**Title: ConvRecog: Convolutional Recurrent Neural Network for Handwritten Text Recognition**

**Abstract:**

Recognizing handwriting in different scenarios can be complicated. This paper aims to introduce an ideal approach to handwriting recognition by combining Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN).It aims to build a bridge to fill the gaps between human written documents and machine readable formats. While CNN excels at extracting spatial features from images, they face difficulties to capture the sequential nature of text due to the lack of memory. That's where RNN comes in. RNN has a memory but it lacks the extraction of spatial features from images. So combining both will help mitigate the limitations of both the models. This work aims to contribute significantly to the field of Handwriting recognition through the introduction of a powerful and efficient architecture.

**Introduction:**

In the landscape of computer vision and pattern recognition, the intersection of Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) has given rise to innovative architectures designed for complex tasks. One such paradigm, the Convolutional Recurrent Neural Network (CRNN), presents a holistic approach to the intricate challenge of Handwritten Text Recognition (HTR). In this paper, we delve into the design, implementation, and exploration of a CRNN model tailored for the specific demands of HTR.

Handwritten Text Recognition stands as a quintessential problem in document analysis, with applications ranging from digitizing historical manuscripts to automating data entry processes. The dynamic nature of handwritten text, with variations in style, size, and orientation, necessitates a sophisticated model capable of deciphering these nuances. The CRNN architecture emerges as a promising solution, seamlessly marrying the spatial awareness of CNNs with the sequential understanding of RNNs.

Our investigation begins with a detailed examination of the image feature extraction process, employing a CNN to distill pertinent information from handwritten text images. This initial phase sets the foundation for subsequent sequence learning, where the unique characteristics of the handwritten script are comprehensively captured using Bidirectional Long Short-Term Memory (Bi-LSTM) networks. The bidirectional nature of Bi-LSTMs empowers the model to discern temporal dependencies both forward and backward in time, a crucial attribute for understanding the nuanced order of strokes and characters.

A pivotal aspect of our methodology lies in the 'labeling' block, where Connectionist Temporal Classification (CTC) is employed to convert the sequence of learned features into the final output. This step ensures that our CRNN model excels in recognizing variable-length sequences of characters, a common challenge in HTR tasks.

As we present our experimental configurations, including the architecture of the CNN block and the specifics of the Bi-LSTM layers, we aim to highlight the efficacy of the CRNN model in handling the intricacies inherent in handwritten text recognition. Additionally, we discuss future directions and potential enhancements, envisioning a path forward for CRNNs in addressing evolving challenges and expanding applications within the realm of document analysis and beyond. Through this exploration, we contribute to the growing body of knowledge in leveraging deep learning architectures for the nuanced task of Handwritten Text Recognition.

**Related Works:**

**Pape Name:**  Improving CNN-RNN Hybrid Networks for Handwriting Recognition  
  
The research aims to enhance the accuracy of recognising handwritten text in scanned offline document pictures by employing deep learning models. The authors suggest a revised CNN-RNN hybrid structure that emphasizes efficient training. The key contributions include employing synthetic data for pretraining, applying picture normalization to correct slant, and utilizing domain-specific data transformation to acquire crucial invariances. The work does an ablation analysis to assess the influence of specific modules and attains cutting-edge outcomes on well-known datasets like IAM, RIMES, and GW. The suggested model achieves superior performance compared to current approaches by integrating advanced techniques such as spatial transformer networks, dropout, residual learning, and data augmentation. This study emphasizes the significance of these strategies in improving the accuracy of recognising handwritten text in different environments and data collections.

**Paper Name:** Cursive Text Recognition in Natural Scene Images Using Deep Convolutional

Recurrent Neural Network

The study presents a method for recognising cursive writing in natural scene photos without the need for segmentation. The specific focus is on recognising Urdu language. The approach employs a CRNN, which is a kind of neural network that combines convolutional and recurrent layers. It incorporates shortcut connections to improve performance and does not require pre-segmentation of characters. The model integrates multiple deep convolutional neural network architectures (VGG-16, VGG-19, ResNet-18, ResNet-34) to extract features and evaluate their respective performance. The tests conducted on a recently developed extensive Urdu dataset showcase the exceptional performance of the suggested deep Convolutional Recurrent Neural Network (CRNN) model. The contributions encompass the creation of a pioneering VGG-16 structure with shortcut connections, as well as the establishment of a publicly accessible dataset for recognising Urdu text in real-life environments. The study examines obstacles, suggests efficient procedures, and presents experimental findings that validate the effectiveness of the offered methodology.

