International Journal of Computing and Digital Systems
ISSN (2210-142X)
Int. J. Com. Dig. Sys. 12, No.1 (Dec-2022)

https://dx.doi.org/10.12785/ijcds/1201114

Efficient CRNN Recognition Approaches for Defective Characters in Images

Hashem Al-Nabhi [1], K.Lokesh Krishna [2] and Ahmed Abdullah A. Shareef [3]

[1] *Department of Electronics and Information, Northwestern Polytechnical University, Xian City, China*
[2] *Department of ECE, S.V. College of Engineering, Tirupathi, A.P., India*
[3] *Department of Computer Science and IT, Dr. Babasaheb Ambedkar Marathwada University, Aurangabad, India.*

Received 10 Feb. 2021, Revised 7 Aug. 2022, Accepted 1 Dec. 2022, Published 8 Dec. 2022

Abstract: Defective Characters exist frequently and broadly in images such as license plates, electricity, water meters, street boards, etc. Thus, building robust recognition systems or enhancing the accuracy and robustness of the existing recognition systems to recognize such characters on images is a challenging research topic in image processing and computer vision. This paper Investigates and adopts ReId dataset for all the experimental work and introduces two deep learning models (CNN5-BLSTM and CNN7-GRU) based on convolutional recurrent neural networks (CRNN) to address the problem of defective characters sequence recognition. The two proposed deep learning models are segmentation-free, lightweight, End-To-End trainable, and slightly different from each other. The models are evaluated on testing data of ReId dataset, and the achieved accuracies are 95% of characters' sequence accuracy and 98% of character-level accuracy. Moreover, their performance on ReId dataset outperforms other models' performance in the literature.
.

Keywords: Deep Learning, Image Processing, Characters Sequence recognition, Defective Characters, Convolutional Recurrent Neural Networks, Convolution Neural Networks, Connectionist Temporal Classification, Gated Recurrent Unit, Long Short-Term Memory.

1. Introduction

Image-Based Characters Sequence Recognition System is essentially a digital system that uses Optical Character Recognition (OCR) Technology to recognize characters sequence (A-Z, a-z, 0-9) in images. These images can be vehicle number plates, streets' nameplates, expiry dates of products, characters on electricity meters, and so on. The system takes the image that contains a sequence of characters – usually taken by a camera- as an input and generates an output which is in the form of a text -or- a string representing the characters written in the input image.

Recently, Recognizing of Image-Based Characters Sequence has become a critical and challenging research topic on AI field particularly on Image processing and Computer Vision because of the difficulties and challenges in recognizing the characters present on low-quality images such as low-quality license plates, moreover, the existence of the flawed characters on the images can be another challenge for recognition systems. Thus, this optical character recognition problem is still under the sight and the investigation of many researchers and scholars in the area of image processing and computer vision.

The ability to build robust recognition systems such as License plate recognition (LPR) systems that can extract the alphanumeric characters sequence and other information from the input images captured by a camera and store all the extracted information as characters sequence inside computers contributed massively in building intelligent systems such as Intelligent Traffic System (ITS) that becomes one of the popular core technologies and services in almost all the developed and modern cities and countries because of its various services and valuable applications.

There are many other applications for characters' sequence recognition systems. It can be used in the parking system to collect the parking fees, protect vehicles from stealing, and provide each car's position. The police can also use it for security, such as checking whether the vehicle is licensed or not, inspecting the number plates of the vehicles that break the traffic law, and saving information about the vehicle such as vehicle image, vehicle color, vehicle license, and its owner. In addition to the mentioned applications, characters' sequence recognition Systems can be used to control the borders between the counties by monitoring and enhancing the level of security inside the country against illegal activities such as smuggling, terrorism, unlawful entrance to the country, and some other illegal activities. It significantly contributed to reducing the number of crimes and increases the levels of safety inside the countries. Another prominent and useful applications of character sequence recognition systems in daily life, they are used in many governmental and private sectors such as airports for extracting passport numbers, in the universities for reading student ID numbers, in the banks for reading bank statements and receipts. Furthermore, these systems are used to encode invoices, read handwritten and printed texts, inspect the expiry dates on the products, and to read the electricity and water meters.

2. Related Work

Extracting characters' sequences or texts from images is an early research problem as many researchers tried to address it in many research works. There are different types of algorithms used to solve this research dilemma. The characters' irregular arrangements on images (seriously distorted, curved, arbitrarily-oriented, various lengths, etc.) make their recognition more difficult and challenging [1]. This part of the paper discusses different methods (algorithms) used to address the characters' sequence recognition problem, and these methods are classified into two categories illustrated as follows.

A. *Segmentation-Based Approach*

This part of the paper discusses the recognition systems based on segmentation algorithms. Note that the majority of the existing systems on image-based characters' sequence recognition require two phases of processing to extract the characters' sequence from the input image: The first phase is the segmentation phase which segments all characters that appear on the input image. The second phase is the character recognition phase which recognizes the segmented characters in the first phase.

Researchers proposed different segmentation-based algorithms to segment the characters' sequence on images, for instance, the method ”projection segmentation analysis” used in [2],[3]. This method is considered the most straightforward process mathematically. In this method, image-based character sequence is converted into a binary image (black and white). Then, the vertical and horizontal projections can be obtained by adding image columns and rows. Each character's segmentation can is done by utilizing the minimum values on the vertical projection, representing the spaces between characters. It is used to segment the characters along the horizontal direction, and the minimum values on the horizontal projection used to locate the upper and lower boundaries of the characters.

Connected Components Analysis (CCA) is another method based on character segmentation [4],[5]. This method is widely used in image-based text recognition systems, image-based characters' sequence recognition systems, and license plate recognition systems. In this method, the input image is converted into a binary image (black and white). The binarized image pixels are labelled based on the algorithms of either 8-neighborhood connectivity or 4-neighborhood connectivity. Extremal Region (EM) Method and Maximally Stable Extremal Region (MSER) Method are adopted for character segmentation in [6],[7]. However, the latter method has been proven to be more efficient than the former one.

The sliding window method is another character segmentation method utilized in [8] for character segmentation. This approach is used to capture the input image characters using a small window sliding over the image. The captured characters are then recognized using optical character recognition by choosing the highest count character consecutively. This method is adopted in [9] to improve the recognition accuracy of Arabic characters. Hence, this method can achieve high accuracy for character-level recognition.

