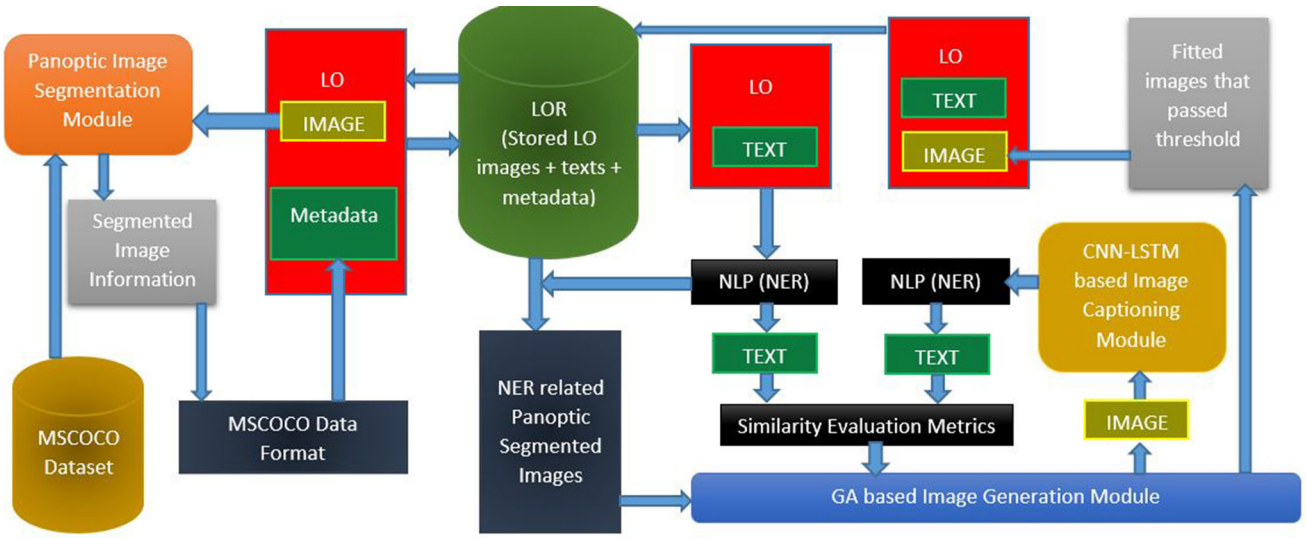

Fig. 1 Schema of proposed automatic and intelligent content visualization system

precision and recall and attempts to identify how effectively the model can make a correct prediction. PQ measures the performance of all classes with the same simple formula.

Texts created or translated using NLP methods should be evaluated. One frequently used method for doing so is BLEU-n [77]. In this method, the similarity between the desired reference text and the produced or translated text to be tested is examined. If the similarity between the texts is close, the value converges to 1, and if the similarity is distant, it approaches 0. It is calculated according to the frequency of the *n*-grams between the reference text and the tested text. Another metric is METEOR [78], which is defined as the harmonic mean of the precision and recall of unigram matches between sentences. It is used for synonyms and paraphrase matching. METEOR is better than BLEU on recall evaluation and when confronted with a lack of explicit word matching. *n*-gram-based measures work reasonably well when there is a significant overlap between the reference and candidate sentence [78]. CIDEr-D is another special metric designed for image-captioning evaluation to measure the similarity between a candidate image description and the reference sentences [79]. This method uses initial stemming to apply each sentence, represented with a set of 1–4 *n*-grams. Then, the co-occurrences of *n*-grams in the reference sentences and candidate sentence are calculated.

2.3 Panoptic image segmentation module

The first module of our proposed approach is the PIS module, which detects instances in all images in the LOs as well as images that belong to newly added LOs. The simple panoptic segmentation algorithm is as follows [53]. When a set of L semantic classes encoded by $\mathcal{L} := \{0, \ldots, L-1\}$ is given, the panoptic segmentation method attempts to map each pixel i of an image to a pair$(l_i, z_i) \in \mathcal{L}x\mathbb{N}$, where l_i represents the semantic class of pixel i and z_i represents its instance ID [54]. The semantic label set consists of subsets $\mathcal{L}^{St}$ for stuff and $\mathcal{L}^{Th}$ for things, where $\mathcal{L} = \mathcal{L}^{St} \cup \mathcal{L}^{Th}$ and$\mathcal{L}^{St} \cap \mathcal{L}^{Th} = \varnothing$. When a pixel is labeled with$l_i \in \mathcal{L}^{St}$, its corresponding instance ID z_i is irrelevant. Thus, all pixels belong to the same instance, called stuff. On the contrary, all pixels with the same assignment, where $l_i \in \mathcal{L}^{Th}$ belong to the same instance, are called things. In our study, Li et al. [57] AUNET was modified and used to perform the panoptic segmentation of the LO images. AUNET uses an RPN and pixel-level attention. In order to achieve more correct segmentation, object-level and pixel-level attention-based modules were used in AUNET. PAM was used to detect foreground objects while MAM was used for background objects. Consequently, both thing and stuff segmentation accuracies are considerably improved by using different aiming modules. When an image is given to the PIS module, thing and stuff instances are detected and segmented. Moreover, the system generates Mscoco data-formatted information (Fig. 2) [75].

For example, when images that include apples are given to the panoptic segmentation module, detected and segmented apple instances are masked with different colors (Fig. 3). The generated descriptive texts (annotation) for these apple instances by the panoptic segmentation module are as follows: “a group of apples on a tree in an orchard” for Fig. 3a, “a green apple is surrounded by a group of bananas” for Fig. 3b, and “a bowl of bread next to an orange and apple and a glass of orange juice” for Fig. 3c.

```
annotation{
   "image_id"          : int,
   "file_name"         : str,
   "segments_info"     : [segment_info],
}

segment_info{
   "id"                : int,
   "category_id"       : int,
   "area"              : int,
   "bbox"              : [x,y,width,height],
   "iscrowd"           : 0 or 1,
}

categories[{
   "id"                : int,
   "name"              : str,
   "supercategory"     : str,
   "isthing"           : 0 or 1,
   "color"             : [R,G,B],
}]
```

Fig. 2 Mscoco data format for panoptic segmentation [75]

Detected and segmented panoptic thing and stuff image instances and their Mscoco data-formatted information are stored in LOR with their LO_ID, image_id, detected instance pixels (segment_info), and detected instance descriptive texts (annotation).

2.4 Genetic algorithm-based image generation module

The second module, which is the core of the proposed approach, is the GA-based image generation module. To produce an image that best suits the text according to the searched text (plaintext of LO), stored segmented image instances containing the same named entity recognition (NER) in their descriptive metadata are gathered. NLTK was used to create NERs of the given LO text that needed to be visualized with images. These image instances are combined to produce an image with GA. For this purpose, the segmented image pixels are cropped from the related image in the LO. This cropped image instance object is then exposed to the processes of resizing, rotating, relocating, flipping (y-axes), and z-indexing. The value of these processes on the image are encoded into the GA solution chromosomes (Fig. 4). Each produced image is exposed to a CNN-LSTM-based encoder–decoder module to produce text description. The produced text is analyzed with NLP techniques to produce NERs. Then, two texts consisting of NERs (the first being the searched reference LO text and the second the generated image-descriptive text) are compared using text similarity methods (BLEU-n) during the GA fitness evaluation.

In the GA module, SC_1, $SC_2 \ldots SC_n$ are candidate solution chromosomes in the search area of the image generation problem. Each gene corresponds to an image property. SC_iG_1 to SC_nG_5 are encoded genes of the chromosome by using direct encoding as integer values, where i is the index of the chromosome and n is the chromosome count in the population. SC_iG_1 represents resized cropped segmented image instance values. The lower boundary of resizing is 32×24, and the upper boundary is 320×240 (half of the corresponding system output, 640×480). SC_iG_2 represents rotated cropped segmented image instance values. Rotation values change from 0° to 360° in 20° increments. SC_iG_3 represents relocated cropped segmented image instance values. Image instances are relocated in 640×480 image boundaries. SC_iG_4 represents flipped cropped segmented image instance values according to y-axes. If the image instance is flipped, the value of this gene is 1, otherwise 0. The last gene, SC_iG_5, represents the z-indexing values of cropped segmented image instances. If the image instance is at the back, the value of this gene is -1. If the image instance is at the same level, the value of this gene is 0. If the image instance is in the front, the value of this gene is 1.

After these processes, the produced image is provided to the CNN-LSTM (encoder–decoder)-based image-captioning module to generate caption y, which is encoded as a sequence of 1 to K encoded words:

(a) (b) (c)

Fig. 3 Detected and segmented apple instances with colorful masks [75]