**Paper Name :** Text Recognition Model Based on Multi-Scale Fusion CRNN

The research presents a new method called the Multiple Scales Fusion Convolutional Recurrent Neural Network (MSF-CRNN) to enhance the precision of scene text recognition. Scene text recognition refers to the process of understanding text sequences seen in real-world settings. The suggested approach is designed to address the challenges associated with recognising longer text. The MSF-CRNN integrates a convolutional layer with a multi-scale output CNN, a feature fusion layer, a recurrent layer, and a transcription layer. The use of multi-scale fusion enhances the model's ability to accurately identify bigger text by extracting a wider range of comprehensive information. The experimental findings on benchmark datasets clearly indicate that the MSF-CRNN model outperforms the CRNN model in terms of accuracy. The model undergoes evaluation on many datasets, revealing enhanced performance, notably in the identification of bigger text. The research also evaluates the performance of the MSF-CRNN model in comparison to other current models, emphasizing its superior accuracy on various datasets. The suggested approach enhances scene text recognition by efficiently integrating multi-scale information for feature extraction. Subsequent studies might investigate various fusion techniques and take into account the reduction of model complexity for implementation. The research is funded by many funds, and the authors have no conflicts of interest to disclose.

**Paper Name :** Complexity Reduction over Bi-RNN-Based Nonlinearity

Mitigation in Dual-Pol Fiber-Optic Communications via a

CRNN-Based Approach

This research presents a hybrid neural network structure designed to reduce the effects of nonlinear channel impairments in long-distance fiber-optic communications. The main objective is to tackle the significant computational complexity linked to bidirectional recurrent neural networks (bi-RNNs), such as bidirectional long short-term memory (bi-LSTM) and bidirectional gated recurrent unit. The suggested framework integrates a convolutional neural network (CNN) encoder with a unidirectional many-to-one vanilla recurrent neural network (RNN). The Convolutional Neural Network (CNN) is designed to capture short-term relationships, whereas the Recurrent Neural Network (RNN) is specialized in handling long-range connections with a lower level of complexity. The model is evaluated in two receiver setups, one for mitigating nonlinearity after linear equalization and the other for compensating for both nonlinearity and polarization mode dispersion after chromatic dispersion correction

**Methodology:**

This study has been carried out for recognizing handwriting . In the first stage, After Loading the dataset , we cleaned it according to our model needs. First we checked for null values and removed them.Also, there are some images in our data with the label 'UNREADABLE'. We detected them and removed them from the dataset. There are some labels which are in lowercase. To maintain uniformity in the labels, I convert all the labels to uppercase. In the second phase , we pre processed the data. Then we prepared the labels for the CTC loss. Then We build and run a model using three blocks of Convolutional Neural Network (CNN) combined with Recurrent Neural Network (RNN).

**Dataset:**

The dataset that was used in this proposed method was the Handwriting Recognition dataset collected from Kaggle. This dataset consists of more than four hundred thousand handwritten names collected through charity projects. Character Recognition utilizes image processing technologies to convert characters on scanned documents into digital forms. It typically performs well in machine-printed fonts. However, it still poses difficult challenges for machines to recognize handwritten characters, because of the huge variation in individual writing styles. There are 206,799 first names and 207,024 surnames in total. The data was divided into a training set (331,059), testing set (41,382), and validation set (41,382) respectively.

