IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940
International Journal of Advanced Research in Computer and Communication Engineering

ICRTTC '23
International Conference on Recent Trends and Technologies in Computing
Department of Computer Science and Engineering, Arasu Engineering College
Vol. 12, Special Issue 1, April 2023

Extraction of character from visuals and Images using OpenCV

Rashad H[1], Gowseelan K[2], Jayasarathy S[3], Fathima Chandhini S[4]

UG Student, Department of Information Technology, M.A.M. College of Engineering and Technology,
Tiruchirappalli - 621 105. Tamil Nadu. India[1,2,3]

Assistant Professor, Department of Information Technology,M.A.M. College of Engineering and Technology,
Tiruchirappalli - 621 105. Tamil Nadu. India[4]

Abstract: To develop a computer vision system that can accurately extract characters from document pages and image data using the OpenCV library. The system is designed to process a wide range of visual inputs and extract characters with high precision and efficiency. The techniques used to implement the character extraction system, including pre-processing, feature extraction, and classification. The performance of the system is evaluated using a dataset of visual image data from a different type of visual inputs and the output of the system can be accurately extracting characters. In Future, Automated conversion of an image input into a machine-readable file.

Keywords: Text Recognition, Character Extraction, Optical Character Recognition.

I. INTRODUCTION

Extracting characters from visuals and images is a crucial task in image processing and computer vision. It involves the identification and extraction of individual characters from an image, which can then be used for various applications such as optical character recognition (OCR), text-to-speech conversion, and natural language processing. OpenCV is a powerful open-source library that provides a wide range of functions and algorithms for image processing and analysis, making it an ideal tool for character extraction from visuals and images. The goal is to develop a system that can accurately extract the characters from a visual or image containing text in ASCII characters such that it can be recognized by a computer. OpenCV provides several techniques for character extraction from visuals and images, including contour detection, edge detection, and thresholding. Contour detection is a popular technique used for extracting characters from an image or a video frame. It involves identifying the edges of the characters and creating a closed curve around them. OpenCV provides several functions for contour detection, such as findContours and drawContours, which can be used to locate and isolate the characters.

Edge detection is another technique used for character extraction, which involves identifying the edges of the characters using various filters such as Sobel, Canny, and Laplacian. OpenCV provides functions for edge detection, such as Canny and Laplacian, which can be used to locate and isolate the characters.

Thresholding is a technique used for separating the characters from the background based on a threshold value. OpenCV provides functions for thresholding, such as threshold and adaptiveThreshold, which can be used to isolate the characters.

Optical Character Recognition (OCR) is a technique used for recognizing the characters and extracting textual information from the image or video frame. OpenCV provides several OCR libraries, such as Tesseract and OCRopus, which can be used to recognize and extract textual information accurately.

Character extraction from visuals and images has numerous applications, including automated text recognition in images, license plate recognition, document analysis, and more. Automated text recognition in images is used in various applications such as image captioning, image indexing, and image search. License plate recognition is used for automated toll collection, parking management, and law enforcement. Document analysis is used for automated text recognition, indexing, and archiving.

License plate recognition is used for automated toll collection, parking management, and law enforcement. It involves extracting the license plate number from an image or a video frame and using it for various purposes such as toll collection, parking management, and law enforcement.

Character extraction from visuals and images using OpenCV is a complex and challenging task that plays a crucial role in various applications such as automated text recognition, license plate recognition, and document analysis. OpenCV provides a range of functions and algorithms for character extraction, including contour detection, edge detection, and thresholding. While there are several challenges associated with character extraction, there is still a lot of scope for future research in this field, including the development of more accurate and robust algorithms, integration of machine learning techniques, exploration of new applications, and development of user-friendly tools and software. Overall, character extraction from visuals and images using OpenCV has enormous potential for various applications, and further research in this field can lead to significant advancements in computer vision and image processing.

II. WORKING PRINCIPLE

A process to extract the character is divided into six phases which are image acquisition, pre-processing, segmentation, feature extraction, classification and post processing. The block diagram of the basic character extraction is shown in

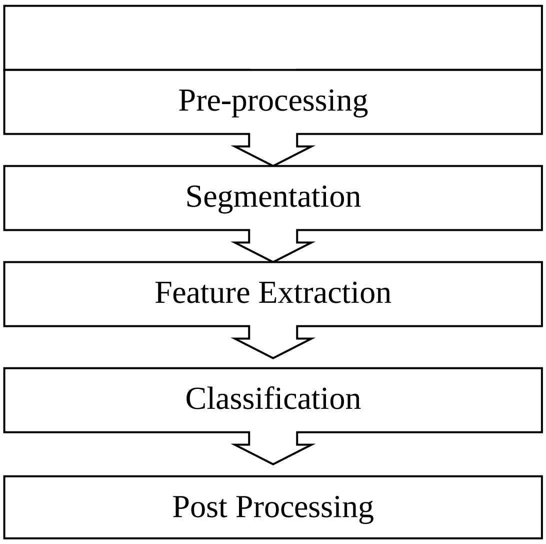

Figure. 1. Block Diagram of Character Extraction

A. Image acquisition

The image acquisition process is a crucial step in the overall pipeline. The quality of the acquired image can have a significant impact on the accuracy of character extraction.