The segmentation-based method proposed in [10] is similar to the sliding window search method but showed more robustness. This approach combined the Inception Net and Region proposal Network, and the experimental results showed that this approach achieved state-of-the-art results in segmentation and text detection. In addition to the previously mentioned approaches, many other traditional digital image processing approaches are used for character segmentation, such as Hidden Markov model utilized in [11] and the Hypothesis Generation Method introduced in [12],[13].

Segmentation of characters on images can be achieved using one segmentation algorithm or a combination of two or more algorithms. For example, the segmentation method proposed in [14] is based on morphological operations and connected components analysis to perform segmentation on license plates. The segmentation is achieved by Firstly applying erosion and dilation to the image. Then the image is subtracted, and the holes in the resulting image are filled. Secondly, removing undesired small components from the image. Finally, the remaining characters represent the outcome of segmentation. However, most traditional segmentation techniques have many drawbacks, such as high computational complexity and can't extend to work on other formats of license plates and characters' sequences.

Moreover, the segmentation techniques' performance on degraded images based on characters' sequence is not satisfactory because segmentation algorithms fail to segment the characters on noisy images or images with different illumination levels. The robustness of the segmentation part is crucial in such recognition systems because the segmentation part's failure can greatly affect the recognition part even if it is robust.

The character recognition algorithm follows the character segmentation algorithm to recognize and classify the segmented characters. In the recent years, researchers have developed and implemented different recognition algorithms to identify and recognize the characters in the segmented images. Such recognition algorithms are common for instance, the ”Template Matching Algorithm” used in [15]. This algorithm is used to compare the images of input characters with template characters, and then calculate the similarities between them. After that, the recognition result is the template character with the highest similarity with the input character. ”Feature Matching” algorithm is another recognition algorithm proposed in [16]. This algorithm compares the extracted character features with the standard character features. Based on the similarity between the features, the output is decided. This method is very complicated, tiring, and time-consuming. Moreover, the standard features of each character need to be done manually. Nevertheless, it is more efficient and reliable compared to the template matching method.

Segmentation-based recognition systems contribute big to the area of characters’ sequence recognition by introducing partial solutions to address characters’ sequence recognition on images. However, these solutions failed to address flawed character recognition on images due to the drawbacks of the segmentation process.

B. Segmentation-free Approach

The drawbacks of the recognition systems based on traditional digital image processing (DIP) techniques or segmentation-based techniques still exist and affect recognition systems’ accuracy. Therefore, researchers in image processing and computer vision areas hardly worked to find alternative solutions based on machine learning algorithms. Thus many models based on deep learning algorithms using Convolutional Neural Network (CNN) are developed; for instance, the deep learning model based on Convolutional Neural Networks (CNNs) proposed in [17] for Chinese license plate character recognition. This model consists of seven layers, three convolutional layers, and two max-pooling layers for feature extraction. The next two layers are fully connected layers used for character classification.

Another approach-based character recognition is introduced in [18] for car plate character recognition called SHL-CNN (Shared Hidden Layers CNN). SHL-CNN consists of some shared hidden layers for feature extraction and two softmax layers for character classification, one of the softmax layers is used for Chinese characters classification, and the other one is used for alphanumeric characters classification. Furthermore, the characters recognition module proposed in [19] uses two Artificial Neural Networks (ANNs), one of them is used for letter recognition, and another one is used for numbers recognition. Using two distinctive networks helps the network differentiate between similar letters and numbers such as B-8, O-0, Z-2, etc.

Two models are proposed in [20] for flawed character recognition on low-quality images. The models are developed using convolutional neural networks, and both proposed models showed almost the same performance despite the difference in the number of CNN layers and number of trainable parameters. This approach’s drawback is that it cannot be extended to recognize the characters’ sequence of arbitrary lengths.

A robust approach based on SHL-CRNN proposed in [21] is used to recognize the flawed characters in the form of arbitrarily-oriented texts. Similarly, a novel model proposed in [22] to recognize the distorted scene text (arbitrary-oriented texts) in images, and for this problem a word-level encoder and a character-level encoder are proposed for feature extraction and encoding, and LSTM-based decoder is used for decoding the extracted features. These models achieved state of the art accuracy; however, they turned out to be very large models.

The approach proposed in [23] for scene text recognition combined both CNN and RNN networks as a unified network trained End-To-End on synthetic text datasets under one loss function. This approach achieved very competitive and remarkable accuracy on scene text recognition. TextScannar in [24] is another method proposed for scene text recognition which can generates predictions for character position and class. This method utilizes semantic segmentation for pixel-wise generation, and RNN for the text modelling.

A selective context attention recognizer proposed in [25] which made use of a superposition of five layers of bidirectional LSTM to enhance the performance of the recognition system. To address the problem attention drift, a decoupled text decoder introduced in [26] and in which a decoder was decoupled with a traditional attention mechanism into a convolutional alignment module. Moreover, there are many methods based on self-attention [27],[28] were utilized to address the problem of scene text recognition (STR) and achieve satisfactory accuracy and better performance.

The problem of defective characters’ sequence recognition has not been studied and investigated by many researchers. Therefore, this paper highlights this problem and proposes two lightweight, segmentation-free, End-To-End trainable deep learning models based on convolutional recurrent neural networks. CNN7-GRU and CNN5-LSTM are the two proposed models that significantly achieved excellent recognition accuracy in predicting the characters’ sequence that appears on the low-quality license plates. The models’ achieved accuracy is 98 of characters’ level accuracy and 95 of characters’ sequence accuracy. Moreover, the performance of the proposed models outperformed other models’ performance in the literature.

The paper’s remaining parts are organized as follows: Sect 3 illustrates the system components and their main functions. Sect 4 describes the experimental results, Error analysis and results comparisons. Section 5 discusses the conclusion and future work.

3. Proposed Scheme

A. Network Architecture

This section of the paper illustrates and discusses in detail the architecture of the proposed CRNN Networks and the main components of the networks. The Models' novelty is that the proposed CRNN models achieved better accuracy in recognizing defective characters' sequences than other works in the literature. Moreover, the proposed models are lightweight in terms of the number of trainable parameters in the whole network. In most previous works, the recognition systems' components such as character segmentation network and character recognition network are trained and fine-tuned separately. Moreover, every network needs to be trained on different datasets; thus, it is a very time-consuming and expensive process. This work focuses on building unified deep learning models, particularly recognition models that integrate the feature extraction network (Feature Encoding Network) and the character recognition network (Feature Decoding Network) by training both networks End-To-End under one loss function utilizing the same dataset.