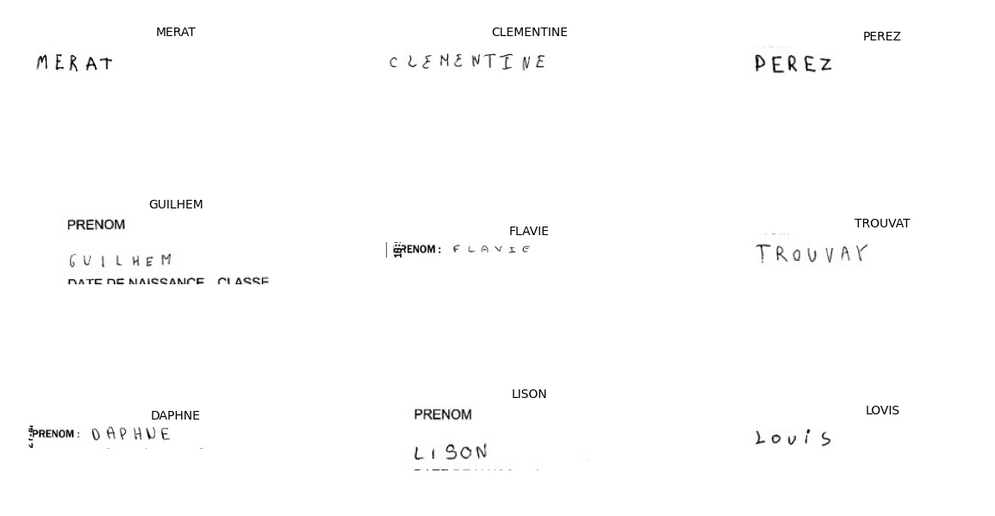
**Pre process:**

For pre-processing the data, the images are loaded and reshaped to width 256 and height 64. Converted colored images to grayscale if the information conveyed by color is not essential for the handwriting recognition task. Grayscale images are computationally more efficient and often sufficient for character recognition. Pad or crop images to a consistent size to create a fixed-size input for the CRNN model.Padding can be applied to smaller images, while cropping may be necessary for larger images. The width and height are cropped if they are greater than 256 and 64 respectively. If they are smaller, then the image is padded with white pixels. This step ensures that all input images have the same dimensions, facilitating batch processing during training. Finally the image is rotated clockwise to bring the image shape to (x, y). The image is then normalized to range [0, 1] to enhance convergence during training. Normalization helps in mitigating the impact of varying pixel intensity across different images.

Computed the maximum length from words and padded every output label to make it of the same size as the maximum length. This is done to make it compatible with the output shape of our RNN architecture.

Our model consists of three parts:

1. The convolutional neural network to extract features from the image.
2. Recurrent neural network to predict sequential output per time-step.
3. CTC loss function which is a transcription layer used to predict output for each time step.



Here, we are using the CTC loss function. CTC loss is very helpful in text recognition problems. It helps us to prevent annotating each time step and helps us to get rid of the problem where a single character can span multiple time steps which needs further processing if we do not use CTC. A CTC loss function requires four arguments to compute the loss, predicted outputs, ground truth labels, input sequence length to LSTM and ground truth label length. To get this we need to create a custom loss function and then pass it to the model. To make it compatible with our model, we will create a model which takes these four inputs and outputs the loss.  
  
We splitted the Pre processed images into three groups. Train , Test and Validation.

**CNN :**

One of the most popular deep neural networks is the Convolutional Neural Network (CNN). It takes this name from mathematical linear operation between matrices called convolution [3] . CNN has multiple layers; including convolutional layer, non-linearity layer, pooling layer and fully-connected layer. The convolutional and fully-connected layers have parameters but pooling and non-linearity layers don't have parameters. CNN has an excellent performance in machine learning problems. CNNs primarily focus on the basis that the input will be composed of images. This focuses the architecture to be set up in a way to best suit the need for dealing with the specific type of data.[4] .

One of the key differences is that the neurons that the layers within the CNN

are composed of neurons organized into three dimensions, the spatial dimensionality of the input (height and the width) and the depth. The depth does not refer to the total number of layers within the ANN, but the third dimension of a

activation volume. Unlike standard ANNS, the neurons within any given layer

will only connect to a small region of the layer preceding it.