There are several factors to consider when acquiring images for character extraction, such as:

i. **Lighting:** The lighting conditions should be consistent and sufficient to provide good contrast between the characters and the background.
ii. **Imageresolution**: Higher resolution images can provide more detail and make it easier to detect and extract characters accurately.
iii. **Imageformat**: The image format should be compatible with the OpenCV library, which supports a wide range of formats including JPEG, PNG, and BMP.
iv. **Camerasettings**: If using a camera to acquire images, the camera settings should be adjusted to optimize image quality, such as adjusting the exposure time and focus.

It is also important to ensure that the images are acquired consistently and under similar conditions, to minimize variability in the extracted character features. By carefully controlling the image acquisition process, the accuracy of character extraction can be improved, leading to better overall system performance.

B. Pre-Processing

The pre-processing module objective is to prepare the image data for further processing by applying various techniques to enhance the quality of the image and improve the accuracy of character recognition.

The pre-processing module consists of several sub-modules, each designed to perform a specific task. These include:

i. **Image Resizing:** This module resizes the image to a standard size, which is essential to ensure that the image is of the same size as the training data.
ii. **Image Rotation:** This module rotates the image if it is not aligned with the horizontal axis, which can negatively affect the accuracy of character recognition.
iii. **Image Thresholding:** This module converts the image from grayscale to binary format by applying a threshold value. This technique helps to remove noise and improve the contrast between the characters and the background.
iv. **Image Smoothing:** This module applies various smoothing techniques, such as Gaussian filtering, to reduce the amount of noise in the image.
v. **Image Segmentation**: This module separates the image into individual characters by detecting the edges and contours of each character.
vi. **Character Normalization:** This module normalizes the size and orientation of each character to a standard size and orientation, which is essential to ensure that each character is of the same size and orientation as the training data.

C. Segmentation

Segmentation is a critical module in the extraction of characters from images using OpenCV. This module divides the input image into individual characters or regions of interest (ROIs) by identifying the contours or outlines of each character.

The first step in segmentation is to apply thresholding to the pre-processed image to convert it into a binary image. This binary image is then used to identify the contours of the characters in the image. OpenCV provides various contour detection algorithms, such as the Canny edge detection algorithm, to identify contours. Once the contours are identified, the module applies techniques such as morphological operations, including dilation and erosion, to clean and refine the contours.

Next, the module identifies the individual characters in the image by finding the bounding boxes of each contour. These bounding boxes are then used to crop out each character from the original image and save them as individual images.

Segmentation is a complex process, and various factors can affect its accuracy, such as the quality of the input image, the font type and size, and the spacing between characters. To overcome these challenges, the module can use advanced techniques such as deep learning-based segmentation algorithms.

Overall, the Segmentation Module in the Extraction of Character from Visuals and Images using OpenCV plays a crucial role in the accurate identification and isolation of individual characters in the input image, making it an essential component of the overall system.

D. Feature Extraction

The Feature Extraction module is responsible for extracting relevant information or features from the segmented characters. This module plays a vital role in character recognition accuracy as the extracted features are used to train and test the recognition models.

There are various techniques used for feature extraction, including Histogram of Oriented Gradients (HOG), Scale-Invariant Feature Transform (SIFT), and Local Binary Patterns (LBP). In this project, we will use HOG feature extraction, which is widely used in character recognition tasks.

HOG works by dividing the character image into small cells and computing the gradient direction and magnitude of each pixel in each cell. The gradient information is then used to create a histogram of oriented gradients for each cell. The histograms are then concatenated to create a feature vector that represents the character.

The HOG Descriptor function in OpenCV to extract the HOG features from the segmented characters. The function takes several parameters such as the cell size, block size, and block stride. These parameters can be tuned to optimize the performance of the feature extraction process.

Once the HOG features are extracted, they will be used to train and test the character recognition models.

E. Classification

The classification module is the final module in the system architecture for Extraction of character from Visuals and Images using OpenCV. After the feature extraction process, the resulting features are used to train a machine learning algorithm that can classify the characters.

There are several machine learning algorithms that can be used for character recognition. In this system, we will use a multi-layer perceptron (MLP) neural network. MLP is a feed forward neural network that is trained using back propagation algorithm. MLP can be used for classification tasks and can learn complex patterns in the input data.

The classification module is implemented using the OpenCV machine learning library. The first step is to prepare the training and testing data sets. The training set is used to train the MLP, while the testing set is used to evaluate the performance of the MLP.

