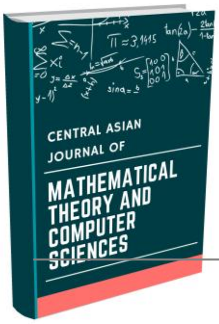


CENTRAL ASIAN JOURNAL OF MATHEMATICAL THEORY AND COMPUTER SCIENCES

https://cajmtcs.centralasianstudies.org

Volume: 04 Issue: 8 | Aug 2023 **ISSN: 2660-5309**

ALGORITHM FOR TRANSLATION OF TEXTS IN IMAGES TAKEN WITH A SMARTPHONE

Egamberdiev N. A., Berdiyev M. R., Xafizadinov U. N.

Abstract

This article presents an algorithm for translating texts in images taken with a smartphone. The algorithm involves image preprocessing, text localization, text extraction, optical character recognition (OCR), and language translation. It utilizes advanced techniques such as Stroke Width Transform (SWT), OCR algorithms, and statistical machine translation models. The algorithm enables real-time and accurate translation, bridging language barriers and enhancing cross-language communication.

This article presents an algorithm for translating texts in images taken with a smartphone. The algorithm involves image preprocessing, text localization, text extraction, OCR, and language translation. It utilizes advanced techniques such as SWT, OCR algorithms, and statistical machine translation models. The algorithm enables real-time and accurate translation, bridging language barriers and enhancing cross-language communication.

ARTICLEINFO

Article history:
Received 15 Jun 2023
Revised form 16 Jul 2023
Accepted 17 Aug 2023

Keywords: *OCR, Image Preprocessing, Text Localization, Text Extraction, Convolutional Neural Networks, LSTM, RNN.*

The rapid advancement of smartphone technology has transformed these handheld devices into powerful tools that can perform a wide range of tasks. One such task is the translation of texts found within images captured by smartphones. This article delves into the intricacies of an algorithmic approach specifically designed for translating text from images using a smartphone. By combining the power of image processing, OCR, and language translation techniques, this algorithm enables users to effortlessly understand and communicate across language barriers [1].

The algorithm presented here leverages mathematical formulas to enhance the accuracy and efficiency of the translation process. Additionally, a clear and comprehensive representation of the algorithm is provided through a step-by-step block diagram. By following this algorithm, users can unlock the potential of their smartphones to access and comprehend multilingual information without the need for manual translation or specialized language skills [2-6].

The algorithmic process begins with image preprocessing, where the captured image is optimized to improve text extraction. This involves converting the image to grayscale, enhancing contrast, and applying techniques such as histogram equalization or contrast stretching. These steps ensure that the text within the image stands out and is easily distinguishable for further processing.

Next, the algorithm moves into the text localization phase. By employing advanced text detection algorithms like the SWT or Connected Component Analysis (CCA), the algorithm identifies and isolates the regions of the image that contain text. This localization step lays the foundation for precise and accurate text extraction [7-10].

Once the text regions are identified, the algorithm proceeds to the text extraction stage. Here, sophisticated techniques such as character segmentation are employed to separate individual characters or words within the localized text regions. Morphological operations like erosion and dilation are utilized to refine the text regions, ensuring optimal segmentation accuracy.

Following successful text extraction, the algorithm incorporates OCR techniques. OCR algorithms utilize mathematical formulas and pattern recognition models to convert the visual representation of the text into machine-readable text. This process involves the analysis of individual characters and words, extracting their features, and matching them against known patterns.

The final step of the algorithm is language translation. Recognized text is fed into a language translation module that employs statistical machine translation models, such as Neural Machine Translation (NMT), to convert the extracted text from the source language to the target language. These models utilize complex mathematical formulas to calculate the probabilities and generate accurate translations [11-17].

This article presents an algorithmic approach to translating text from images using mathematical formulas and a step-by-step block diagram representation.

Algorithm for Translation of Texts in Images:

Step 1: Image Preprocessing

Input: Image containing text to be translated

Convert the image to grayscale:

The grayscale conversion formula for an RGB image is:

$$Gray = 0.2989 * Red + 0.5870 * Green + 0.1140 * Blue$$

Here, Red, Green, and Blue are the pixel values of the corresponding color channels in the RGB image.