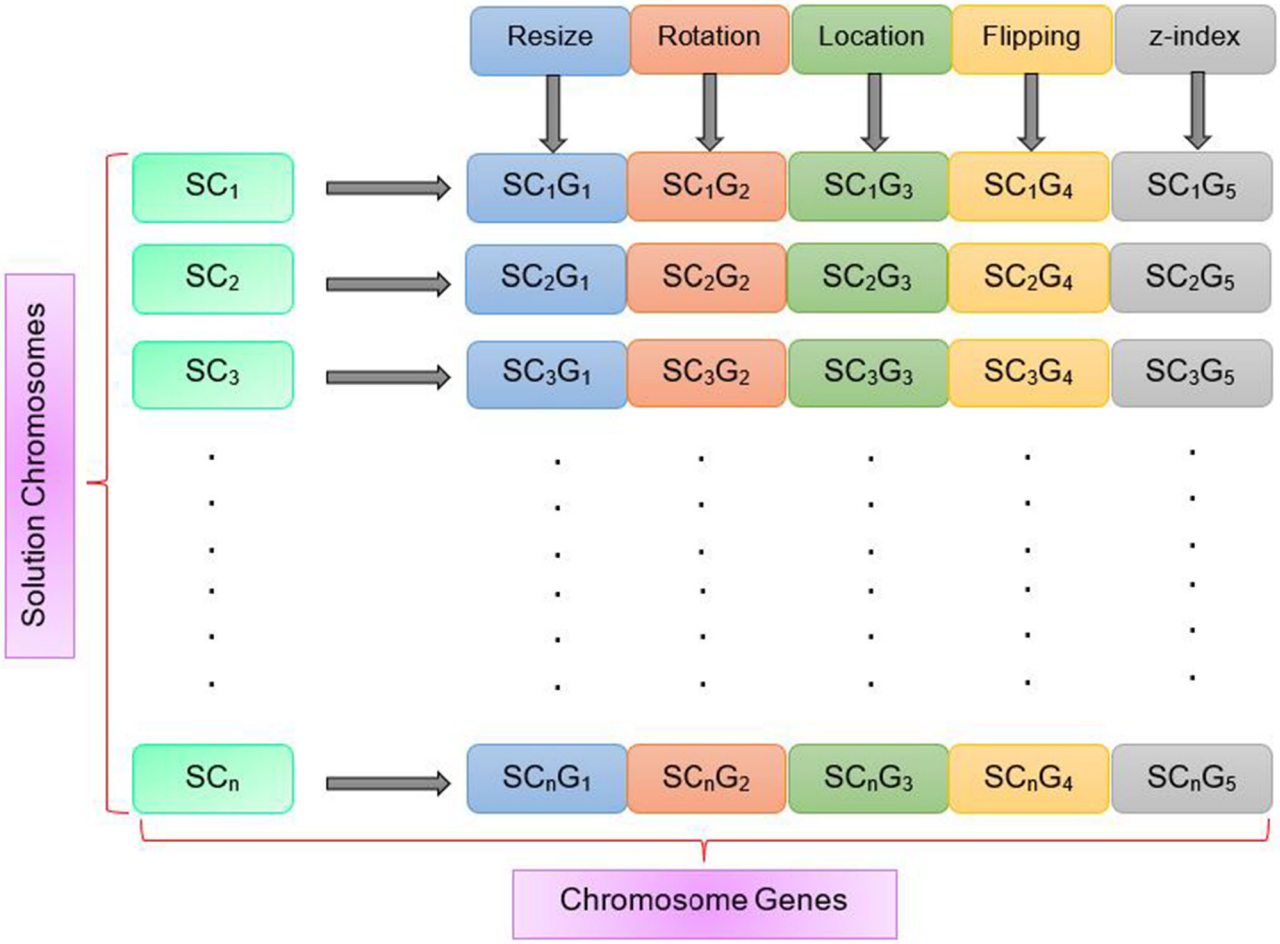

Fig. 4 Encoded image processing property chromosomes

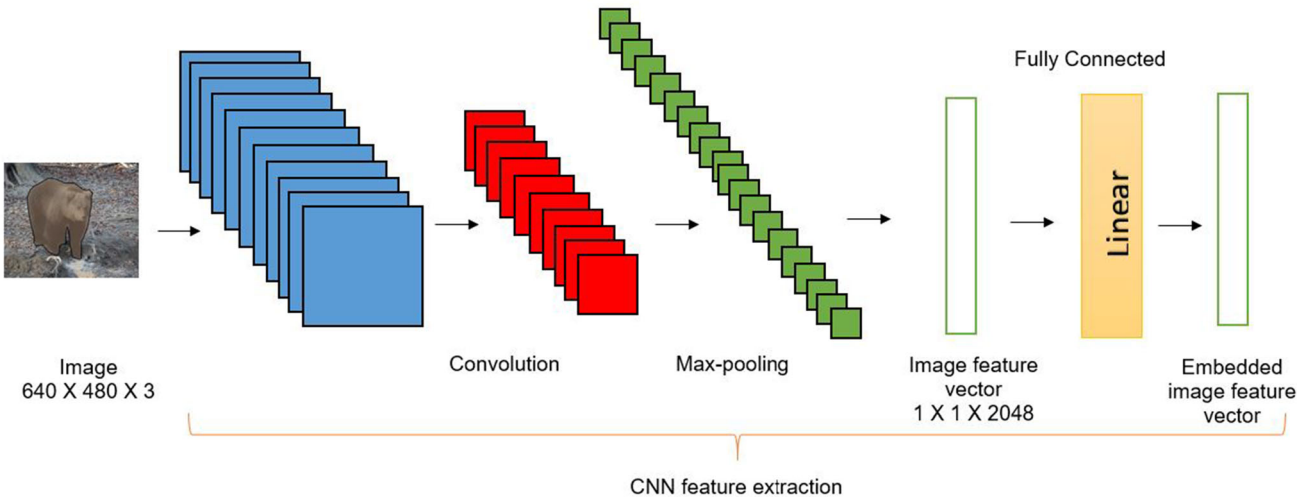

Fig. 5 CNN-based feature extraction structure used in the proposed study

$$y = \{y_1, y_2, \ldots y_C\}, y_i \in \mathbb{R}^K \quad (2)$$

where K is the size of the vocabulary and C is the length of the caption. A CNN is used to create image feature vectors (Fig. 5). Encoder CNN produces L vectors, each of which is a D-dimensional representation of the input image.

Encoded vectors are decoded with an LSTM network to generate word-by-word captions (Fig. 6). The LSTM network's success on long sequences is based on its memory cell [80]. The memory cell (c) can preserve the state over long periods of time and consists of an input gate (i) that decides whether input data should be prevented or conveyed to the memory cell, an output gate (o) to produce or prevent an output itself by recurrent connections in two consecutive time steps (t), and finally a forget gate (f) that decides whether to recall or omit the previous cell (Fig. 7).

Sigmoid gates are used to control the read and write process of the memory cell. At a given time step t, the LSTM network receives inputs from various sources: the current input x_t, the previous hidden state of all LSTM units h_{t-1}, and the previous memory cell state c_{t-1}. These gates are updated at time step t for given inputs x_t, h_{t-1}, and c_{t-1} as follows [80]:

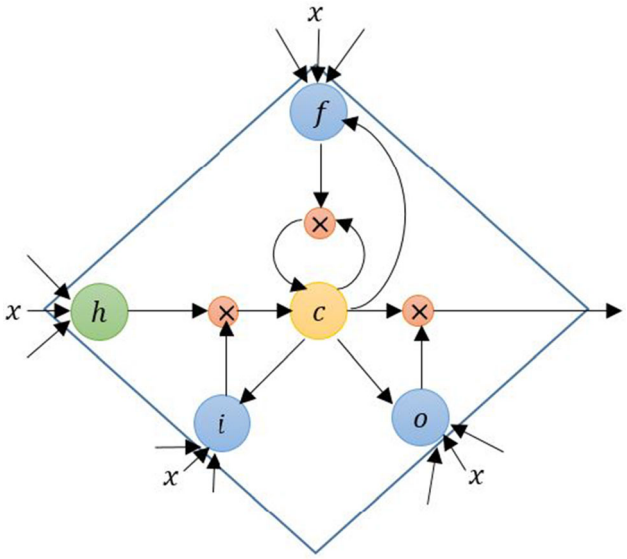

Fig. 7 LSTM memory cell [80]

$$i_t = \sigma(W_{xi}x_t + W_{hi}h_{t-1} + b_i) \quad (3)$$

$$f_t = \sigma\left(W_{xf}x_t + W_{hf}h_{t-1} + b_f\right) \quad (4)$$

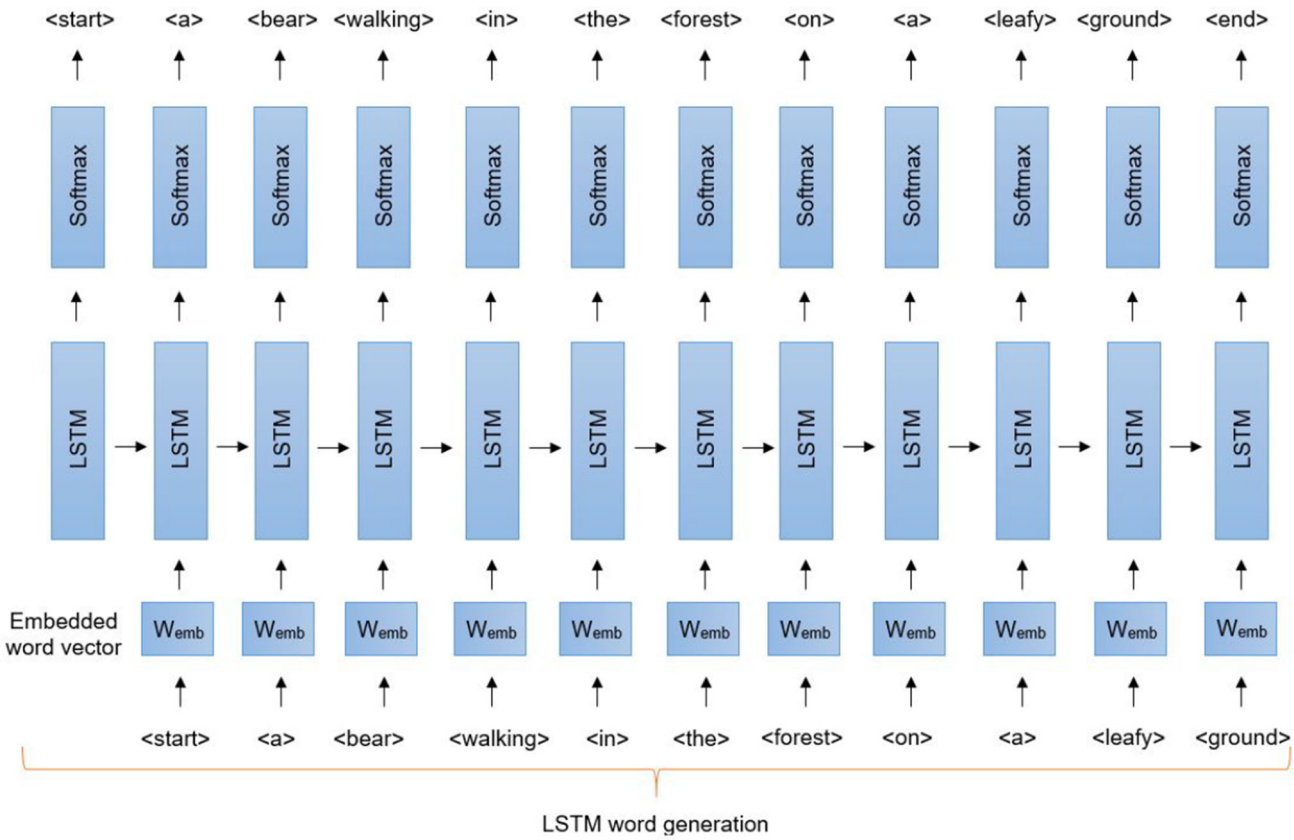

Fig. 6 LSTM-based text (caption) generation used in the proposed system

$$o_t = \sigma(W_{oi}x_t + W_{ho}h_{t-1} + b_o) \quad (5)$$

$$g_t = \phi(W_{xc}x_t + W_{hc}h_{t-1} + b_c) \quad (6)$$

$$c_t = f_t \odot c_{t-1} + i_t \odot g_t \quad (7)$$

$$h_t = o_t \odot \phi(c_t) \quad (8)$$

where W are the weight matrices learned from the network and b are the bias vectors. The sigmoid activation function is represented by σ. Thus, $\sigma(x) = 1/(1 + \exp(-x))$ and ϕ present a hyperbolic tangent $\phi(x) = (\exp(x) - \exp(-x))$. $\odot$ points to the dot products of the vectors. At $t = 0$, the input data are sigmoid-modulated to input gate $i(t)$, where values lie within [0, 1]. In this step, the values of the forget gates $f(t)$ of the different LSTM units are 0. Along with increasing the time step, the forget gate begins to decide which unimportant information should be forgotten and, meanwhile, retaining information that is deemed useful. The memory cell states $c(t)$ and output gate $o(t)$ then gradually absorb the valuable context information over time and make a rich representation $h(t)$ of the output data [80]. The hidden output with K possible outcomes $(h_t = \{h_{tk}\}_{k=0}^{K}, h_t \in R^K)$ is used to predict the next word using the softmax function with parameters W_s and b_s [80]:

$$F(p_{ti}; W_s, b_s) = \frac{exp(W_s h_{ti} + b_s)}{\sum_{j=1}^{K} exp(W_s h_{tj} + b_s)} \quad (9)$$

where p_{ti} is the estimated word probability)[80]. In this way, at the end of the time steps, the image caption is generated as text. The generated image caption (text) is then analyzed with NLTK to obtain the NERs, which are compared for text similarity with the NERs of the plaintext LO (that is to be visualized). For this purpose, the fitness function $F(x)$ of the chromosomes in the population is calculated according to the selected similarity metrics: BLEU-n, METEOR, and CIDEr-D (Eq. 10). In this study, each similarity metric was tested on the same parameters. For this purpose, mscocoapi [81], which can be downloaded from GitHub, was used. The best chromosomes (i.e., that maximized similarity) are then selected for the new population by using roulette wheel selection. Meanwhile, new chromosomes are regenerated and mutated for the new population.

$$F(x) = \max(Sim_{sm}(GT_i)) \quad (10)$$

where sm is the similarity metrics (BLEU-n, METEOR, CIDEr-D) used to evaluate similarity and GT_i is the generated text of the corresponding ith generated image in the population.

