

OCR Model for Reading Captcha

Abstract:

The focus of this project is to compare several common machine learning classification algorithms for Optical Character Recognition of CAPTCHA codes. The main part of a research focuses on the comparative study of Neural Networks, k-Nearest Neighbour, Support Vector Machines and Decision Trees implemented in MATLAB Computing environment. Achieved success rates of all analysed algorithms overcome 89%. The main difference in results of used algorithms is within the learning times. Based on the data found, it is possible to choose the right algorithm for the particular task.

Existing System:

CAPTCHA, A Completely Automated Public Turing test to tell Computers and Humans Apart, is a well-known system widely used in all sorts of internet services around the world designated secure the web from an automatic malicious activity. For almost two decades almost every system utilized a simple approach to this problem containing a transcription of distorted letters from image to a text held. The ground idea is to use imperfection of Optical Character Recognition algorithms against the computers.

The development of Optical Character recognition algorithms leads only to state, where the CAPTCHA schemes become more complex and human users have a great difficulty with the transcription. This paper aims to present a new way of development of CAPTCHA schemes based more a human perception. The goal of this work is to implement new Captcha scheme and assess human capability to read unusual fonts never seen before.

Proposed System:

The recognition of handwritten or printed text by computer is referred to as O[ptical] C[haracter] R[ecognition]. When the input device is a digitizer tablet that transmits the signal in real time (as in pen-based computers and personal digital assistants) or includes timing information together with pen position (as in signature capture) we speak of dynamic recognition. When the input device is a still camera or a scanner, which captures the position of digital ink on the page but not the order in which it was laid down, we speak of static or image-based OCR.

In the recognition of hand print, algorithms with successive segmentation, classification, and identification (language modelling) stages are still in the lead. For cursive handwriting, → Hidden Markov Models that make the segmentation, classification, and identification decisions in parallel have proven superior, but performance still leaves much to be desired, both because the spatial and the temporal aspects of the written signal are not necessarily in lock-step (discontinuous constituents arising e.g. at the crossing of t-s and dotting of i-s) and because the

inherent variability of handwriting is far greater than that of speech, to the extent that we often see illegible handwriting but rarely hear unintelligible speech.

Software Tools:

1. VS Code
2. Jupyter Notebook
3. Colab
4. TensorFlow
5. Keras
6. OpenCV
7. Python3
8. Matplotlib

Hardware Tools:

1. Laptop
2. Operating System: Windows 11
3. RAM: 16GB
4. ROM: 4GB
5. Fast Internet Connectivity
6. GPU

Applications:

1. OCR Model for Blind
2. This kind of architecture can be implemented as an option for accessibility in mobile phones or laptops.