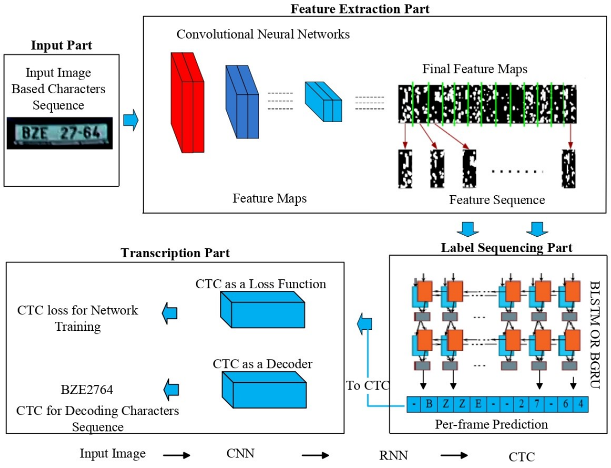


Figure 1. The architecture of CRNN Network

Two deep learning models based on Convolutional Recurrent Neural networks (CRNN) are developed to address the problem of defective characters' sequence recognition on images. The architecture overview of the proposed networks is depicted in Figure 1. The proposed recognition systems consists of three main parts named as follows, Feature extraction part, Label sequencing part, and the Transcription part. The first part of the system is the encoder part which is represented by Convolutional Neural Network (CNN) layers that are used to extract the features from the input image-based characters' sequence and encode the extracted features to a feature sequence. The label sequencing part is represented either by bidirectional Long Short-Term Memory (LSTM) or Bidirectional Gated Recurrent Unit (GRU) that are used to propagate the information through the network and make per-frame predictions for the extracted feature sequence. The decoding (Transcription) part is represented by the Connectionist Temporal Classification (CTC) that is very essential for the network training and decoding the predicted characters from the output of BLSTMs and BGRUs utilizing the conditional probability distribution function concept. Therefore, both designed models are trained End-To-End using the CTC loss function that can handle the characters' sequences of variable lengths. For models' optimization, Adadelta Optimizer is adopted to update the weights through the back-propagation process.

B. Feature Extraction Overview

Convolutional Neural Networks (CNNs) are powerful feature extraction tools because the potent kernels of these networks can extract specific and desired features from the input image. Thus, these networks are utilized to build many robust deep learning models of excellent performance and competitive accuracy. In this experimental work, the VGG network's basic idea is utilized to design lightweight feature extraction encoding networks to extract the features of characters' sequence and texts on input images. Each convolutional layer in the network consists of a different number of kernels, and the number of kernels increases as the neural network's depth increases. The higher number of kernels, the more in-depth features get extracted. Note that each convolutional layer is followed by batch normalization and rectified linear unit (ReLU).

Two well-designed Convolutional Neural networks (CNNs) are used for feature extraction. The first network (CNN-5) consists of five groups of layers, and the second network (CNN-7) consists of seven groups of layers, as shown in Figure 2. Each group consists of at least one Convolutional Layer (Conv2D), Batch Normalization Layer (BatchNorm), and Non-linear Rectified Unit (ReLU). In both designed networks, each group is followed by Max Pooling Layer (MaxPool) except group 3 in the first network (CNN-5) and groups 3, 4 in the second network (CNN-7). In CNN-5, each convolutional layer consists of 3×3 kernels, except the convolutional layer in group 5 that consists of 2×2 kernel. Similarly, each max-pooling layer utilizes a 2×2 kernel, except the max-pooling layer that follows group 4 uses a 1×2 kernel.

In CNN-7, each convolutional layer consists of 3×3 kernels, except the convolutional layer in group 7 consisting of 2×2 kernel. Similarly, each max-pooling layer utilizes a 2×2 kernel except the max-pooling layer that follows groups 4 and 6, and it uses a 1×2 kernel. In the first CNN architecture, the number of filters in the first layer is 64, in the second layer is 128, in the third layer and the fourth layer is 256, in the fifth layer is 512. In CNN7 architecture, the number of filters in the first layer is 64, in the second layer is 128, in the third and fourth layer is 256, in the fourth and fifth layers is 256, and in the sixth and seventh layer is 512. Each convolutional layer has a stride of 1, padding of 1 on both sides of the image, and different kernels' size. We intended to use the "same convolution" type utilizing

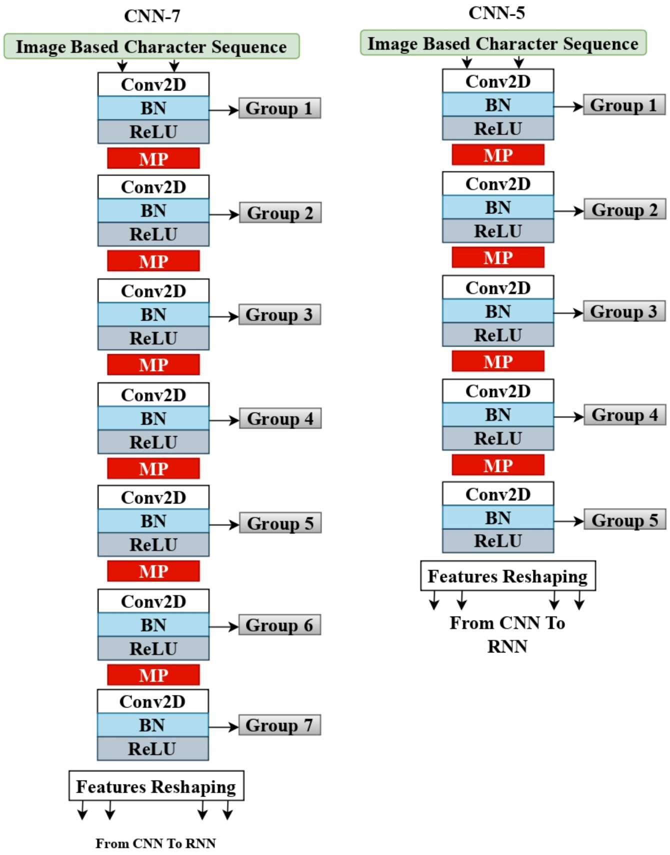


Figure 2. CNN Networks for Feature Extraction

a padding algorithm to keep the height, and the width of the feature maps the same after each convolutional layer, and all the meaningful features will be preserved. Rectified linear unit (ReLU) is an activation function used after each convolutional operation to perform element-wise activation on the feature maps and introduce the non-linearity to the network; hence the network can efficiently learn. Rectified Linear Unit (ReLU) is mathematically written as follows:

$$f(x) = max(0, x) \tag{1}$$

Where x stands for the activation of the extracted feature maps, Max pooling layers are used to decrease the feature maps' dimensionality, accelerating the training process. In our CNN-5 network and CNN-7 network, we used 3 max-pooling layers and 4 max-pooling layers respectively to decrease the width and the height of the feature maps. In CNN-5 and CNN-7, MaxPool1, MaxPool2 are used to reduce the height and the width of the feature maps by half. But, MaxPool3 in CNN-5 and Maxpool3, MaxPool4 in CNN-7 are used to decrease the feature maps' height by half while maintaining the width of the feature maps unchanged. Batch Normalization layers are used after each convolutional layer in the network to accelerate the training process, stabilize it, and reduce the training epochs. Batch Normalization is achieved by normalizing the output activations of the previous convolutional layers.

C. Feature Extraction Process

Convolutional Neural Network (CNN) layers, with the help of their powerful kernels can extract and encode the features of characters' sequence that appear on the input image by transforming the original image-based characters sequence into a dense stack of many feature maps. The extracted feature maps are mapped to a sequence of feature vectors using a designed layer at the top of the CNN layer called the map2seq layer, as shown in Figure 3. Groups of feature vectors are called a feature sequence, and each feature vector in a feature sequence is equal to all corresponding concatenated columns in the feature maps. On the other hand, each feature vector in a feature sequence corresponds to a rectangular region in the feature maps. Thus, it corresponds to a rectangular region in the original image. Note that all the feature vectors are of the same order with their corresponding rectangular regions in the original image from left to right, as shown in Figure 4.

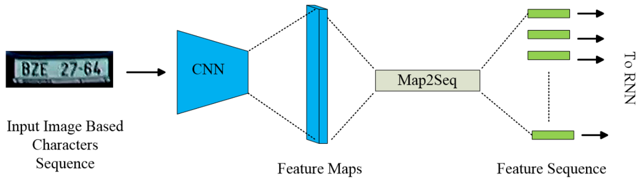


Figure 3. Mapping Feature Maps to a Feature Sequence

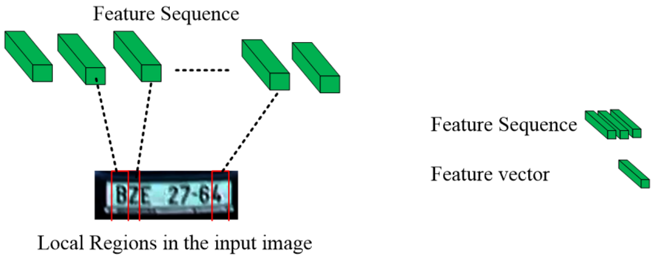


Figure 4. Mapping Feature Maps to a Feature Sequence

D. Recurrent Neural Network Layer

The features extracted from the image-based defective characters' sequence are transformed from feature maps to a feature sequence of stacked vectors. Each feature vector represents a rectangular region on the input image-based characters sequence. The vectors of feature sequence are given as an input to the recurrent neural networks (RNNs) used to propagate information through the sequence. Recurrent Neural Networks (RNNs) are built on the top of convolutional Neural Networks (CNNs) using a designed layer that is used to map the outputs of CNNs to the inputs of the RNN network. The features sequence inputted to RNNs consists of several frames, and each frame is a vector representing a stack of deep columns in the feature maps. Recurrent Neural Networks are used to make a prediction

$$y'_t \,(y'_t = y'_1, y'_2, \dots y'_T) \tag{2}$$

for each frame χ_t in the feature sequence. In traditional RNNs, each unit consists of a hidden state h_t that can be updated once it receives input χ_t and the previously hidden state information h_{t-1} ; on the other hands, a hidden state h_t is a function of the current input, and the previously hidden state output, and the output y'_t will be the function of h_t as shown in the following equations.

$$h_t = g\left(x_t, h_{t-1}\right) \tag{3}$$

$$y'_t = f\left(h_t\right) \tag{4}$$

$$h_t = \tanh\left(b_h + W_h^t h_{t-1} + W_x^t x_t\right) \tag{5}$$

$$h_t = \tanh\left(b_h + W_h^t h_{t-1} + W_x^t x_t\right) \tag{6}$$

The output y'_t of the unit depends directly on the hidden states h_t and indirectly on the previous hidden state h_{t-1} , so RNNs can utilize and capture past information; however, Traditional RNNs suffer from vanishing gradient problems. Instead, we have employed two bi-directional LSTMs, and two bi-directional GRUs stacked together to capture the context of image-based characters' sequence from forward and backward. This property will enhance the performance and the rate of learning of the recognition system. The information propagates through the units of bi-directional LSTMs and bi-directional GRUs, and eventually output a matrix consisting of probabilities for each character in the input sequence. The outputted matrix is given to the CTC function either for training (calculating the loss) or for making predictions. Note that, at the stage of training the neural networks, the overall loss is calculated and the weights are getting updated through the back-propagation process using Adadelta optimizer. Moreover, the error differentials are propagated through time in opposite directions of forwarding information flow. The process of Back-propagation in RNNs is called Back-propagation through time. During the back-propagation operations, the back-propagated differential sequence at the bottom of LSTMs and GRUs reconstructs the feature maps in the Feature extraction part to update the weights and reduce the errors.