When the GA is stopped, the images that pass a threshold similarity value (0.5 for BLEU-1) are listed as the best solutions and sorted by fitness values (similarity values) from largest to smallest. The image with the highest value is the first in the list and is used to visualize the image from the plaintext. This image is then stored with the corresponding text-based LO in the LOR.

3 Application and evaluation

This section provides information regarding the application side of the proposed method. It is important to apply such a method to a real use case and run the necessary evaluation parameters to better understand its performance.

3.1 Application on panoptic segmentation and image generation

On the application side of this study, Python was used for all modules in the proposed system. A computer system running an Intel Xeon e5-2620 6c/12t 2.00 GHz CPU with 24 GB RAM was employed for the test application.

In the PIS module, AUNET was used with pre-trained ResNet-50, ResNet-101, and ResNet-152 models. Mask R-CNN was used for foreground object detection. The ResNet-50, ResNet-101, and ResNet-152-based networks were optimized using stochastic gradient descent (SGD) with a weight decay of 4e-5, a momentum of 0.9, and a batch size of 100. The learning rate was initialized with 0.002 for extra data added Mscoco dataset with 100 epochs.

In the GA module, the population size was 1,000, the crossover rate was 0.7, the mutation rate was 0.25, the elitism rate (the ratio of elitist chromosomes that is kept into the next generation) was 0.1, the maximum iteration count was 300, and roulette wheel selection was used for selecting potentially useful solutions for recombination. Also, single-point crossover and random resetting gene mutation were used in the GA. The elitism means the best string seen up to the current generation which is preserved in a location either inside or outside the population and it can be introduced in various ways [82]. For example, the best solution of initial population can be preserved in a separate location and then compare it with the best result of the new generation and replace it if it is better. In some other studies, some of the top fitted chromosomes are transferred directly into the next generations or the best of the i^{th} generation can be compared with the worst of the $(i + 1)$th generation and replace it if it is better [82]. The elitist strategy compensates the defects of easy loss of good genes and it can rapidly increase the performance of GA by not losing the best-found solutions [83]. Similarly, in this study, the elitism rate is equal to the fraction of best individuals in the current population that are copied into the new generation directly before the rest of the population is generated. In the CNN-LSTM-based image-

captioning module, image-captioning library was used [84]. In this module, CNN was used to create an encoded image feature vector (encoder), while an LSTM network was used for word-by-word text generation for the encoded input vector (decoder). The CNN consists of an image input layer, a convolutional layer, a max pooling layer, and a fully connected layer. In the proposed system, produced input images are 640 × 480 × 3. These numbers indicate width, height, and RGB, respectively. In the convolutional layer, a 5 × 5 filter is used for feature mapping. The max pooling layer has a 3 × 3 filter to decrease the number of parameters and help avoid overfitting. The last layer is the fully connected layer, which is used to gather all image features for classification. The CNN is optimized using SGD with an initial learning rate of 0.001 for 100 epochs. Pre-trained ResNet-50, ResNet-101, and ResNet-152 were used as the CNN model to extract image features. The image feature vector was the mean output of the last convolutional layer of the CNN and thus had a dimension of 2048.

The decoder LSTM network uses 512 units in each layer. The Adam [85] optimizer was used with a weight decay of 4e-5, a momentum of 0.9, and a batch size of 100. The learning rate is 0.001. After each LSTM layer, a dropout layer with probability 0.2 was added. The probability of 0.2 means that one in five inputs will be randomly excluded from each cycle of the updating process. Dropout is a regularization technique used to prevent overfitting. In short, this technique randomly selects neurons to be ignored during training. When a neuron is dropped out, it cannot contribute to the network and receive some values in the backpropagation process. The next layer was the dense layer, which used the total number of words in the vocabulary as the dimension. The last layer used softmax as the activation function. The output of this function is modeled as the probability distribution over K possible outcomes. Specifically, it shows the probability that each word in the vocabulary is the most suitable output for the specific element in the output vector.

3.2 Testing and evaluation on generated images

In the PIS module of the proposed system, AUNET is used with pre-trained ResNet-50-FPN, ResNet-101-FPN, and ResNet-152-FPN backbones based on extra data added to the Mscoco dataset. In this study, 100,000 training images, 5000 validation images, and 10,000 testing images from the Mscoco dataset were used. In addition, (a total of 800 images and annotations) were added to the dataset. These images were added 10 to each of the 80 object categories and used to test whether the system could distinguish them when new data were added to the existing dataset during the testing phase. The quality of segmentation results in the percentages shown in Table 1. The best (i.e., highest) value was obtained with the ResNet-152-FPN backbone.

Table 1 Panoptic quality results of backbones used

Backbone	PQ (%)	SQ (%)	RQ (%)
ResNet-50-FPN	25.3	69.7	30.4
ResNet-101-FPN	40.1	77.2	49.2
ResNet-152-FPN	43.7	79.4	53.7

Different amounts of text-based LOs and science topics were used to test the efficiency and validity of the proposed content visualization system (Table 2).

In the GA-based image caption module, LO texts are analyzed with NLP to obtain the NERs. Corresponding segmented image instances for detected NERs are obtained from LOR. The GA then attempts to combine these instances to create an image, and each image is exposed to the CNN-LSTM-based image-captioning module, which uses ResNet-50, ResNet-101, and ResNet-152 backbones with extra data added to the Mscoco dataset. The generated image captions are analyzed with NLP to represent text, which is then compared using the BLEU-N, METEOR, and CIDEr-D similarity metrics to evaluate the fitness of the GA chromosomes in a range of 0 to 100 (Table 3). Images that pass the threshold value of 0.5 (percent equivalent = 50) for the BLEU-1 metric are listed from highest to lowest. The first image, which has the highest (i.e., best) value, is selected as the corresponding visualization image of the text-based LO.

The CNN-LSTM-based produced captions (texts) were best for the biology and foreign language topics, as the Mscoco dataset's thing category instances were most suitable for these topics. The worst results were obtained for geography, which is more closely related to the stuff category, for which there are few instances in the dataset. If the Mscoco dataset is enhanced with different types of thing and stuff category instances for the panoptic segmentation task, better results can be obtained using our proposed system. In addition, the success of the system is

Table 2 Text-based LO count according to science topic

Topic	Text-based LO count
Foreign language	20
History–arts	10
Biology	26
Math	20
Physics–chemistry	14
Geography	10

Table 3 Evaluation metrics of captions for generated images in GA module

Backbone	Topic	BLEU-1	BLEU-2	BLEU-3	BLEU-4	METEOR	CIDEr-D
ResNet-50	Foreign language	61.2	38.7	26.5	17.4	16.9	53.8
	History–arts	53.4	33.8	23	15.2	14.7	47
	Biology	63	39.8	27.2	17.8	17.2	55.1
	Math	58.7	37.1	25.4	16.7	16.2	51.6
	Physics–chemistry	56.3	35.6	24.3	16	15.5	49.5
	Geography	50.3	31.8	21.6	14.3	13.8	44.3
ResNet-101	Foreign language	66.4	40.2	28.1	19.2	18.1	58.7
	History–arts	55.7	33.6	23.5	16.1	15.2	49.3
	Biology	68.4	41.4	28.9	19.7	18.5	60.2
	Math	63.1	38.1	26.7	18.2	17.1	55.8
	Physics–chemistry	59.9	36.2	25.4	17.3	16.3	52.9
	Geography	51.1	30.9	21.6	14.7	13.9	45.2
ResNet-152	Foreign language	69.7	41.7	29.3	22.0	18.4	62.1
	History–arts	59.1	35.3	24.8	18.6	15.6	52.6
	Biology	71.9	43	30.2	22.6	18.9	63.7
	Math	66	39.5	27.7	20.8	17.4	58.8
	Physics–chemistry	63.1	37.8	26.5	19.9	16.6	56.2
	Geography	53.8	32.2	22.6	16.9	14.2	47.9

related to the LO texts, which should contain suitable NERs for the corresponding segmented images. Using the proposed system, the same segmented image instances can be used for different text-based LOs and for different fields of science. For example, a segmented dog image can be used to visualize texts in foreign language for the phrase "The dog is playing ball," in biology for "Dogs have four legs," and in math for "How many dogs are in the garden?" Therefore, the reusability of the LOs is ensured.

Furthermore, to examine the efficiency and validity of the proposed approach with regard to the human factor, 250 text-based LOs were selected randomly according to the proportion of each science field in Table 2. The distribution of the total 250 generated images for text-based LOs is shown in Table 4.

The generated 250 images were shown to seven instructors (I1 to I7) and 20 students (S1 to S20) at the secondary school level, who labeled the images as usable or not usable for the defined text-based LO. Next, we calculated how many images were labeled as usable and how many images labeled as unusable according to instructors' (Table 5) and students' (Table 6) labeling.

The results are similar to the proposed system with regard to proportion and ranking of values. The categories with generated images that instructors labeled as usable were, from highest percentage to lowest, BIO, FLA, MAT, PHY, HIS, and GEO. The categories with generated images that students labeled as usable were, from highest percentage to lowest, BIO, HIS, PHY, FLA, MAT, and GEO. The usable and not usable image counts as labeled by instructors and students are given as Online Resource 1 for each topic.

Table 4 Distribution of text-based LO count according to science topic for human evaluation

Topic	Text-based LO count
Foreign language (FLA)	50
History–arts (HIS)	25
Biology (BIO)	65
Math (MAT)	50
Physics–chemistry (PHY)	35
Geography (GEO)	25

Table 5 Comparison of usable images for different topics according to instructors

Instructors	FLA	HIS	BIO	MAT	PHY	GEO
I1	43	19	58	41	26	18
I2	48	23	64	43	30	18
I3	46	24	62	44	29	22
I4	50	21	61	45	32	18
I5	46	20	63	47	32	20
I6	47	23	62	42	31	20
I7	46	22	63	43	33	17
Mean	46.57	21,71	61.85	43,57	30,43	19
Percentage (%)	93.14	86.85	95.16	87.14	86.94	76

Table 6 Comparison of usable images for different topics according to students

Students	FLA	HIS	BIO	MAT	PHY	GEO
S1	39	19	49	36	27	15
S2	47	23	62	42	29	16
S3	41	24	59	40	31	19
S4	45	23	56	44	32	18
S5	46	22	60	42	31	17
S6	41	23	60	43	32	17
S7	44	22	62	41	33	16
S8	45	23	61	39	33	19
S9	46	22	61	47	31	16
S10	45	24	59	45	32	16
S11	41	21	63	45	30	19
S12	45	22	60	40	33	17
S13	47	23	60	39	30	15
S14	43	24	58	43	34	19
S15	48	24	59	43	32	20
S16	42	21	59	41	30	19
S17	46	25	62	45	31	15
S18	40	22	61	42	32	19
S19	42	23	57	44	33	16
S20	37	19	55	35	25	15
Mean	43.5	22.45	59.15	41.8	31.05	17.15
Percentage (%)	87	89.8	91	83.6	88.71	68.6

For an example in the field of biology, if the LO text is "Bears live in the forest," NLP detects "bear" and "forest" NERs (Fig. 8). Next, input segmented image instances for "bear" in the thing category and "forest" in the stuff category are taken from the LOR by searching the LO metadata. In the first chromosome SC_1, resize is 300×200, rotation is 60°, location coordinates are (50,40), flipping value is 0 and z-index is 1. After iterations, when the GA stopped, output image is obtained. In the corresponding last chromosome SC_n of the output image, resize is 135×90, rotation is 0°, location coordinates are (65,90), flipping value is 0 and z-index is 1 (Fig. 8). The related images are then combined according to the fitness value produced by the GA module to create an image (Fig. 9).