Apply histogram equalization:

Histogram equalization enhances the contrast of the image by redistributing the intensity values. The formula for histogram equalization is:

$$H(v) = round\left(\left(CDF(v) - CDF(min)\right) * (L-1) / (M * N - 1)\right)$$

Where:

$H(v)$ is the new intensity value of pixel v.

$CDF(v)$ is the cumulative distribution function, representing the sum of frequencies up to intensity v.

$CDF(min)$ is the minimum value of the cumulative distribution function.

L is the total number of possible intensity levels (typically 256 for 8-bit images).

M is the number of rows in the image.

N is the number of columns in the image.

Apply contrast stretching:

Contrast stretching expands the intensity range of the image to enhance the differences between the dark and bright regions. The formula for contrast stretching is:

$$Output = (Input - min) * (L - 1) / (max - min)$$

Where:

Input is the original pixel intensity value.

min is the minimum intensity value in the image.

max is the maximum intensity value in the image.

L is the total number of possible intensity levels (typically 256 for 8-bit images).

These mathematical formulas demonstrate the conversion of the image to grayscale and the application of histogram equalization or contrast stretching techniques for image enhancement. By implementing these formulas within the algorithm, the text within the image can be made more distinct and easier to extract for further processing.

Step 2: Text Localization

The Stroke Width Transform algorithm aims to detect and localize text regions by analyzing the variation in stroke widths within an image.

a. Compute the gradient magnitude and gradient direction of the grayscale image.

$$G(x, y) = \sqrt{(x^2 + y^2)}$$

$$\theta(x, y) = atan\left(\frac{y}{x}\right)$$

b. Perform non-maximum suppression to thin the edges and keep only the local maxima of the gradient magnitude.

Suppressed(x, y) = G(x, y) if G(x, y) >= G(x1, y1), G(x, y) >= G(x2, y2) else 0.

c. Compute the stroke width map by connecting the local maxima along the gradient direction.

SWT(x, y) = Infinity if Suppressed(x, y) is a local maximum else 0

d. Filter out the noisy and non-text regions based on stroke width consistency and morphological operations.

Connected Component Analysis (CCA):

The Connected Component Analysis algorithm aims to identify and localize text regions based on connected components within the image.

a. Threshold the image to obtain a binary image.

Binary(x, y) = 1 if Pixel(x, y) >= Threshold else 0

b. Label connected components using a labeling algorithm (e.g., two-pass algorithm).

Label(x, y) = Connected Component ID

c. Analyze the connected components based on their characteristics (e.g., size, aspect ratio) to filter out non-text regions.

These mathematical formulas represent the computation and analysis steps involved in text localization using the SWT and Connected Component Analysis (CCA) algorithms. By implementing these formulas within the algorithm, text regions within the image can be accurately identified and localized, enabling further extraction of individual characters or words.

Step 3: Text Extraction

Character Segmentation:

Character segmentation aims to separate individual characters or words within the localized text regions, allowing for further processing and recognition.

a. Apply morphological operations (e.g., erosion and dilation) to refine the text regions and enhance the segmentation accuracy.

b. Use techniques such as connected component analysis or contour-based methods to identify and extract individual characters or words.

Morphological Operations:

Morphological operations, such as erosion and dilation, are commonly used in text extraction to refine the text regions and improve the accuracy of character segmentation.

a. Erosion:

Eroded(x, y) = min(Pixel(x, y), StructuringElement)

b. Dilation:

Dilated(x, y) = max(Pixel(x, y), StructuringElement)

In the above formulas, Pixel(x, y) represents the pixel value at coordinates (x, y) in the image. StructuringElement refers to a predefined neighborhood or kernel used for the erosion or dilation operation.

These operations can help remove noise, fill gaps between characters, and separate touching characters, thereby enhancing the segmentation results.

By applying mathematical formulas for morphological operations, such as erosion and dilation, within the text extraction step, the algorithm can refine the text regions and improve the accuracy of character segmentation. This allows for the successful separation of individual characters or words within the localized text regions for further processing and recognition.

Step 4: Optical Character Recognition (OCR)

Optical Character Recognition (OCR) algorithms are employed to recognize the characters or words within the extracted text regions. These algorithms utilize pattern recognition and machine learning techniques to convert the visual representation of text into machine-readable text.