E. Decoding Characters by CTC

CTC's operation is segmentation-free and doesn't care about the exact position of characters sequence on the input image. CTC algorithm performs two main operations during the training process and testing process. During the model's training operation, it acts as a loss function and it is fed with the matrix of per-frame predictions (output of BLSTM or BGRU units), the ground truth characters, the length of the input characters' sequence, and the length of ground truth characters' sequence. During the model's testing operation, the CTC algorithm acts as a decoding function. For this purpose, it requires the output distributions of BLSTM or BGRU units and the length of the input characters' sequence. CTC function transcribes the characters from BLSTM and BGRU units' output distributions using the conditional probability distribution function. BLSTM's and BGRU's output is a matrix of dimensions Width * Height, where the width is equal to the maximum length of the character sequence (it is called time steps). The height is equal to each character's scores on the input image at each time step plus a score for the introduced blank token (-) representing no character or separating the similar characters in the input image. CTC function takes the most probable character at each time step and then merges all the repeated characters in the predicted sequence; after that, the blank tokens are removed. The remaining characters represent the final predicted output. The fundamental steps of decoding operation are depicted in Figure 5.

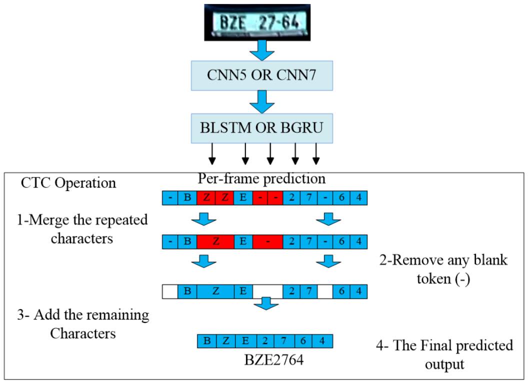


Figure 5. CTC Decoding Operations

F. Experimental Environment

All the experimental works are conducted online utilizing the Google Colab platform with an Intel Xeon Processor of 2.3 GHz and NVIDIA Tesla K80 GPU, including 2496 CUDA cores and GDDR5 VRAM of 12 GB. All our codes are written in Python 3.7, with Keras: 2.1.3, CUDA, CUDNN: 9.0, 7.0. The whole training process of image-based characters sequence has taken from 3.5 to 6 hours, and the training time depends on the complexity of the CRNN architecture. Google colab platform offers 12 hours of continuous training on GPUs for each user every day. Thus, these GPUs are utilized to speed up the models training process, evaluation and testing. A large capacity drive has to be adopted for saving the dataset, pre-processing the images, storing the models during the training, and saving the accuracy and errors metrics. Moreover, Google colab Environment has to be lively connected with the targeted drive during the training, evaluation and testing process.

4. Dataset Description

The dataset used in building and evaluating the models is a real-world license plate dataset that is called ReId Dataset. The source of the dataset is Brno University of Technology, Brno, Czech Republic on Europe, The data was collected from the real urban road, and it contains images-based on characters sequence belong to 8762 vehicles. The data is augmented by adding different level of Gaussian noise to the original image, zooming the image, slightly rotating the text in the image, blurring the image and adding different levels of illumination to the image.The dataset has been split into training, validation and testing data to train, fine-tune and evaluate the models. The training and validation dataset consists of 105,856 image based characters sequence, and the testing dataset consists of 76,288 image based characters sequence. The training, validation and testing datasets are preprocessed by converting the colored images to greyscale images and reshaping them into 64 height and 128 width. After splitting the data, it was divided into batches, and each batch (chunk) consists of 128 images to speed up and smoothen the training process.

5. Results and Discussions

A. *Experimental Settings and Achieved Accuracy*

For the sake of experimental work, ReId Dataset is investigated and adopted because it contains a large number of defective characters. Moreover, ReId dataset is a real-world license plate that perfectly fulfils the needs of the experimental work. The dataset is divided into training data with 182,336 images, validation data with 18,185 images, and testing data with 59,619 images. The training, validation, and testing datasets are pre-processed by converting the coloured images to greyscale images and reshaping them into (64×128).

Connectionist Temporal Classification (CTC) is adopted for training the CRNN models because the CTC loss function can work very well with unaligned (unsegmented) data. Adadelta optimizer is utilized for models' optimization because of its robustness and ability to overcome the continual decay in the learning rate; hence, it doesn't need to initialize the learning rate before the training process starts. Since the rectified linear unit (ReLU) is used to introduce the non-linearity to the models, ”he Initializer” is adopted to bring the variance to the outputs of the ReLU, or generally for neural network activation initialization, hence this initializer is always used with ReLU and Leaky ReLU. Batch size is set to 128, and Val batch size is set to 64, and the maximum characters sequence length was set to 9.

Four CRNN models are developed utilizing ReId dataset for training and testing purposes. During the training process, it is realized that the training losses and validation losses are decreasing together with a slight difference between them but never equalize each other. We came to know that the Convolutional Recurrent Neural Networks (CRNN) Models are not overfitting or underfitting from the behaviour of the training losses and validation losses. Furthermore, Earlystopping Keras technique is adopted to monitor the models' learning and prevent overfitting, and ModelCheckpoint technique is used to save a model after a specific number of successful epochs in the form of a checkpoint file (in the format of hdf5). The saved hdf5 models are adopted to predict on testing data. In this experimental work, the models are set to save a checkpoint file after every five successful epochs in the format of hdf5 named with epoch's number.

From the four trained CRNN models, the two best models are chosen. The first CRNN model (CNN5-BLSTM) trained for about 5 hours, and once it started the training process, the values of the training loss and the validation loss started with 36.1789 and 20.0124, respectively. Both loss values kept decreasing until the training process was terminated at epoch 45, utilizing the earlystopping algorithm. Figure 6 shows the difference between training loss and validation loss of the trained (CNN5-BLSTM) model. This difference is relatively low, so it indicates that the model is not overfitting or underfitting. For the model's performance evaluation, three saved models (hdf5 format models) of CNN5-BLSTM are used to predict on the testing data. The second CRNN model (CNN7-BGRU) trained for about 3.5 hours, and once it started the training process, the values of the training loss and the validation loss started with 22.6871 and 16.0657, respectively, and both loss values kept decreasing until the training process was terminated at epoch 32, using the earlystopping algorithm.