Another example can be provided for the foreign language topic. If the LO text is "There is an apple on the table," NLP detects "apple" and "table" NERs (Fig. 10). Next, segmented image instances for "apple" and "table" in the "thing" category are taken from the LOR by searching the LO metadata. Related images are combined according to the fitness value produced by the GA module to create an image (Fig. 11).

In our proposed system, there is a service that works in the background of the content visualization system. This service is responsible for detecting stored plaintext-based LOs in the system. When a plaintext-based LO is detected, the GA-based image generation module attempts to visualize the corresponding text-based LO by using panoptic-segmented and stored image instances. This service runs automatically on an everyday basis as well as when a new text-based LO is added to the system. Moreover, the user can start this service manually whenever it is required. Thus, there is no time or space (memory) limitation for image generation. However, further analyses were investigated by means of the performance and time complexity of the GA. Generally, metaheuristic algorithms are compared empirically rather than using Big-O notation because such algorithms do not always guarantee that the global solution will be found within a given time period. For these reasons, in addition to the GA, different metaheuristic algorithms were tested, such as artificial bee colony (ABC), differential evolution (DE), and mutation-based particle swarm optimization (MPSO). For comparison, the same population size and maximum iteration count were defined for all algorithms (200 and 100, respectively). Other tuning parameters used were also optimized for all algorithms. For the GA, the crossover rate was 0.7, the mutation rate was 0.25 and elitism rate was 0.1, and roulette wheel selection was used for selecting potentially useful solutions for recombination. Also, single-point crossover and random resetting gene mutation were used in the GA [86]. For the ABC, the onlooker number was 0.5, the employed bee number was 0.5, and the scout number was 1 [87]. For the DE, the crossover factor was 0.8 and the scaling factor was 0.5 [88]. For the MPSO, the inertia weight was 0.1, the lower bound was 1, the upper bound was 2, the mutation rate (for not to be stagnated after finding a local minimum) was 0.1, and attraction terms were equal which c1 was 2 and c2 was 2 [89]. In addition, the same computer system configuration, same data (image generated from panoptic-segmented instances), same similarity method (BLEU-1), and same backbone model (ResNet-152) were used. For the biology example given in Fig. 8 above, best cost (BC) (Fig. 12) and computation time (CT) (Fig. 13) were calculated for all algorithms.

As shown in Fig. 12, BC was calculated for GA (0.280), then MPSO (0.315), DE (0.330), and ABC (0.342) consecutively for 100 iterations. The GA reached BC on iteration 87 and did not change through the end of the run. As shown in Fig. 13, DE required less CT per iteration as well as total time during 100 iterations. The other lowest CT was obtained with MPSO. The GA and ABC algorithms had the third and last CT, respectively. CT can be reduced with more powerful computer configurations. In our proposed system, there is no time limitation because a

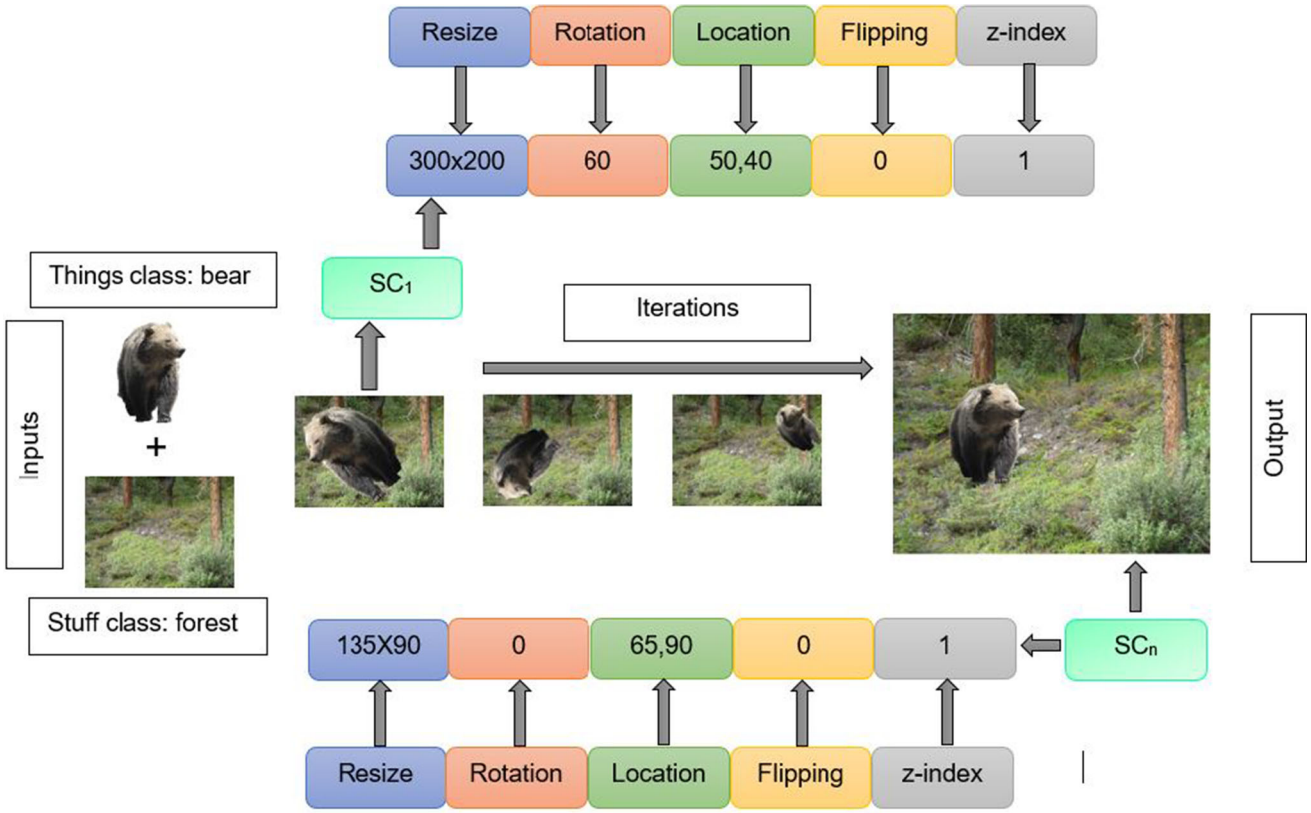

Fig. 8 Segmented image instance and generated image with GA module for "Bears live in the forest" text in biology topic

background service works to track plaintext Los and attempts to visualize them daily. If the user wants to produce an instant visualization by running the service manually, the GA produces an image within 5–8 s per iteration, and the total generation time based on total iteration count (100 in test) is nearly 671.304 s.

Moreover, to evaluate the effectiveness of the proposed study for several runs, further calculations were performed to compare the GA, ABC, DE, and MPSO algorithms in terms of CT and BC. In this process, each algorithm is run 100 times for 10 image generations (attempts to visualize text-based Los) per topic (FLA, HIS, BIO, MAT, PHY, and GEO). CT values are calculated for each topic. For example, for FLA-1 image generation, each algorithm is run 100 times, and the minimum, maximum, and mean values of these 100 runs are calculated using CT values (Online Resource 1). This process is performed for each topic and its corresponding image generations. Best values in terms of CT are calculated for the DE, MPSO, GA, and ABC algorithms, respectively. There are no significant differences among these algorithms in terms of CT values (Table 7). The same processes are also performed for BC values (Online Resource 1). Best values in terms of BC are calculated for the GA, MPSO, DE, and ABC algorithms, respectively. In addition, the Wilcoxon ranked-sum test [90] was used for comparison of the GA and other three algorithms (ABC, DE, and MPSO) in order to show significance (Table 7). The *p*-value was calculated to compare CT, ABC, and GA; CT, DE, and GA; and CT, MPSO, and GA, as well as BC, ABC, and GA; BC, DE, and GA; and BC, MPSO, and GA. In all calculations, *p*-values were well below the significance threshold ($\alpha = 0.05$), so the distributions were significantly different. A " + " sign was used to indicate when the GA yielded significantly better performance than the other algorithms (ABC, DE, and MPSO) with respect to CT and BC. A "-" sign was used to indicate when the GA yielded significantly worse performance than other compared algorithms (ABC, DE, and MPSO) with respect to CT and BC.

The mean values and *p*-values for each comparison with the GA are shown in Table 7, which indicates that there are some differences between these algorithms in terms of CT and BC values. In this study, there is no time problem, so any stochastic algorithm can be used for this process. However, BC is important because the fitness values of the solutions in these algorithms are related to BC. The lowest cost value means best fitness. Thus, for this proposed study (for several runs), the best BC values were obtained with GA. Thus, we chose the GA in the GA-based image