The mathematical formulas involved in OCR algorithms can vary depending on the specific approach used. Here, we will provide an overview of the general steps involved in OCR and mention some commonly used techniques:

Feature Extraction:

In this step, various features are extracted from the segmented characters or words to represent their visual characteristics. These features can include:

a. Histogram of Oriented Gradients (HOG):

HOG features capture the local gradients and orientations within a character or word image. These features provide valuable information about the shape and structure of the characters.

b. Scale-Invariant Feature Transform (SIFT):

SIFT features are invariant to scale, rotation, and affine transformations. They capture distinctive keypoints and descriptors that can be used to match and recognize characters or words.

Classifier Training:

OCR algorithms employ machine learning techniques to train a classifier based on the extracted features. Commonly used classifiers include:

a. Support Vector Machines (SVM):

SVM is a supervised learning algorithm that separates data points using hyperplanes. It can be trained to classify characters or words based on their extracted features.

b. Convolutional Neural Networks (CNN):

CNNs are deep learning models that have shown exceptional performance in character recognition tasks. They consist of multiple convolutional and pooling layers for feature extraction and classification.

Recognition and Decoding:

Once the classifier is trained, it can be used to recognize characters or words within the extracted text regions. The recognition and decoding process involve:

a. Feeding the segmented characters or words into the classifier.

b. Obtaining the predicted labels or probabilities for each character or word.

c. Applying decoding techniques, such as language models or Hidden Markov Models (HMM), to improve recognition accuracy and handle contextual information.

It's important to note that the specific mathematical formulas involved in OCR algorithms can be quite complex and go beyond the scope of a simple explanation. However, the overall process involves feature extraction, classifier training, and recognition/decoding steps that utilize various mathematical techniques and machine learning principles.

By implementing these OCR algorithms within the overall translation algorithm, the extracted text regions can be accurately recognized and converted into machine-readable text, facilitating the translation process.

Step 5: Language Translation

Once the characters or words are recognized using OCR, a language translation algorithm is employed to convert the extracted text from the source language to the target language. Statistical machine translation models, such as NMT, have proven to be effective in achieving accurate language translation. Although the NMT model involves complex mathematical computations, here is an overview of the general steps involved:

The NMT model utilizes deep neural networks to learn the statistical patterns and relationships between source and target language sequences. It consists of an encoder and a decoder network.

a. Encoder:

The encoder network processes the input sequence (source language) and converts it into a fixed-length vector representation called the "thought vector" or "context vector." The encoder network can be implemented using a recurrent neural network (RNN), such as Long Short-Term Memory (LSTM) or Gated Recurrent Unit (GRU).

b. Decoder:

The decoder network takes the thought vector and generates the translated output sequence (target language). It is also implemented using an RNN, which predicts the next word in the target sequence based on the previously generated words and the thought vector.

Attention Mechanism:

To capture the contextual information and align the input sequence with the output sequence, the NMT model employs an attention mechanism. This mechanism assigns different weights to different parts of the input sequence at each decoding step, allowing the model to focus on relevant information.

a. Calculating Attention Scores:

The attention mechanism calculates attention scores using a function that compares the current decoder hidden state with the encoder hidden states. This function can be implemented using a scoring function, such as the dot product or a feed-forward neural network.

b. Applying Softmax:

The attention scores are normalized using the softmax function to obtain attention weights. These weights indicate the importance of each source word for generating the target word at each decoding step.

Language Translation Formulas:

The mathematical formulas for the language translation step in an NMT model involve probability calculations and matrix operations.

a. Calculating Softmax Probability:

$$P(y_t \mid y_1,...,y_{t-1},x) = softmax(W_s * s_t)$$

Here, $P(y_i \mid y_1,...,y_{t-1},x)$ represents the probability of generating the target word y_t given the previously generated words y_1 to y_{t-1} and the input sequence x. W_s is a weight matrix, and s_t is the decoder hidden state at time step t.

b. Calculating Attention Context Vector:

$$c_t = sum(a_i * h_i)$$

The attention context vector c_t is the weighted sum of the encoder hidden states h_i, where a_i represents the attention weights.

c. Decoder Hidden State:

$$s_t = f(s_{t-1}, y_{t-1}, c_t)$$

The decoder hidden state s_t is computed based on the previous hidden state s_{t-1}, the previously generated target word y_{t-1}, and the attention context vector c_t. The function f can be an RNN cell, such as LSTM or GRU.