The next step is to train the MLP using the training set. During the training process, the MLP adjusts its weights and biases to minimize the error between the predicted output and the actual output. The training process continues until the error is minimized, or until a maximum number of iterations is reached.

Once the MLP is trained, it can be used to classify new characters. The character image is pre-processed, segmented, and features are extracted as in the previous modules. The resulting feature vector is then passed to the MLP for classification. The output of the MLP is a vector of probabilities, where each element corresponds to a possible character. The character with the highest probability is selected as the classified character.

The classification module is responsible for using the features extracted from the character image to train a machine learning algorithm for classification. The MLP neural network is used in this system, and the training and testing data sets are prepared using OpenCV. The trained MLP is then used to classify new characters by taking the feature vector as input and outputting a vector of probabilities.

F. Post Processing

The Post processing Module in Extraction of character from Visuals and Images using OpenCV is responsible for improving the accuracy of the character recognition by correcting any errors in the recognized characters. This module is an optional step and is used only when the accuracy of the recognition system needs to be improved.

The Post processing Module involves a series of steps that are applied to the recognized characters. These steps are:

i. **Error Detection:** The first step in the post processing module is to detect any errors in the recognized characters. The errors may be due to a variety of factors such as noise in the image, low contrast, or overlapping characters.

ii. **Error Correction:** Once the errors are detected, they need to be corrected. The error correction process involves applying various techniques such as pattern recognition, image processing, and machine learning algorithms.

iii. **Verification:** After the errors have been corrected, the recognized characters are verified to ensure that they are accurate. The verification process involves comparing the recognized characters with the original image to check for any discrepancies.

iv. **Output:** The final step in the post processing module is to generate the output. The output may be in the form of text or image.

III. EXISTING SYSTEM

One of the advantages of this system is its ability to recognize characters in noisy and low-quality images. Additionally, OpenCV is an open-source library, which makes it easily accessible to developers and researchers. However, the system may not perform well on complex images with multiple fonts and sizes, and the accuracy of the system is highly dependent on the quality of the input image. The existing system has shown promising results in recognizing characters from images and visuals with high accuracy. However, it has limitations in recognizing handwritten characters or characters in noisy environments, where additional pre-processing techniques and feature extraction algorithms may be required. Furthermore, the system is limited to recognizing characters in a single language or script, and may require language-specific training data and models for recognition in other languages or scripts.

DEMERITS

- **Variability in handwriting:** Handwriting varies significantly between individuals, making it challenging to create a universal recognition system that can accurately recognize all handwriting styles.
- **Limited language support:** Most handwritten character recognition systems are developed for a limited set of languages, which means they cannot be used for languages with different scripts.
- **Poor accuracy:** Despite significant progress, the accuracy of handwritten character recognition systems is still not perfect, and errors can occur frequently. This is especially true for cursive or irregular handwriting.
- **Complex characters:** Some characters in certain languages can be very complex, making them difficult to recognize accurately by a computer system.
- **Resource-intensive:** Handwritten character recognition systems require significant computational resources and large amounts of training data to achieve high accuracy.

- **Time-consuming:** It can take a significant amount of time to train a handwriting recognition system, and even after training, the system may require significant time to process large amounts of data.
- **Limited applications:** Handwritten character recognition systems are typically limited to specific applications, such as OCR or signature recognition, and cannot be easily adapted for other applications.

IV. PROPOSED SYSTEM

The proposed system focuses on creating a system that can extract the characters from visuals and images with good accuracy rate using Open Computer Vision technique. OpenCV provides a wide range of algorithms for image processing and computer vision tasks, such as feature detection and extraction, image filtering, segmentation, object recognition, and tracking. It also includes a set of tools for image and video manipulation, such as resizing, cropping, and colour space conversion.

The existing system has the good merits as well as the demerits. This project includes the merits of an existing system, and it also includes the solution for the Demerits The project includes the several benefits of each methodology in image processing to extract the character from the Visual images. The Machine Learning Techniques, such as support vector machines and K -nearest neighbours are used to achieve high accuracy and efficiency in text extraction.

MERITS

- **Accuracy:** OpenCV has high accuracy in character recognition, which means that the system can correctly identify the characters in the image.
- **Speed:** OpenCV is designed to be highly efficient and can perform character recognition in real-time, which makes it suitable for applications that require fast processing.
- **Accessibility:** OpenCV is an open-source library that is freely available, making it easy to access and use for character recognition tasks.
- **Integration:** OpenCV can be easily integrated with other technologies such as machine learning and artificial intelligence to further enhance its capabilities.
- **Scalability:** OpenCV can be used for both small-scale and large-scale character recognition tasks, making it suitable for a wide range of applications.