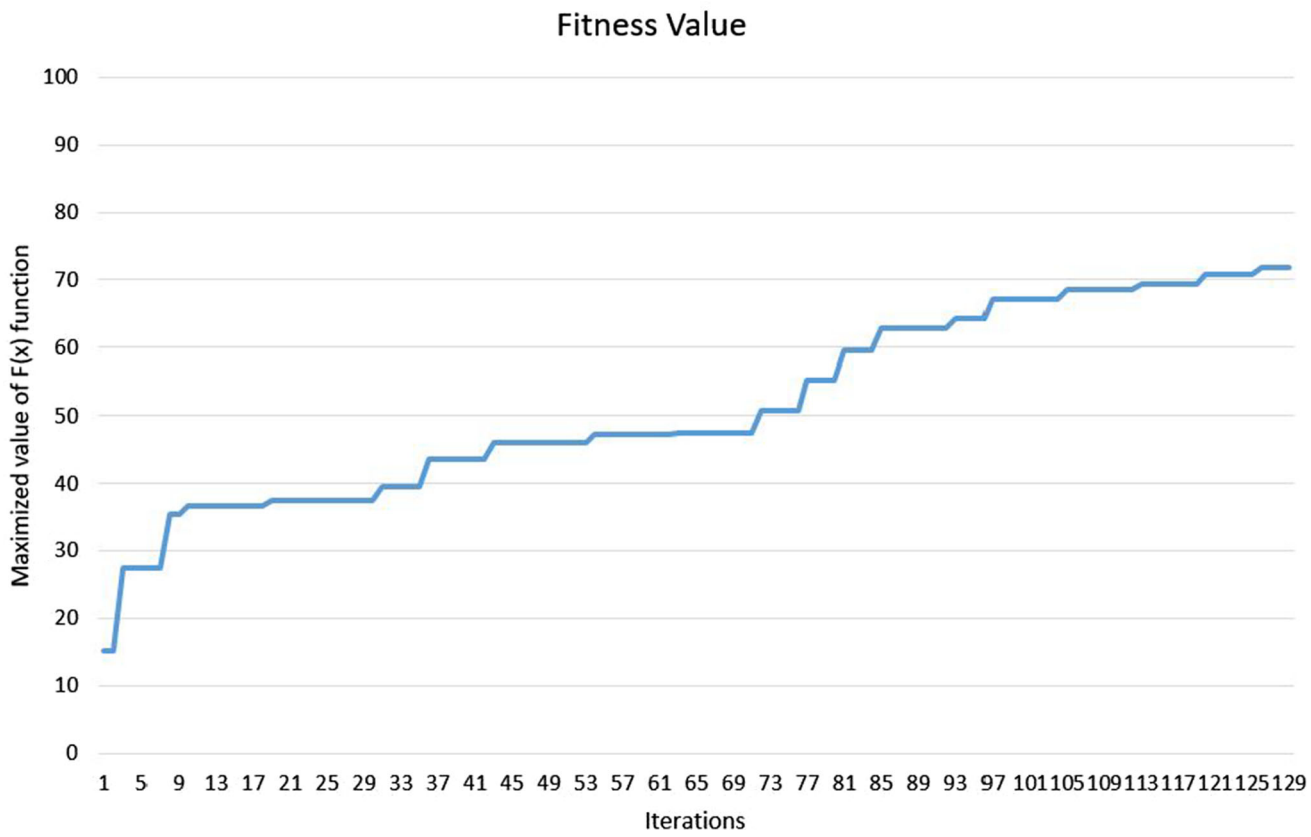

Fig. 9 GA fitness value for "Bears live in the forest" text in biology topic

Fig. 10 Segmented image instance and generated image with GA module for "There is an apple on the table" text in foreign language topic

generation module of our proposed system to generate images from segmented image instances.

The results obtained in our study lead to the following contributions:

- E-content generation has high costs and can lead to time-wasting problems. Using panoptic-segmented image instances to create new images using a GA provides reusability.
- Deep learning and GAs provide an effective and interesting perspective on the image visualization problem. Therefore, it is important to note that developing hybrid systems is a trendy topic that will likely continue to be popular in the future. Currently, deep learning and other artificial intelligence methods are the strongest members of such systems.
- Several studies have generated images, especially using generative adversarial networks, but no study has yet generated images from panoptically segmented image instances which are obtained from LOs. Thus, our proposed systems provided reusability of LOs.

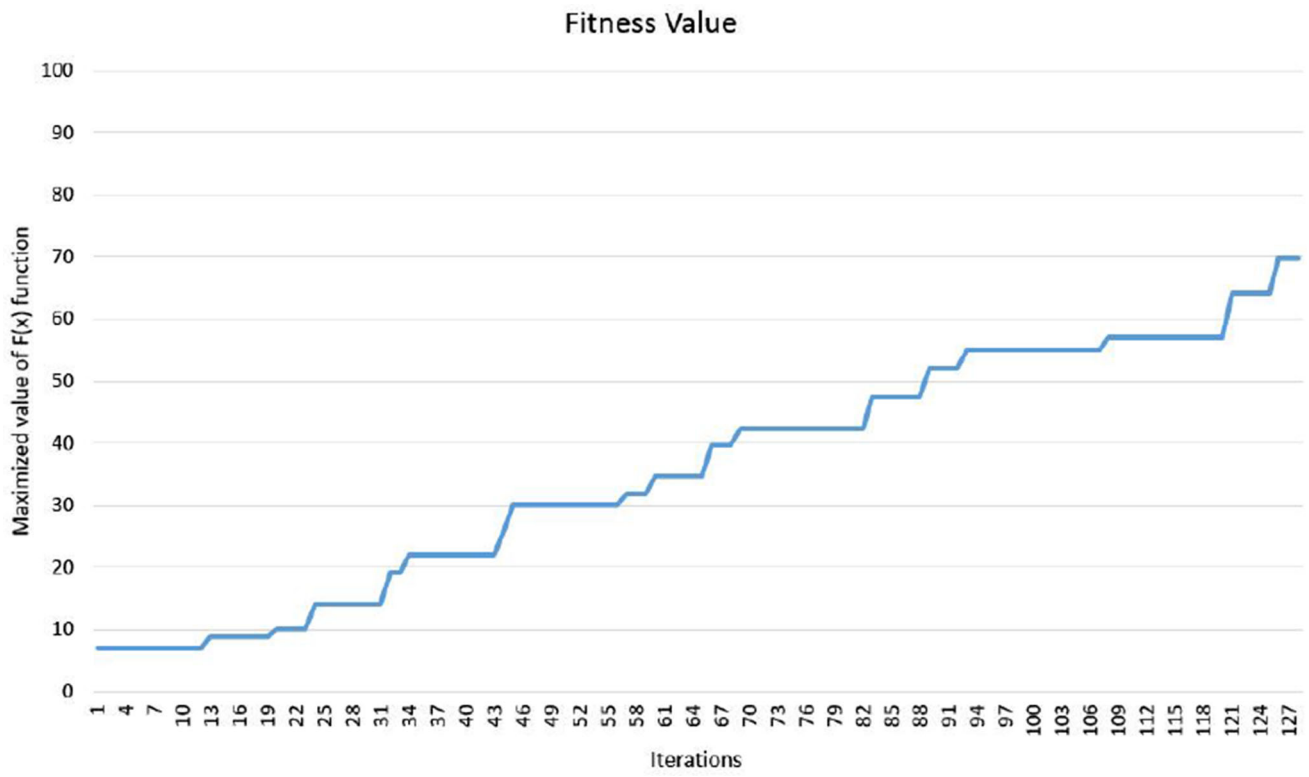

Fig. 11 GA fitness value for "There is an apple on the table" text in foreign language topic

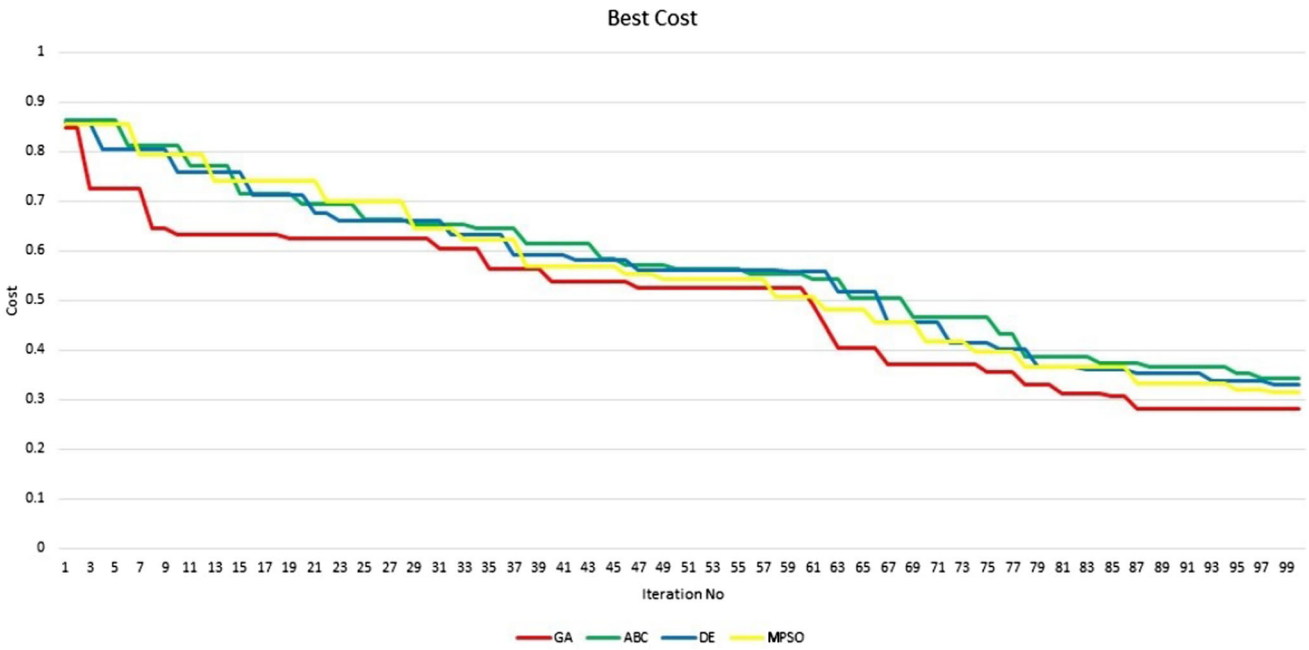

Fig. 12 Comparison of cost values of algorithms

There are also some gaps (limitations) that should be expressed as contributive suggestions for further studies by interested researchers. Briefly, these gaps are as follows:

- The method has been designed for an easy-to-use approach. The application parameters of the proposed methods in the system may be optimized in detail to further improve the system.

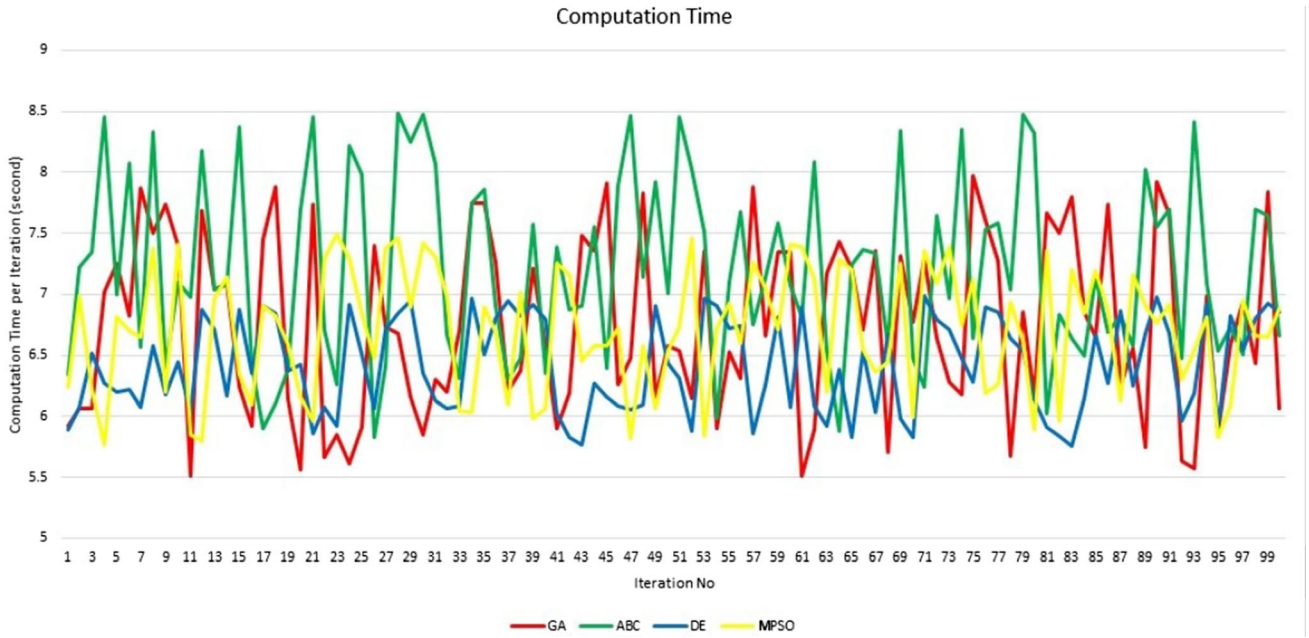

Fig. 13 Comparison of computation time per iteration for algorithms

- This study was focused only on the Mscoco database because it supports panoptic segmentation. However, for further comparison and evaluation, additional data (thing and stuff instances) may be added to this dataset.