In practice this would mean that for the example given earlier, the input ’vol-

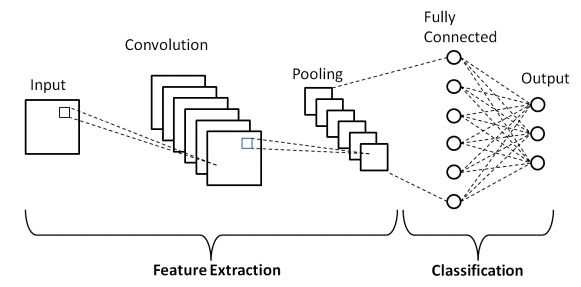
ume’ will have a dimensionality of 64 × 64 × 3 (height, width and depth), lead-

ing to a final output layer comprised of a dimensionality of 1 × 1 × n (where

n represents the possible number of classes) as we would have condensed the

full input dimensionality into a smaller volume of class scores filed across the

depth dimension.



**RNN:**

Recurrent Neural Networks (RNNs) are a class of neural networks designed to handle sequential data and capture temporal dependencies. Unlike traditional feedforward neural networks, RNNs have connections that form a directed cycle, allowing them to maintain a memory of past inputs [5]. This makes RNNs well-suited for tasks involving sequences, such as time series prediction, natural language processing, and speech recognition.The fundamental building block of an RNN is the recurrent unit, which takes an input and considers the information from the previous step. The basic structure consists of three main components:

1. **Input Layer (X):**

Represents the current input at the current time step.

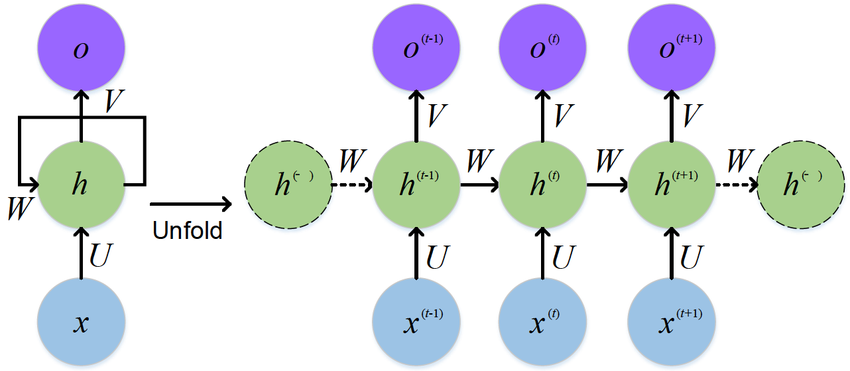
1. **Hidden Layer (H):**

Captures information from the previous time step and the current input. It serves as the memory of the network, allowing it to retain information over time.

1. **Output Layer (Y):**

Produces the output for the current time step based on the current input and the information stored in the hidden layer. [6]

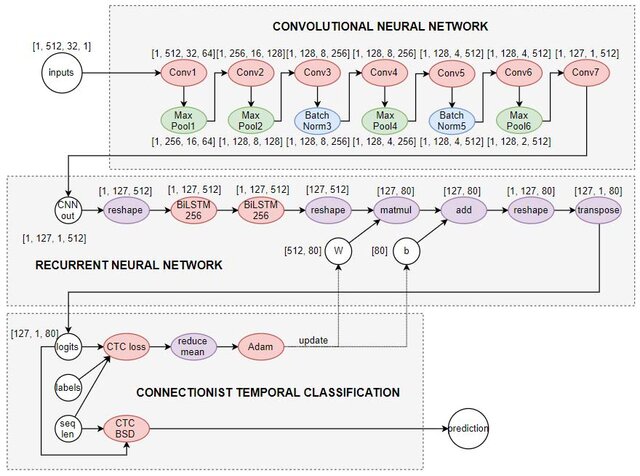
The connection between the hidden layer's output and its own input forms a loop, allowing the network to maintain a memory of previous inputs and incorporate them into the current computations. However, RNN may struggle with capturing long-term dependencies due to the vanishing or exploding gradient problem.This led to the development of more sophisticated architectures, such as Long Short-Term Memory (LSTM) networks and Gated Recurrent Unit (GRU) networks, which are designed to address these issues and better capture long-range dependencies.