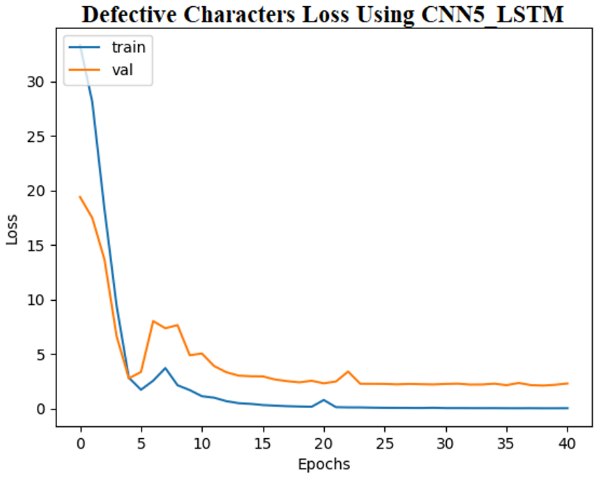


Figure 6. Training and Validation losses in CNN5-LSTM

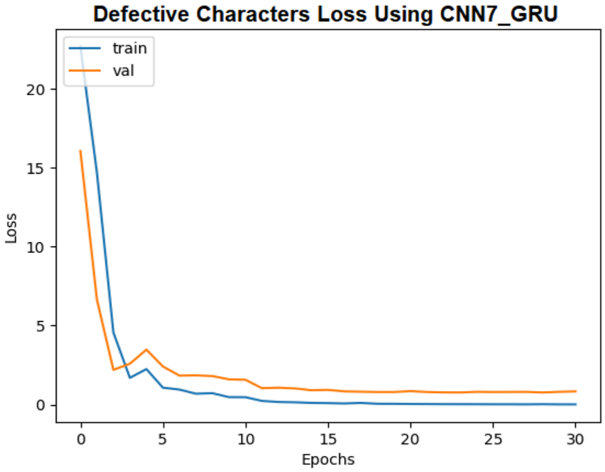


Figure 7. Training and Validation losses in CNN7-GRU

Figure 7 shows the difference between training loss and validation loss of the trained (CNN7-BGRU) model.

TABLE I. Achieved Accuracies by the proposed models

No.	Model's Name	Character's Level Accuracy	Characters' Sequence Accuracy
1	CNN5-BLSTM	98.09	94.34
2	CNN7-BGRU	98.28	95.05

This difference is relatively low, so this difference indicates that the model is not overfitting or underfitting. For model testing, three saved models of CNN7-BGRU are used for the prediction on the testing data. The saved hdf5 models of CNN5-BLSTM and CNN7-BGRU are evaluated on testing data, and their performance was realized to be slightly close, but CNN7-BGRU models outperformed CNN5-BLSTM models in terms of prediction time and achieved accuracy.

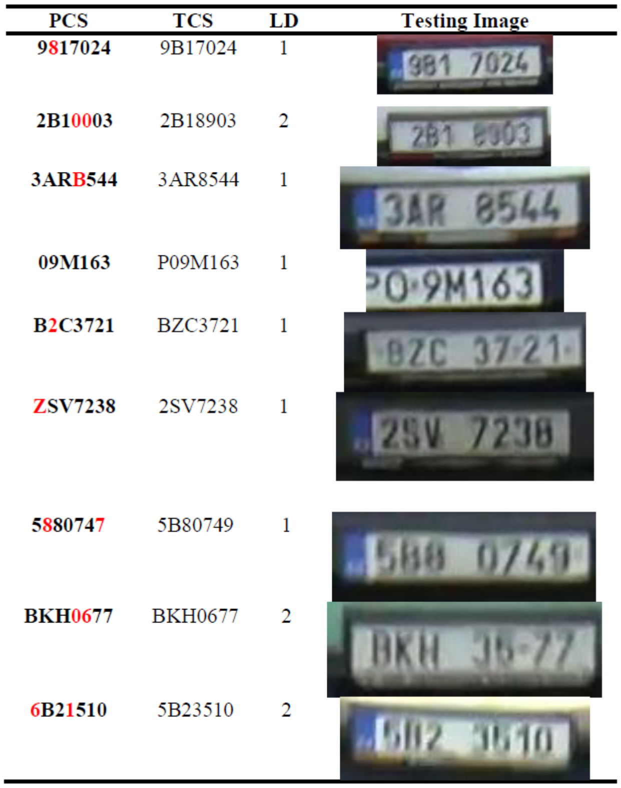
For the evaluation criteria, two kinds of accuracy are calculated: 1) Characters sequence accuracy (or) the accuracy of the perfect match indicates that the prediction result matches the ground truth characters perfectly. 2) Characters' level accuracy demonstrates that at least one of the characters in the sequence is mislabelled. These two kinds of accuracy are broadly and commonly measured by scientists or researchers who worked on building intelligent algorithms for either text recognition or character sequence recognition. Table 1 shows the accuracies of the best performed CRNN models on testing data. It can be realized that CNN7-BGRU performed better than CNN5-BLSTM with overall 95.1% of characters' sequence accuracy and 98.3% of characters' level accuracy.

B. Results and Error Analysis

To find out the limitations and weaknesses of the proposed models, the mislabeled characters sequences are investigated and the Levenshtein Distance (LD) is calculated to show character level error. Levenshtein distance is a method or an algorithm that is used to evaluate how much difference between the predicted characters sequence and their true labels by measuring the minimum number of single-character edits (insertion, deletion or substitution) needed to convert one characters sequence to another. For instance, if the actual characters sequence is "ABC1234" and the predicted character sequence is "ARC1234". In this example, the Levenshtein distance is "1" because the required edit distance to convert the predicted characters sequence "ARC1234" to the actual characters sequence "ABC1234" is by substituting R in the predicted characters sequence by B.

According to the results analysis shown in Figure 8, it is found that most of the mislabeled samples in our testing data have a Levenshtein distance within 1 to 2; therefore, all the predictions made by our models are not far from the actual right labels. Moreover, most of the wrongly labeled plates can be difficult to distinguish by human beings. We have investigated the characters that are incorrectly predicted or mislabeled by our models and found that most of these characters are cut or have significant similarity in their structure, such as (0,8, B, G, D,) , (Z,2), (5, S).

The models performed very well in prediction of the characters sequences on images which are less distorted or at least recognizable by human being and in some times failed to recognize the characters which are fully or partially covered by shadow or the characters that are partially cut by the edge of the image. However, the overall performance of the models is quite well compared to other methods in literature.