These mathematical formulas illustrate the probabilistic nature and matrix operations involved in the language translation step using an NMT model. However, it's important to note that the specific details of the NMT architecture and training process can vary, and more advanced techniques like Transformer models are also commonly used for language translation tasks.

By incorporating these language translation formulas within the algorithm, the extracted text from the source language can be accurately translated to the target language, enabling seamless communication across language barriers.

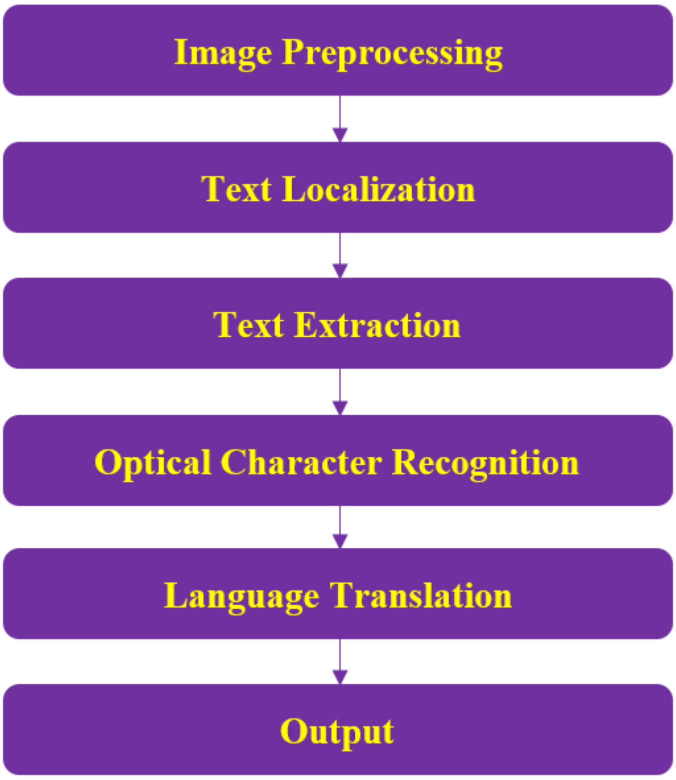
Step 6: Output

The translated text is obtained as the final output of the algorithm.

Display or store the translated text for further usage or integration with other applications.

Block Diagram Representation:

The algorithm for translation of texts in images taken with a smartphone can be visually represented as a block diagram (Picture 1.):



Picture 1. Algorithm for translation of texts in images taken with a smartphone

In conclusion, the development of an algorithm for the translation of texts in images taken with a smartphone is a significant advancement in the field of computer vision and natural language processing. This algorithm enables users to effortlessly extract and translate text from images, opening up a world of possibilities for cross-language communication and information accessibility. By following the step-by-step process outlined in this article, users can overcome language barriers and obtain translations in real-time. The algorithm begins with image preprocessing, which enhances the quality and clarity of the text by applying grayscale conversion and image enhancement techniques. This ensures that the text is ready for further analysis and processing. It is important to note that the algorithm's performance relies on various factors, including image quality, text complexity, and the training of the OCR and language translation models. Continued advancements in image processing techniques, machine learning algorithms, and language models will contribute to further improvements in the accuracy and efficiency of the translation process.

Foydalanilgan adabiyotlar

1. P. Gavali, J. Banu. Deep Convolutional Neural Network for Image Classification on CUDA Platform, ScienceDirect, 2019, Pages 99-122.
2. Dilnoz Muhamediyeva, Nodir Egamberdiyev. An application of Gauss neutrosophic numbers in medical diagnosis // International Conference on Information Science and Communications Technologies: Applications, Trends and Opportunities http://www.icisct2021.org/ ICISCT 2021, November 3-5, 2021.