In sum, using the proposed content visualization system, text-based LOs in the LOR can be visualized automatically and effectively for e-content production and other educational purposes, such as web-based learning systems, education portals, and learning management systems. In accordance with the importance of the visualization of texts in the given literature, the proposed system, enhanced with artificial intelligence, increases the quality of education and the level of learning. Moreover, generated images can be used not only for educational purposes and but also for other text-based digital repositories, advertisements, entertainment, and so on. For example, in daily life, if we want to visualize text while preparing a PowerPoint presentation, we can search the web for suitable images, which may take time. However, if the proposed automatic and intelligent content visualization system is used, slides in the presentations can be visualized in a novel and efficient method. For these reasons, this software is beneficial for education in terms of saving content generation time and decreasing costs.

4 Conclusion

The increasing demand for web-based and distance education has triggered new solutions for e-content generation, which is difficult and time consuming. LOs in the LOR can be used for this purpose, but most LOs are text-based. For educational purposes, using only text-based content may not be sufficiently effective. Using visualization techniques such as images attracts users by improving their concentration and also clarifies the given information with the content. For this reason, in this study, a novel and automatic intelligent content visualization system was developed. This large-scale system consists of subsystems (modules): an AUNET-based PIS module, a GA-based image generation module, and a CNN-LSTM-based image-captioning system. In the first module, LO images are exposed to panoptic segmentation, and the detected instances are stored in the LOR with LOs and metadata. The second module generates images for text in LOs. Produced images are sent to the CNN-LSTM module to generate text for images. Generated text and corresponding LO text are exposed to similarity metrics, and images that achieve the best values are selected as images for text-based LO. Segmentation results and similarity results were tested for different LOs and topics. The test results showed that the proposed system can be used to visualize text in LOs for educational purposes as well as for other digital visualization systems. In the future, using these segmented images and the proposed GA-based image generation system, long text can be converted to dynamic animations. In addition, using our fitness evaluation function, automatic

Table 7 Comparison of algorithms in terms of computation time (CT) and best cost (BC)

		GA		ABC				DE				MPSO			
Image No	Topic	Mean CT	Mean BC	Mean CT	CT, ABC vs GA (p−value)	Mean BC	BC, ABC vs GA (p−value)	Mean CT	CT, DE vs. GA (p−value)	Mean BC	BC, DE vs GA (p−value)	Mean CT	CT, MPSO vs GA (p−value)	Mean BC	BC, MPSO vs GA (p−value)
1	FLA	686.074	0.389	704.695	+ (5.979e−08)	0.446	+ (2.524e−34)	643.500	−(6.195e−24)	0.429	+ (1.093e−33)	664.218	−(9.33e−11)	0.408	+ (6.467e−17)
2	FLA	681.433	0.389	699.075	+ (1.085e−07)	0.448	+ (2.524e−34)	643.559	−(4.452e−23)	0.428	+ (1.757e−33)	661.222	−(2.895e−09)	0.407	+ (2.213e−15)
3	FLA	680.568	0.390	705.573	+ (1.118e−12)	0.446	+ (2.524e−34)	643.500	−(1.313e−21)	0.429	+ (4.738e−34)	666.260	−(1.044e−05)	0.407	+ (1.525e−19)
4	FLA	686.136	0.390	706.024	+ (1.399e−08)	0.446	+ (2.524e−34)	645.179	−(8.558e−24)	0.429	+ (1.126e−33)	663.216	−(5.76e−12)	0.409	+ (5.898e−19)
5	FLA	682.441	0.389	708.272	+ (3.619e−13)	0.444	+ (2.524e−34)	643.952	−(2.729e−23)	0.431	+ (7.417e−34)	665.644	−(4.407e−07)	0.408	+ (1.215e−18)
6	FLA	685.112	0.388	703.649	+ (1.707e−08)	0.446	+ (2.524e−34)	644.772	−(1.017e−25)	0.429	+ (2.228e−33)	666.716	−(9.358e−09)	0.409	+ (2.278e−19)
7	FLA	681.744	0.389	701.471	+ (4.719e−09)	0.443	+ (2.524e−34)	648.643	−(5.28e−18)	0.428	+ (1.656e−33)	664.649	−(9.753e−07)	0.411	+ (7.989e−22)
8	FLA	681.744	0.390	704.469	+ (1.114e−10)	0.448	+ (2.524e−34)	646.470	−(1.63e−19)	0.427	+ (1.656e−33)	665.857	−(4.729e−06)	0.410	+ (3.944e−20)
9	FLA	685.113	0.389	703.979	+ (8.746e−08)	0.444	+ (2.524e−34)	648.159	−(1.042e−19)	0.427	+ (2.583e−33)	663.029	−(1.562e−10)	0.407	+ (2.519e−14)
10	FLA	685.440	0.391	701.678	+ (4.256e−06)	0.446	+ (2.524e−34)	645.144	−(3.01e−23)	0.428	+ (1.345e−33)	662.184	−(1.326e−11)	0.408	+ (3.69e−17)
11	HIS	687.066	0.420	703.315	+ (2.135e−06)	0.475	+ (2.524e−34)	646.668	−(1.588e−23)	0.455	+ (1.672e−31)	665.070	−(3.924e−11)	0.436	+ (9.361e−13)
12	HIS	680.210	0.417	700.553	+ (3.618e−09)	0.475	+ (2.524e−34)	646.218	−(6.3e−19)	0.456	+ (4.802e−33)	665.578	−(6.468e−06)	0.438	+ (1.918e−18)
13	HIS	685.897	0.418	700.991	+ (7.317e−07)	0.475	+ (2.524e−34)	647.592	−(1.685e−24)	0.455	+ (5.75e−32)	666.034	−(7.173e−10)	0.436	+ (1.451e−14)
14	HIS	684.482	0.419	700.592	+ (4.729e−06)	0.477	+ (2.524e−34)	647.335	−(2.254e−21)	0.459	+ (9.721e−33)	662.807	−(1.587e−10)	0.439	+ (8.625e−18)
15	HIS	680.361	0.418	702.912	+ (3.924e−11)	0.478	+ (2.524e−34)	646.492	−(7.927e−21)	0.453	+ (1.448e−31)	664.952	−(8.719e−07)	0.436	+ (8.052e−16)
16	HIS	684.111	0.418	702.433	+ (1.207e−07)	0.475	+ (2.524e−34)	645.826	−(1.121e−22)	0.456	+ (3.218e−32)	667.543	−(7.791e−07)	0.437	+ (1.966e−15)
17	HIS	686.857	0.418	702.351	+ (3.967e−06)	0.476	+ (2.524e−34)	643.000	−(5.005e−27)	0.457	+ (1.85e−32)	670.223	−(1.308e−07)	0.438	+ (8.052e−16)
18	HIS	685.513	0.419	703.453	+ (6.144e−08)	0.474	+ (2.524e−34)	646.990	−(6.417e−23)	0.455	+ (2.103e−31)	666.260	−(1.065e−08)	0.437	+ (3.988e−16)
19	HIS	682.945	0.419	703.589	+ (1.145e−08)	0.474	+ (2.524e−34)	648.034	−(2.74e−20)	0.456	+ (1.721e−31)	664.710	−(5.898e−08)	0.438	+ (3.533e−16)
20	HIS	683.277	0.419	701.174	+ (1.257e−07)	0.478	+ (2.524e−34)	644.705	−(3.486e−23)	0.457	+ (1.055e−31)	668.829	−(3.562e−05)	0.437	+ (5.65e−15)
21	BIO	683.771	0.379	699.891	+ (2.497e−06)	0.438	+ (2.524e−34)	647.064	−(4.203e−22)	0.420	+ (9.699e−34)	665.548	−(6.852e−08)	0.399	+ (2.328e−17)
22	BIO	683.628	0.382	700.383	+ (3.034e−07)	0.439	+ (2.524e−34)	645.718	−(3.516e−21)	0.419	+ (1.979e−33)	664.603	−(1.659e−08)	0.401	+ (1.186e−17)
23	BIO	680.929	0.382	703.520	+ (1.719e−10)	0.436	+ (2.524e−34)	644.208	−(1.548e−21)	0.418	+ (7.641e−34)	664.569	−(1.577e−06)	0.399	+ (6.347e−15)
24	BIO	683.590	0.379	700.923	+ (2.806e−07)	0.437	+ (2.524e−34)	647.212	−(2.231e−20)	0.418	+ (6.388e−34)	665.867	−(4.735e−08)	0.401	+ (2.052e−21)
25	BIO	681.981	0.378	699.622	+ (3.642e−08)	0.438	+ (2.524e−34)	646.411	−(4.037e−23)	0.419	+ (4.079e−34)	669.167	−(0.0001023)	0.399	+ (1.022e−20)
26	BIO	684.619	0.380	704.723	+ (3.259e−08)	0.434	+ (2.524e−34)	647.097	−(9.759e−21)	0.421	+ (4.203e−34)	668.886	−(1.915e−06)	0.400	+ (9.768e−19)
27	BIO	684.676	0.379	703.118	+ (1.596e−07)	0.437	+ (2.524e−34)	644.826	−(8.783e−22)	0.418	+ (6.582e−34)	665.475	−(4.582e−09)	0.399	+ (3.222e−18)
28	BIO	681.952	0.381	702.756	+ (1.006e−09)	0.437	+ (2.524e−34)	650.809	−(2.138e−18)	0.420	+ (4.203e−34)	664.359	−(5.433e−08)	0.398	+ (4.181e−17)
29	BIO	683.375	0.379	699.667	+ (7.989e−07)	0.437	+ (2.524e−34)	642.656	−(4.054e−24)	0.417	+ (1.608e−33)	666.859	−(4.819e−07)	0.398	+ (1.089e−17)
30	BIO	683.794	0.379	704.895	+ (1.006e−08)	0.438	+ (2.524e−34)	646.296	−(3.73e−22)	0.419	+ (1.06e−33)	661.517	−(1.467e−11)	0.400	+ (5.541e−20)
31	MAT	685.528	0.399	703.885	+ (1.613e−08)	0.456	+ (2.524e−34)	646.067	−(3.762e−24)	0.437	+ (1.268e−33)	666.055	−(7.867e−09)	0.422	+ (2.032e−23)
32	MAT	682.600	0.399	703.263	+ (8.114e−10)	0.457	+ (2.524e−34)	644.565	−(1.061e−21)	0.435	+ (1.646e−32)	666.014	−(6.785e−07)	0.415	+ (2.93e−14)
33	MAT	679.983	0.399	704.086	+ (9.614e−12)	0.455	+ (2.524e−34)	647.632	−(4.171e−18)	0.437	+ (6.258e−33)	661.426	−(1.885e−08)	0.419	+ (8.443e−18)
34	MAT	683.626	0.398	703.807	+ (3.119e−09)	0.454	+ (2.524e−34)	646.698	−(6.607e−22)	0.437	+ (8.902e−33)	665.889	−(1.683e−07)	0.419	+ (1.09e−18)