PCS	TCS	LD	Testing Image
9817024	9B17024	1	
2B10003	2B18903	2	
3ARB544	3AR8544	1	
09M163	P09M163	1	
B2C3721	BZC3721	1	
ZSV7238	2SV7238	1	
5880747	5B80749	1	
BKH0677	BKH0677	2	
6B21510	5B23510	2	

Figure 8. Samples of Mislabeled Characters Sequences

C. Results Comparison

Finally, the proposed models' performance is compared with other works' performance in the literature using the ReId dataset as the reference data. It is found that the proposed models outperformed the other models in prediction on ReId dataset. Figure 9 shows the recognition performance of four works from the literature on ReId Dataset, and their results are compared with the results of the proposed models in this paper. The comparison includes two kinds of accuracy: 1) character's level accuracy and 2) characters' sequence accuracy or the accuracy of the perfect match. The methods that are evaluated on ReId dataset are introduced as follows.

1) OpenALPR. It is an open-source library that was written in C++ for Automatic LPR. This library is totally based on OpenCV, Computer Vision and Tesseract OCR. Moreover, this LP system was adopted in EU license plates, for instance EasyPR. Hence this method was evaluated on ReId Dataset.

2) SOTA. This method was recently proposed in [29]. This method involves four essential processing phases: the first phase is image acquisition phase that is used to acquire the image from the source, the second stage is license plate extraction phase that is utilized to crop the image, and the segmentation and recognition phases are used to segment and then recognize the input image. In the experimental work, ReId Dataset was adopted to evaluate the recognition system of this method.

3) Holistic. Is a segmentation-free method introduced by J. Špaňhel, J. Sochor, R. Juránek, A. Herout, L. Maršík and P. Zemčík in their paper entitled "Holistic recognition of low-quality license plates by CNN". This method is considered to be an end-to-end method that is used to recognize license plate images with low quality, and it was trained and evaluated on ReId Dataset. This method is robust and has shown significant performance in recognizing the characters sequence in low quality license plates.

For more illustration, four existing methods based on Convolutional Neural Networks (CNN) are evaluated on ReId Dataset. These methods are as follows: ResNet50, ResNet101, DenseNet169, DenseNet201, Inception-v3, and CliqueNet-S3. Note that, these methods are directly trained on cropped characters' sequence images and their evaluation accuracies on ReId Dataset are illustrated in Figure 10.

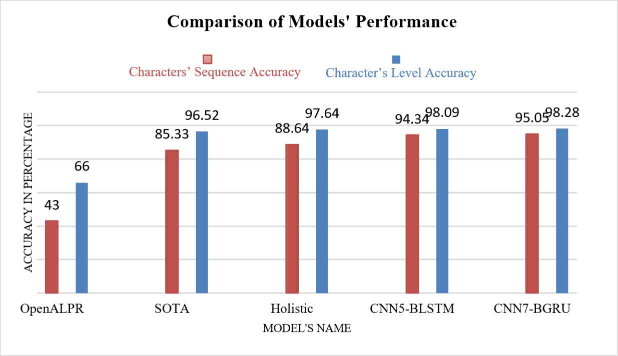


Figure 9. Results Comparison with other Models in Literature

6. Conclusion

This paper presents two deep learning recognition models based on Convolutional Recurrent Neural Networks (CRNN). The two proposed models are lightweight because the number of trainable parameters is very less compared to other models. Both models are experimentally trained, evaluated and tested on ReId Dataset and it is realized that, the performance accuracy of the CNN7-BGRU Model is slightly higher than the performance accuracy of the CNN5-BLSTM Model. Furthermore, the proposed CRNN models outperform other models in the literature using ReId dataset as a reference.

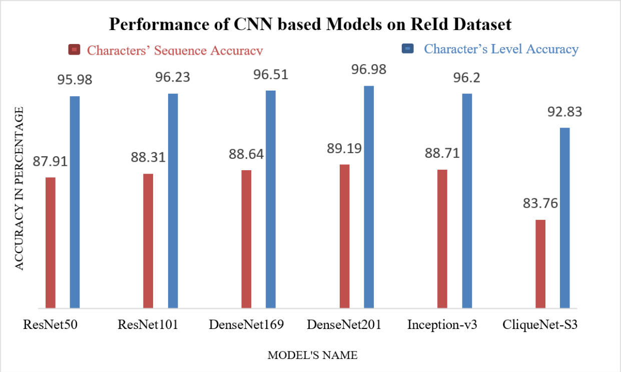


Figure 10. Performance of CNN Based Models on ReId Dataset

Building intelligent systems to recognize the defective characters on images became a very crucial and interesting research problem because this problem has not yet been totally solved as there are many challenges in recognizing the defective characters perfectly. Digital image processing techniques and segmentation-based techniques Failed to achieve satisfactory accuracy in recognizing the defective characters on images. Thus, our research work focuses on segmentation-free deep learning algorithms to address this problem. This work can be improved in the future by experimentally adopting capsule Network for feature extraction, transformers and attention-based models for character recognition.

References

[1] S. Long, X. He, and C. Yao, "Scene text detection and recognition: The deep learning era," *International Journal of Computer Vision*, vol. 129, no. 1, pp. 161–184, 2021.

[2] R. Ptak, B. Żygadło, and O. Unold, "Projection-based text line segmentation with a variable threshold," *International Journal of Applied Mathematics and Computer Science*, vol. 27, no. 1, 2017.

[3] A. Zoizou, A. Zarghili, and I. Chaker, "A new hybrid method for arabic multi-font text segmentation, and a reference corpus construction," *Journal of King Saud University-Computer and Information Sciences*, vol. 32, no. 5, pp. 576–582, 2020.

[4] M. Y. Arafat, A. S. M. Khairuddin, and R. Paramesran, "Connected component analysis integrated edge based technique for automatic vehicular license plate recognition framework¡? show [aq="" id=" q1]"?" *IET Intelligent Transport Systems*, vol. 14, no. 7, pp. 712–723, 2020.

[5] C. Gou, K. Wang, Y. Yao, and Z. Li, "Vehicle license plate recognition based on extremal regions and restricted boltzmann machines,"

IEEE Transactions on Intelligent Transportation Systems, vol. 17, no. 4, pp. 1096–1107, 2015.

[6] S. Liu, H. Xie, C. Zhou, and Z. Mao, "Uyghur language text detection in complex background images using enhanced msers," in *International Conference on Multimedia Modeling*. Springer, 2017, pp. 490–500.