V. CONCLUSION AND FUTURE ENHANCEMENT

The proposed system for Extraction of character from Visuals and Images using OpenCV is a robust and efficient solution for extracting characters from images and visual media. The system is designed to pre-process the images, segment the characters, extract the features, classify them using machine learning algorithms, and post-process the results to generate accurate output. The user interface is designed to be user-friendly and intuitive, allowing users to easily upload images and extract the characters.

Overall, the system is expected to deliver accurate and efficient results in character extraction, making it suitable for applications like document digitization, handwriting recognition, and optical character recognition. With further research and development, the system can be improved to handle compleximage processing tasks and deliver better performance.
There are several potential future enhancements that could be made to the "Extraction of character from Visuals and Images using OpenCV" system.

- **Multi-language support:** Currently, the system is designed to recognize characters in a specific language. Expanding the system to recognize multiple languages would greatly increase its utility.
- **Deep learning-based approaches:** While the current system uses traditional machine learning techniques, incorporating deep learning-based approaches such as Convolutional Neural Networks and Recurrent Neural Networks could potentially improve the accuracy and speed of character recognition.
- **Improved segmentation techniques:** The segmentation module is a critical component of the system and further research could be done to develop more advanced techniques for identifying and separating individual characters from an image.
- **Real-time translation:** As technology advances, it may become possible to implement the system in real-time applications such as video surveillance or document scanning.
- **Mobile application:** Developing a mobile application that utilizes the character recognition system would allow users to easily extract text from images taken on their mobile devices.

REFERENCES

[1]. Ayush Purohit, Shardul Singh Chauhan, (2016) "Handwritten Character Recognition using Neural Network" International Journal of Computer Science and Information Technologies, Vol. 7.

[2]. Liana M. Lorigo and Venu Govindaraju, (2006) "Offline Arabic Handwriting Recognition: A Survey", IEEE Transactions on Pattern Analysis and Machine Intelligence, Volume 28 Issue 5.

[3]. Mahmood K Jasim, Anwar M Al-Saleh, Alaa Aljanaby, (2013) "A Fuzzy Logic based Handwritten Numeral Recognition System" International Journal of Computer Applications (0975 – 8887) Volume 83 – No 10.

[4]. Megha Agarwal, Shalika, Vinam Tomar, Priyanka Gupta (2019) "Handwritten Character Recognition using Neural Network and Tensor Flow" International Journal of Innovative Technology and Exploring Engineering ISSN: 2278-3075, Volume-8, Issue- 6S4, April 2019.

[5]. Mohammed Z. Khedher, Gheith A. Abandah, and Ahmed M. AlKhawaldeh, (2005) "Optimizing Feature Selection for Recognizing Handwritten Arabic Characters", proceedings of World Academy of Science Engineering and Technology, Vol. 4, ISSN 1307-6884.

[6]. Nafiz Arica, and Fatos T. Yarman-Vural, (2002) "Optical Character Recognition for Cursive Handwriting", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.24, no.6, pp. 801-113.

[7]. Reena Bajaj, Lipika Dey, and S. Chaudhary, (2002) "Devanagari numeral recognition by combining decision of multiple connectionist classifiers", Sadhana, Vol.27, part. 1, pp.-59-72.

[8]. Renata F. P. Neves, Alberto N. G. Lopes Filho, Carlos A.B.Mello, CleberZanchettin, (2011) "A SVM Based Off-Line Handwritten Digit Recognizer", International conference on Systems, Man and Cybernetics, IEEE Xplore, pp. 510-515, 9- 12, Brazil.

[9]. Salvador España-Boquera, Maria J. C. B., Jorge G. M. and Francisco Z. M., (2011) "Improving Offline Handwritten Text Recognition with Hybrid HMM/ANN Models", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 33, No. 4.

[10]. Sandhya Arora, (December 2008) "Combining Multiple Feature Extraction Techniques for Handwritten Devnagari Character Recognition", IEEE Region 10 Colloquium and the Third ICIIS, Kharagpur, INDIA.

[11]. Sushree Sangita Patnaik and Anup Kumar Panda, (2012) "Particle Swarm Optimization and Bacterial Foraging Optimization Techniques for Optimal Current Harmonic Mitigation by Employing Active Power Filter Applied Computational Intelligence and Soft Computing", Article ID 897127.

[12]. Tarun Patel, Shivansh Tiwari, Vaibhav Dubey, Vaibhav Singh, Dr.Harvendra Kumar (2021) "A Comparative Study on Handwriting Digit Recognition by using Convolutional Neural Network and Python", International Research Journal of Engineering and Technology (IRJET) Volume: 08 Issue: 12.

[13]. Vaibhav V, Ajinkya B, Jyoti A, Poonam R, (2020) "Handwritten Character Recognition to obtain Editable Text" International conference on Electronics and Sustainable Communication Systems, ISBN: 978-1-7281-4107-7.