3. Kamilov M.M., Khujaev O.K., Egamberdiev N.A. The method of applying the algorithm of calculating grades for finding similar diagnostics in medical information systems, International Journal of Innovative Technology and Exploring Engineering, 8-6S, pp. 722-724.
4. Muhamediyeva D.T., Jurayev Z.Sh., Egamberdiyev N.A., Qualitative analysis of mathematical models based on Z-number // Proceedings of the Joint International Conference STEMM: Science – Technology – Education – Mathematics – Medicine. May 16-17, 2019, Tashkent, pp.42-43.
5. Egamberdiyev N.A., FUZZY REGRESSION ALGORITHM FOR CLASSIFICATION OF WEAKLY FORMED PROCESSES // SCIENCE AND PRACTICE: IMPLEMENTATION TO MODERN SOCIETY MANCHESTER, GREAT BRITAIN, 26-28.12.2020.
6. D.Mukhamediyeva, N.Egamberdiev, ALGORITHM OF CLASSIFICATION OF MEDICAL OBJECTS ON THE BASIS OF NEUTROSOPHIC NUMBERS, Proceedings of the 4th International Scientific and Practical Conference SCIENCE, EDUCATION, INNOVATION: TOPICAL ISSUES AND MODERN ASPECTS TALLINN, ESTONIA, 4-5.10.2021, pp. 374-380.
7. Mukhamedieva D.T., Egamberdiev N.A., Zokirov J.Sh., Mathematical support for solving the classification problem using neural network algorithms // Turkish Journal of Computer and Mathematics Education. Vol.12 No.10 (2021).
8. D.T.Mukhamedieva and N.A.Egamberdiev, APPROACHES TO SOLVING OPTIMIZATION TASKS BASED ON NATURAL CALCULATION ALGORITHMS, Scientific-technical journal, 3(2) 2020, pp. 58-67.
9. N.A.Egamberdiev, O.T.Xolmuminov, X.R.Ochilov, ANALYSIS OF CLASSICAL MODELS OF CLASSIFICATION OF SLOWLY FORMED PROCESSES, International Scientific-Online Conference: SOLUTION OF SOCIAL PROBLEMS IN MANAGEMENT AND ECONOMY", Spain, October 7, 2022, pp.12-16.
10. N.A.Egamberdiev, O.T.Xolmuminov, X.R.Ochilov, CHOOSING AN EFFICIENT ALGORITHM FOR SOLVING THE CLASSIFICATION PROBLEM, International Scientific Online Conference: THEORETICAL ASPECTS IN THE FORMATION OF PEDAGOGICAL SCIENCES, October 10, 2022, pp.154-158.
11. D.Muhamedieva, N.Egamberdiev, O.Xolmuminov, APPLICATION OF ARTIFICIAL INTELLIGENCE TECHNOLOGIES FOR CREDIT RISK ASSESSMENT, «Science and innovation» xalqaro ilmiy jurnali. 2022, 6-soni. 388-395 betlar.
12. D.Muhamediyeva, N.Egamberdiyev, An application of Gauss neutrosophic numbers in medical diagnosis, International Conference on Information Science and Communications Technologies ICISCT 2021, Tashkent, Uzbekistan, 2021, pp. 1-4.
13. D.Muhamediyeva, N.Egamberdiyev, A.Bozorov, FORECASTING RISK OF NON-REDUCTION OF HARVEST, Proceedings of the 2nd International Scientific and Practical Conference, SCIENTIFIC COMMUNITY: INTERDISCIPLINARY RESEARCH, Hamburg, Germany, 26-28.01.2021, pp.694-698.
14. Ф.Нуралиев, О.Нарзуллоев, Н.Эгамбердиев, С.Тастанова, V Международная научно-практическая конференция, RECENT SCIENTIFIC INVESTIGATION, Осло, Норвегия, 26-28 апреля, 2022, с. 447-451.
15. Мухамедиева Д.Т., Эгамбердиев Н.А., Подходы к решению задач оптимизации на основе алгоритмов природных вычислений, Фарғона политехника институти илмий-техника журнали, 2020, Том 24, №2, с. 75-84.

16. N.A.Egamberdiyev, M.M.Kamilov, A.Sh.Hamroyev, O‘quv tanlamalar uchun k-o‘lchovli fiksirlangan tayanch to‘plamlar tizimini aniqlash algoritmini ishlab chiqish, Muhammad al-xorazmiy avlodlari, 1(7) 2019, 45-48 betlar.
17. Xo‘jayev O.Q., N.A.Egamberdiyev, Sh.N.Saidrasulov, Sinflarga ajratish masalasini yechishda samarali usulni tanlash algoritmi, Axborotkommunikasiyalar: Tarmoqlar, Texnologiyalar, Yyechimlar. 1(49)2019. Har choraklik ilmiy-texnik jurnal, 39-43 betlar.