Table 7 (continued)

Image No	Topic	GA		ABC				DE				MPSO			
		Mean CT	Mean BC	Mean CT	CT, ABC vs GA (p−value)	Mean BC	BC, ABC vs GA (p−value)	Mean CT	CT, DE vs. GA (p−value)	Mean BC	BC, DE vs GA (p−value)	Mean CT	CT, MPSO vs GA (p−value)	Mean BC	BC, MPSO vs GA (p−value)
35	MAT	685.898	0.397	702.399	+ (1.345e−06)	0.457	+ (2.524e−34)	643.520	−(1.128e−25)	0.437	+ (3.682e−33)	666.141	−(3.512e−09)	0.417	+ (3.684e−20)
36	MAT	683.960	0.399	704.435	+ (9.092e−09)	0.456	+ (2.524e−34)	644.422	−(2.15e−22)	0.438	+ (6.445e−33)	664.510	−(2.769e−09)	0.418	+ (6.333e−16)
37	MAT	685.023	0.398	700.976	+ (9.866e−06)	0.458	+ (2.524e−34)	647.040	−(1.347e−20)	0.439	+ (3.575e−33)	661.091	−(1.419e−11)	0.419	+ (1.022e−20)
38	MAT	683.251	0.400	701.489	+ (3.125e−08)	0.456	+ (2.524e−34)	646.905	−(2.005e−21)	0.436	+ (6.092e−32)	664.445	−(7.532e−09)	0.419	+ (1.152e−16)
39	MAT	681.551	0.400	704.478	+ (9.033e−11)	0.456	+ (2.524e−34)	642.252	−(3.472e−22)	0.438	+ (5.402e−33)	667.264	−(3.96e−05)	0.419	+ (2.047e−18)
40	MAT	681.034	0.399	703.697	+ (2.856e−10)	0.457	+ (2.524e−34)	646.510	−(9.737e−20)	0.439	+ (1.029e−33)	665.756	−(6.93e−06)	0.417	+ (5.845e−16)
41	PHY	685.284	0.409	705.144	+ (1.399e−08)	0.467	+ (2.524e−34)	646.712	−(1.773e−22)	0.448	+ (1.029e−33)	663.793	−(1.045e−10)	0.425	+ (3.988e−16)
42	PHY	681.623	0.409	703.599	+ (8.604e−11)	0.467	+ (2.524e−34)	647.851	−(1.317e−20)	0.446	+ (2.269e−32)	667.578	−(3.377e−05)	0.426	+ (7.071e−14)
43	PHY	684.904	0.409	702.120	+ (1.946e−07)	0.463	+ (2.524e−34)	650.318	−(1.783e−21)	0.445	+ (3.036e−32)	667.440	−(6.061e−08)	0.427	+ (1.028e−14)
44	PHY	681.207	0.410	701.324	+ (2.939e−09)	0.464	+ (2.524e−34)	645.245	−(3.202e−21)	0.448	+ (4.832e−32)	658.578	−(7.574e−12)	0.427	+ (3.367e−13)
45	PHY	683.088	0.408	703.278	+ (6.608e−09)	0.464	+ (2.524e−34)	645.559	−(1.572e−22)	0.447	+ (8.645e−33)	664.899	−(6.143e−09)	0.430	+ (3.288e−20)
46	PHY	680.253	0.407	702.731	+ (7.434e−11)	0.467	+ (2.524e−34)	646.706	−(2.131e−20)	0.446	+ (3.471e−33)	666.581	−(1.782e−05)	0.428	+ (5.93e−20)
47	PHY	683.139	0.409	703.032	+ (3.954e−09)	0.467	+ (2.524e−34)	644.808	−(2.052e−21)	0.445	+ (1.422e−32)	664.268	−(3.796e−08)	0.425	+ (2.291e−14)
48	PHY	685.585	0.407	700.179	+ (6.209e−05)	0.466	+ (2.524e−34)	644.689	−(2.774e−25)	0.446	+ (2.079e−32)	665.828	−(3.396e−10)	0.426	+ (6.106e−15)
49	PHY	685.678	0.408	699.419	+ (6.339e−05)	0.467	+ (2.524e−34)	648.648	−(1.376e−21)	0.446	+ (3.084e−33)	666.424	−(4.194e−09)	0.427	+ (2.328e−17)
50	PHY	683.486	0.409	701.972	+ (3.397e−08)	0.467	+ (2.524e−34)	650.309	−(1.046e−20)	0.447	+ (3.721e−32)	664.201	−(6.608e−09)	0.427	+ (3.754e−16)
51	GEO	687.473	0.428	702.772	+ (7.01e−06)	0.487	+ (2.524e−34)	646.323	−(9.463e−23)	0.466	+ (5.918e−32)	663.746	−(1.8e−12)	0.448	+ (4.389e−15)
52	GEO	685.262	0.430	702.776	+ (2.88e−07)	0.485	+ (2.524e−34)	642.928	−(4.625e−25)	0.465	+ (2.293e−30)	665.416	−(7.423e−09)	0.446	+ (5.446e−14)
53	GEO	680.008	0.427	704.974	+ (1.288e−12)	0.485	+ (2.524e−34)	646.567	−(1.333e−19)	0.463	+ (3.624e−31)	666.612	−(4.22e−05)	0.447	+ (6.881e−17)
54	GEO	684.702	0.430	701.981	+ (6.212e−07)	0.486	+ (2.524e−34)	649.047	−(1.695e−20)	0.465	+ (6.786e−31)	665.315	−(4.241e−08)	0.450	+ (1.491e−15)
55	GEO	682.994	0.426	706.373	+ (7.705e−12)	0.487	+ (2.524e−34)	647.893	−(3.202e−21)	0.464	+ (2.782e−32)	665.382	−(2.052e−08)	0.447	+ (2.228e−19)
56	GEO	683.214	0.428	701.944	+ (1.455e−07)	0.484	+ (2.524e−34)	646.176	−(5.281e−23)	0.464	+ (4.694e−32)	665.490	−(2.718e−08)	0.447	+ (3.393e−16)
57	GEO	685.309	0.429	701.021	+ (4.841e−06)	0.484	+ (2.524e−34)	649.150	−(4.543e−21)	0.465	+ (1.025e−31)	664.905	−(4.32e−09)	0.446	+ (2.164e−14)
58	GEO	687.199	0.428	704.053	+ (5.831e−07)	0.484	+ (2.524e−34)	647.099	−(3.319e−24)	0.466	+ (1.367e−31)	664.856	−(2.681e−11)	0.446	+ (1.11e−14)
59	GEO	681.816	0.431	703.053	+ (7.08e−11)	0.483	+ (2.524e−34)	644.165	−(1.863e−24)	0.466	+ (3.729e−31)	666.110	−(1.132e−06)	0.446	+ (1.288e−10)
60	GEO	684.228	0.430	704.413	+ (1.659e−08)	0.485	+ (2.524e−34)	647.113	−(5.334e−22)	0.465	+ (1.266e−29)	663.103	−(2.217e−10)	0.446	+ (2.06e−13)

text can be generated for paraphrasing tools (the system can generate text similar to a reference text).

Declarations

Conflict of interest The author declares that there is no conflict of interest.

References

1. Tam M (2000) Constructivism, instructional design, and technology: implications for transforming distance learning. Educ Tech Soc 3(2):50–60
2. Bozkurt A, Sharma RC (2020) Emergency remote teaching in a time of global crisis due to CoronaVirus pandemic. Asian J Distance Educ 5(1):1–6
3. Sun L, Tang Y, Zuo W (2020) Coronavirus pushes education online. Nat Mater 19:1–1
4. Simonson M, Zvacek SM, Smaldino S (2019) Teaching and learning at a distance. IAP, Charlotte NC
5. Driscoll M (2002) Blended learning: let's get beyond the hype. E Learn 1(4):1–4
6. Ferreira-Mello R, André M, Pinheiro A, Costa E, Romero C (2019) Text mining in education. Wiley Interdiscip Rev Data Min Knowl Discov 9(6):1–49
7. Romero C, Ventura S (2017) Educational data science in massive open online courses. Wiley Interdiscip Rev Data Min Knowl Discov 7(1):1–12
8. Hamdi M, Hamtini T (2016) Designing an effective e-content development framework for the enhancement of learning programming. Int J Emerg Technol Learn 11(4):131–141
9. Ince M, Yigit T, Isik AH (2017) AHP-TOPSIS method for learning object metadata evaluation. Int J Inf Educ Technol 12:884–887
10. McGreal R, Roberts T (2001) A primer on metadata for learning objects: fostering an interoperable environment. E Learn 2(10):26–29
11. Sinclair J, Joy M, Yau JYK, Hagan S (2013) A practice-oriented review of learning objects. IEEE Trans Learn Technol 6(2):177–192
12. McClelland M (2003) Metadata standards for educational resources. Comput 36(11):107–109
13. Brooks C, McCalla G (2006) Towards flexible learning object metadata. Int J Contin Eng Educ Life Long Learn 16(1–2):50–63
14. Yigit T, Isik AH, Ince M (2014) Multi criteria decision making system for learning object repository. Proced Soc Behav Sci 141:813–816
15. Wiley DA (2000) Connecting learning objects to instructional design theory: a definition, a metaphor, and a taxonomy. The instructional use of learning objects. http://reusability.org/read/chapters/wiley.doc. Accessed 10 August 2020
16. Scheiter K, Gerjets P, Catrambone R (2006) Making the abstract concrete: Visualizing mathematical solution procedures. Comput Hum Behav 22(1):9–25
17. Twyman T, Tindal G (2006) Using a computer-adapted, conceptually based history text to increase comprehension and problem-solving skills of students with disabilities. J Spec Educ Technol 21(2):5–16
18. Bernardi R, Cakici R, Elliott D, Erdem A, Erdem E, Ikizler-Cinbis N, Plank B (2016) Automatic description generation from images: a survey of models, datasets, and evaluation measures. J Artif Intell Res 55:409–442
19. Williamson B (2016) Digital education governance: data visualization, predictive analytics, and 'real-time'policy instruments. J Educ Pol 31(2):123–141
20. Yang Y, Yao Q, Qu H (2017) VISTopic: a visual analytics system for making sense of large document collections using hierarchical topic modeling. Vis Inf 1(1):40–47
21. Buckley J, Seery N, Canty D, Gumaelius L (2018) Visualization, inductive reasoning, and memory span as components of fluid intelligence: implications for technology education. Int J Educ Res 90:64–77
22. Schmidgall SP, Eitel A, Scheiter K (2019) Why do learners who draw perform well? Investigating the role of visualization, generation and externalization in learner-generated drawing. Learn Instr 60:138–153
23. Schnotz W, Bannert M (2003) Construction and interference in learning from multiple representation. Learn Instr 13(2):141–156
24. Braun M, Broy N, Pfleging B, Alt F (2019) Visualizing natural language interaction for conversational in-vehicle information systems to minimize driver distraction. J Multimodal User Interfaces 13(2):71–88
25. Gaona-García PA, Martin-Moncunill D, Montenegro-Marin CE (2017) Trends and challenges of visual search interfaces in digital libraries and repositories. Electron Libr 35(1):69–98
26. Gershon N, Page W (2001) What storytelling can do for information visualization. Commun ACM 44(8):31–37
27. Zheng JG (2017) Data visualization for business intelligence. In: Munoz M (ed) Global business intelligence, 1st edn. Taylor and Francis, New York, pp 67–82
28. Liu S, Wang X, Collins C, Dou W, Ouyang F, El-Assady M, Keim DA (2018) Bridging text visualization and mining: a task-driven survey. IEEE Trans Vis Comput Gr 25(7):2482–2504
29. Van Wierst P, Hofstede S, Oortwijn Y, Castermans T, Koopman R, Wang S, Betti A (2018) BolVis: visualization for text-based research in philosophy. In: 3rd workshop on visualization for the digital humanities, pp 1–6
30. Siirtola H, Isokoski P, Säily T, Nevalainen T (2016) Interactive text visualization with text variation explorer. In: IEEE 20th international conference information visualization, pp 330–335
31. Singh J, Zerr S, Siersdorfer S (2017) Structure-aware visualization of text corpora. In: Proceedings of the 2017 conference on human information interaction and retrieval, pp 107–116
32. Sui Z (2019) Social media text data visualization modeling: a timely topic score technique. Am J Manag Sci Eng 4(3):49–55
33. Sultanum N, Brudno M, Wigdor D, Chevalier F (2018) More text please! understanding and supporting the use of visualization for clinical text overview. In: Proceedings of the 2018 CHI conference on human factors in computing systems, pp 1–13
34. Yamada A, Yamamoto T, Ikeda H, Nishida T, Doshita S (1992, August) Reconstructing spatial image from natural language texts. In: Proceedings of the 14th conference on computational linguistics, pp 1279–1283
35. Joshi D, Wang JZ, Li J (2006) The story picturing engine: a system for automatic text illustration. ACM Trans Multimed Comput Commun Appl 2(1):68–89
36. Mihalcea R, Leong CW (2008) Toward communicating simple sentences using pictorial representations. Mach Transl 22(3):153–173
37. Ustalov D (2012) A text-to-picture system for Russian language. In: Proceedings 6th Russian young scientist conference for information retrieval, pp 35–44
38. Bui D, Nakamura C, Bray BE, Zeng-Treitler Q (2012) Automated illustration of patients instructions. In: AMIA annual symposium proceedings, pp 1158–1167