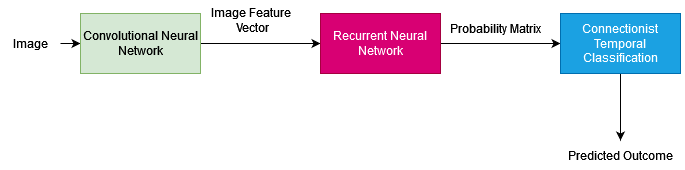
**CRNN:**

Convolutional Recurrent Neural Networks (CRNNs) represent a hybrid architecture that combines the strengths of Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs). This fusion is particularly effective for tasks involving sequential data embedded in spatial structures, making CRNNs well-suited for applications like image captioning, scene understanding, and, notably, handwritten text recognition.

[7]



The designed Convolutional Recurrent Neural Network (CRNN) architecture is a sequence of three blocks.



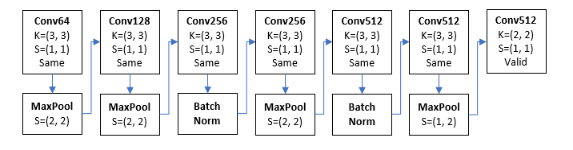
CRNN essentially consists of a sequential connection of three neural network blocks: a ‘convolutional’ block for image feature extraction, a ‘recurrent’ block for sequence learning, and finally a ‘labeling’ block with Connectionist Temporal Classification. [8]

1. **Image Feature Extraction:**

In computer vision tasks, handling images of varying sizes and color channels can be computationally intensive. To address this, a Convolutional Neural Network (CNN) is employed to transform images into a computationally manageable form while preserving crucial predictive features.

The CNN block in this design consists of seven convolutional layers. The architecture includes max-pooling layers for downsampling after the first, second, fourth, and sixth convolutional layers. Additionally, batch normalization is applied after the third and fifth layers to enhance training stability.

The CNN layers are activated using Rectified Linear Unit (ReLU) functions. For instance, the first layer, Conv64 with a 3x3 filter size, 64 filters, and 'same' padding, is followed by a max-pooling layer with a 2x2 kernel size. Similarly, subsequent layers follow a similar pattern. Max-pooling reduces the dimensionality of the data by replacing each sub-region with its maximum value.

[7]

1. **Sequence Learning: Bidirectional LSTM (Bi-LSTM) Block**

After the seventh convolutional layer, the immediate output is a 1-dimensional feature sequence, which represents the original 2-dimensional image matrix. This feature sequence becomes the input for the Recurrent Neural Network (RNN) block.

In this paper, the chosen RNN variation is the Bidirectional Long Short-Term Memory (Bi-LSTM). A Bi-LSTM consists of two independent LSTMs (Long Short-Term Memory networks) arranged in parallel. One LSTM processes inputs in the normal time order, while the other processes inputs in reverse time order. The outputs of these two networks are then concatenated, creating a comprehensive representation of the input sequence at each timestep. This bidirectional structure enables the network to capture information both forward and backward in time, facilitating a more contextual understanding of the input data.

The RNN block in this architecture comprises two consecutive Bi-LSTMs, each containing 256 memory cells. This configuration allows the model to capture intricate temporal dependencies and contextual information within the sequential data. The bidirectional nature of the Bi-LSTM enhances the network's ability to make predictions by considering both past and future context, which can be particularly beneficial for tasks requiring an understanding of the temporal relationships in the data.

1. **Labeling Block and Full Network:**

After the Sequence Learning phase with the Bidirectional LSTM (Bi-LSTM) block, the feature sequence is passed to the 'labeling' block, where the final predictions are made using the Connectionist Temporal Classification (CTC) approach. The labeling block transforms the sequential features into the output sequence, making it suitable for tasks like handwritten text recognition.

**Labeling Block:** The labeling block employs the Connectionist Temporal Classification (CTC) method, a powerful technique for sequence labeling tasks. CTC enables the model to handle sequences of varying lengths and align input sequences with their corresponding labels without the need for explicit alignment information during training. This makes it particularly effective for tasks like recognizing variable-length sequences of characters in handwritten text.