[7] V. Vučković and B. Arizanović, "Efficient character segmentation approach for machine-typed documents," *Expert Systems with Applications*, vol. 80, pp. 210–231, 2017.

[8] M. Delakis and C. Garcia, "text detection with convolutional neural networks." in *VISAPP (2)*, 2008, pp. 290–294.

[9] E. Al-wajih and R. Ghazali, "Improving the accuracy for offline arabic digit recognition using sliding window approach," *Iranian Journal of Science and Technology, Transactions of Electrical Engineering*, vol. 44, no. 4, pp. 1633–1644, 2020.

[10] Z. Zhong, L. Jin, and S. Huang, "Deeptext: A new approach for text proposal generation and text detection in natural images," in *2017 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2017, pp. 1208–1212.

[11] S. Sagar, S. Dixit, and B. Mahesh, "Offline cursive handwritten word using hidden markov model technique," in *Smart Intelligent Computing and Applications*. Springer, 2020, pp. 525–535.

[12] K. K. Kim, J. H. Kim, Y. K. Chung, and C. Y. Suen, "Legal amount recognition based on the segmentation hypotheses for bank check processing," in *Proceedings of sixth international conference on document analysis and recognition*. IEEE, 2001, pp. 964–967.

[13] P. Inkeaw, J. Bootkrajang, P. Charoenkwan, S. Marukatat, S.-Y. Ho, and J. Chaijaruwanich, "Recognition-based character segmentation for multi-level writing style," *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 21, no. 1, pp. 21–39, 2018.

[14] A. Agarwal and S. Goswami, "An efficient algorithm for automatic car plate detection & recognition," in *2016 Second International Conference on Computational Intelligence & Communication Technology (CICT)*. IEEE, 2016, pp. 644–648.

[15] A. Vaishnav and M. Mandot, "Template matching for automatic number plate recognition system with optical character recognition," in *Information and Communication Technology for Sustainable Development*. Springer, 2020, pp. 683–694.

[16] M. Mohamad, H. Hassan, D. Nasien, and H. Haron, "A review on feature extraction and feature selection for handwritten character recognition," *International Journal of Advanced Computer Science and Applications*, vol. 6, no. 2, 2015.

[17] D. Yao, W. Zhu, Y. Chen, and L. Zhang, "Chinese license plate character recognition based on convolution neural network," in *2017 Chinese Automation Congress (CAC)*, 2017, pp. 1547–1552.

[18] Y. Liu and H. Huang, "Car plate character recognition using a convolutional neural network with shared hidden layers," in *2015 Chinese Automation Congress (CAC)*. IEEE, 2015, pp. 638–643.

[19] T. Kumar, S. Gupta, and D. S. Kushwaha, "An efficient approach for automatic number plate recognition for low resolution images," in *Proceedings of the Fifth International Conference on Network, Communication and Computing*, 2016, pp. 53–57.

[20] J. Špaňhel, J. Sochor, R. Juránek, A. Herout, L. Maršík, and P. Zemčík, "Holistic recognition of low quality license plates by cnn using track annotated data," in *2017 14th IEEE International Conference on Advanced Video and Signal Based Surveillance (AVSS)*. IEEE, 2017, pp. 1–6.

[21] M. A. Córdova, L. G. Decker, J. L. Flores-Campana, A. A. dos Santos, J. S. Conceição, A. Pinto, H. Pedrini, and R. d. S. Torres, "Pelee-text: A tiny convolutional neural network for multi-oriented scene text detection," in *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*. IEEE, 2019, pp. 400–405.

[22] W. Liu, C. Chen, and K.-Y. Wong, "Char-net: A character-aware neural network for distorted scene text recognition," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 32, no. 1, 2018.

[23] B. Shi, X. Bai, and C. Yao, "An end-to-end trainable neural network for image-based sequence recognition and its application to scene text recognition," *IEEE transactions on pattern analysis and machine intelligence*, vol. 39, no. 11, pp. 2298–2304, 2016.

[24] Z. Wan, M. He, H. Chen, X. Bai, and C. Yao, "Textscanner: Reading characters in order for robust scene text recognition," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 07, 2020, pp. 12 120–12 127.

[25] T. Wang, Y. Zhu, L. Jin, C. Luo, X. Chen, Y. Wu, Q. Wang, and M. Cai, "Decoupled attention network for text recognition," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 07, 2020, pp. 12 216–12 224.

[26] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[27] L. Yang, P. Wang, H. Li, Z. Li, and Y. Zhang, "A holistic representation guided attention network for scene text recognition," *Neurocomputing*, vol. 414, pp. 67–75, 2020.

[28] F. Sheng, Z. Chen, and B. Xu, "Nrtr: A no-recurrence sequence-to-sequence model for scene text recognition," in *2019 International Conference on Document Analysis and Recognition (ICDAR)*. IEEE, 2019, pp. 781–786.

[29] K. Indira, K. Mohan, and T. Nikhilashwary, "Automatic license plate recognition," in *Recent Trends in Signal and Image Processing*. Springer, 2019, pp. 67–77.

Hashem Al-Nabhi is currently pursuing his Ph.D. in information and Communication Engineering at Northwestern Polytechnical University (NWPU), China. He did his master degree in Information and Communication Engineering at Northwestern Polytechnical University (NWPU). He did his bachelor's degree in Electronics and Communication Engineering at Jawaharlal Nehru Technological University, Anantapur (JNTUA), India. His research interests include image processing, machine learning, deep learning and computer vision.

Dr. K.Lokesh Krishna is currently working as Professor in department of Electronics and Communication Engineering at S.V.College of Engineering, Tirupati with an overall teaching experience of 21 years. He received his Ph.D degree in VLSI Design from SVUCE, S.V.University, Tirupati. He has published more than 55 technical papers in different reputed International journals and International conferences. He is a Fellow member of IE and Life Member of ISTE. His areas of research interest include analog and digital VLSI design and IoT system design..

Ahmed Abdullah Ali Shareef Has received his B.SC in Computer Science in 2009 from Sana'a University, Yemen, Master of Technology in Embedded Systems in 20016 from JNTU, Anantapur, India He is currently a Ph.D candidate at Dept. of Computer Science and IT, Dr. Babasaheb Ambedkar Marathwada University, Aurangabad, Maharashtra, India. His research interests in artificial intelligence, computer vision, and blockchain..