39. Ruan W, Appasani N, Kim K, Vincelli J, Kim H, Lee WS (2018) Pictorial visualization of EMR summary interface and medical information extraction of clinical notes. In: IEEE international conference on computational intelligence and virtual environments for measurement systems and applications, pp 1–6
40. Jiang Y, Liu J, Lu H (2016) Chat with illustration. Multimed Syst 22(1):5–16
41. Jain P, Darbari H, Bhavsar VC (2014) Vishit: A visualizer for hindi text. In: IEEE fourth international conference on communication systems and network technologies, pp 886–890
42. Ramisa A, Yan F, Moreno-Noguer F, Mikolajczyk K (2017) Breakingnews: article annotation by image and text processing. IEEE Trans Pattern Anal Mach Intell 40(5):1072–1085
43. Hassani K, Lee WS (2016) Visualizing natural language descriptions: a survey. ACM Comput Surv 49(1):1–34
44. Adorni G, Di Manzo M, Giunchiglia F (1984) Natural language driven image generation. In: 10th international conference on computational linguistics and 22nd annual meeting of the association for computational linguistics, pp 495–500
45. Coyne B, Sproat R (2001) WordsEye: an automatic text-to-scene conversion system. In: Proceedings of the 28th annual conference on computer graphics and interactive techniques, pp 487–496
46. Huang CJ, Li CT, Shan MK (2013) VizStory: visualization of digital narrative for fairy tales. In: IEEE conference on technologies and applications of artificial intelligence, pp 67–72
47. Karkar AG, Alja'am JM, Mahmood A, (2017) Illustrate it! An Arabic multimedia text-to-picture m-learning system. IEEE Access 5:12777–12787
48. Zhang S, Shen W, Ghenniwa H (2004) A review of Internet-based product information sharing and visualization. Comput Ind 54(1):1–15
49. Afzal S, Maciejewski R, Jang Y, Elmqvist N, Ebert DS (2012) Spatial text visualization using automatic typographic maps. IEEE Trans Vis Comput Gr 18(12):2556–2564
50. Fatemah A, Rasool S, Habib U (2020) Interactive 3D Visualization of chemical structure diagrams embedded in text to aid spatial learning process of students. J Chem Educ 97(4):992–1000
51. Gunarathne WKTM, Chootong C, Sommool W, Ochirbat A, Chen YC, Reisman S, Shih TK (2018) Web-based learning object search engine solution together with data visualization: the case of MERLOT II. In: IEEE 42nd annual computer software and applications conference, pp 1026–1031
52. Willrich R, Mittmann A, Fileto R, Dos Santos AL (2019) Capture and visualisation of text understanding through semantic annotations and semantic networks for teaching and learning. J Inf Sci 46(4):528–543
53. Kirillov A, He K, Girshick R, Rother C, Dollár P (2019) Panoptic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 9404–9413
54. Kirillov A, Girshick R, He K, Dollár P (2019) Panoptic feature pyramid networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 6399–6408
55. Cai W, Xiong Z, Sun X, Rosin PL, Jin L, Peng X (2020) Panoptic segmentation-based attention for image captioning. Appl Sci 10(1):391
56. Li Q, Arnab A, Torr PH (2018) Weakly-and semi-supervised panoptic segmentation. In: Proceedings of the european conference on computer vision, pp 102–118
57. Li Y, Chen X, Zhu Z, Xie L, Huang G, Du D, Wang X (2019) Attention-guided unified network for panoptic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 7026–7035
58. De Geus D, Meletis P, Dubbelman G (2018) Panoptic segmentation with a joint semantic and instance segmentation network. arxiv. https://arxiv.org/pdf/1809.02110.pdf Accessed 10 August 2020
59. Liu H, Peng C, Yu C, Wang J, Liu X, Yu G, Jiang W (2019) An end-to-end network for panoptic segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 6172–6181
60. De Geus D, Meletis P, Dubbelman G (2020) Fast panoptic segmentation network. IEEE Robot Autom Lett 5(2):1742–1749
61. Lazarow J, Lee K, Tu Z (2019) Learning Instance Occlusion for Panoptic Segmentation. Arxiv. https://arxiv.org/pdf/1906.05896.pdf Accessed 10 August 2020
62. Mohan R, Valada A (2020) Efficientps: Efficient panoptic segmentation. Arxiv. https://arxiv.org/pdf/2004.02307.pdf Accessed 10 August 2020
63. Nabiyev VV (2012) Yapay Zeka [Artificial Intelligence]. Seçkin Yayıncılık [Publishing], Ankara
64. Stefanini MH, Demazeau Y (1995, October) TALISMAN: a multi-agent system for natural language processing. In: Brazilian symposium on artificial intelligence, pp 312–322
65. Strzalkowski T, Lin F, Wang J, Perez-Carballo J (1999) Evaluating natural language processing techniques in information retrieval. In: Strzalkowski T (ed) Natural language information retrieval. Springer, Dordrecht, pp 113–145
66. Cushing J, Hastings R (2009) Introducing computational linguistics with NLTK (Natural Language Toolkit). J Comput Sci Coll 25(1):167–169
67. İnce EY (2017) Spell checking and error correcting application for Turkish. Int J Inf Electron Eng 7(2):68–71
68. Volna E, Kotyrba M (2014) A Comparative study to evolutionary algorithms. Eur Con Model Simul 340–345
69. Fogel DB (2006) Evolutionary computation: toward a new philosophy of machine intelligence. Wiley
70. Goldberg DE, Holland JH (1988) Genetic algorithms and machine learning. Mach Learn 3:95–99
71. LeCun Y, Bengio Y, Hinton G (2015) Deep learning. Nature 521(7553):436–444
72. Vinyals O, Toshev A, Bengio S, Erhan D (2016) Show and tell: lessons learned from the 2015 Mscoco image captioning challenge. IEEE Trans Pattern Anal Mach Intell 39(4):652–663
73. Toderici G, Vincent D, Johnston N, Jin Hwang S, Minnen D, Shor J, Covell M (2017) Full Resolution Image Compression with Recurrent Neural Networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 5306–5314
74. Bengio Y, Simard P, Frasconi P (1994) Learning long-term dependencies with gradient descent is difficult. IEEE Trans Neural Netw 5(2):157–166
75. Cocodataset (2020) https://cocodataset.org/#panoptic-2020. Accessed 30 October 2020
76. Yao L, Chyau A (2019) A unified neural network for panoptic segmentation. Comput Gr Forum 38(7):461–468
77. Papineni K, Roukos S, Ward T, Zhu WJ (2002) BLEU: a method for automatic evaluation of machine translation. In: Proceedings of the 40th annual meeting of the Association for Computational Linguistics, pp 311–318
78. Banerjee S, Lavie A (2005) METEOR: an automatic metric for MT evaluation with improved correlation with human judgments. In: Proceedings of the ACL workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization, pp 65–72
79. Vedantam R, Lawrence Zitnick C, Parikh D (2015) Cider: consensus-based image description evaluation. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 4566–4575

80. Wang C, Yang H, Bartz C, Meinel C (2016) Image captioning with deep bidirectional LSTMs. In: Proceedings of the 24th ACM international conference on multimedia, pp 988–997
81. Lin TY, Dollar P (2016) Ms coco api. https://github.com/cocodataset/cocoapi Accessed 15 April 2020
82. Chakraborty S, Seal A, Roy M (2015) An elitist model for obtaining alignment of multiple sequences using genetic algorithm. In: Proceedings 2nd national conference NCETAS, pp 61–67
83. Wang C, Gao Y (2013) Determination of power distribution network configuration using non-revisiting genetic algorithm. IEEE Trans Power Syst 28(4):3638–3648
84. Srivastava S (2019) Image-captioning. https://github.com/siddsrivastava/Image-captioning Accessed 15 April 2020
85. Chang Z, Zhang Y, Chen W (2019) Electricity price prediction based on hybrid model of adam optimized LSTM neural network and wavelet transform. Energy 187:115804
86. İnce M, Yiğit T, Işık AH (2019) A hybrid AHP-GA method for metadata-based learning object evaluation. Neural Comput Appl 31(1):671–681
87. Akay B, Karaboga D (2012) Artificial bee colony algorithm for large-scale problems and engineering design optimization. J Intell Manuf 23(4):1001–1014
88. Karaboğa D, Ökdem S (2004) A simple and global optimization algorithm for engineering problems: differential evolution algorithm. Turkish J Electr Eng Comput Sci 12(1):53–60
89. Andrews PS (2006) An investigation into mutation operators for particle swarm optimization. In: IEEE International Conference on Evolutionary Computation, pp 1044–1051
90. De Barros RSM, Hidalgo JIG, De Lima Cabral DR (2018) Wilcoxon rank sum test drift detector. Neurocomputing 275:1954–1963