**Full Network:** The full network comprises the 'image feature extraction,' 'sequence learning,' and 'labeling' blocks interconnected in a sequential manner. It represents an end-to-end architecture where the model learns to extract hierarchical features from images, capture temporal dependencies, and produce sequence predictions through the CTC-based labeling block.

**Optimization:** The training of the entire CRNN model involves optimizing its parameters to minimize a chosen loss function. Commonly used optimization algorithms include stochastic gradient descent (SGD), Adam, or RMSprop. The optimization process aims to adjust the weights and biases of the network to improve its ability to make accurate predictions on the training data.

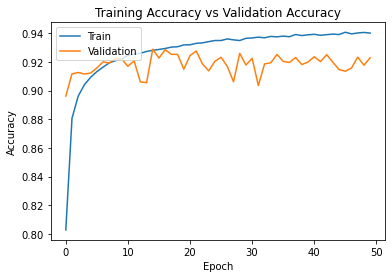
During training, backpropagation through time (BPTT) is typically applied to update the model's parameters. BPTT unfolds the network over time, treating it as an extended network, and updates the parameters based on the gradient of the loss function. Regularization techniques, such as dropout, may be employed to prevent overfitting and enhance generalization.

The choice of hyperparameters, including learning rate, batch size, and the number of training epochs, plays a crucial role in the optimization process. These hyperparameters are tuned to find the optimal balance between model training speed and performance.

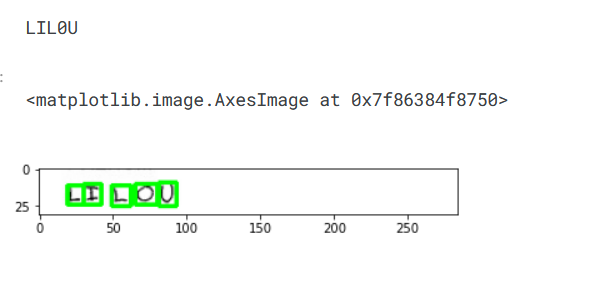
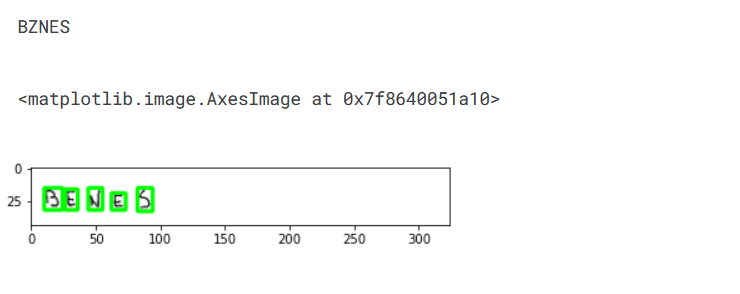
**Result :**

This section provides results from the training stage and examples of predictions using the model after 10 epochs, equivalent to 60,989 gradient descent steps. The results include the change in training cost and edit distance between

ground truth and predicted labels over the whole training period. After running the experiment, we get the training accuracy of the system is 94.00 % and Validation accuracy 92.28%.



After that we did post processing to check how the model is performing. Here are the results we got .



**Limitations:**

1. The recognition part is dependent on the contour detection code, so if the opencv library is not able to find the character contour, then this method will fail.
2. There could be a lot of variation in a single handwritten letter in terms of writing style, therefore a lot more examples are needed for training this model.
3. This model will not work for connected texts like a cursive handwritten word.

**Future Works:**

Explore the integration of human-in-the-loop systems for CRNN, where user feedback is utilized to iteratively improve model performance. This collaborative approach can be valuable in refining the model's predictions over time. Optimize the CRNN model for real-time handwritten text recognition, making it suitable for applications where prompt processing is crucial, such as document scanning or interactive devices.Develop mechanisms for incremental learning, allowing the CRNN model to adapt to new handwriting styles or evolving datasets over time. This can be essential for applications with continuously changing data distributions.

**Conclusion:**

The convolutional recurrent neural network was adopted and applied on a handwritten text recognition. Training time is constrained by configuring the depth and complexity of the neural network architecture so that each epoch does not take too long and thus could be executed on limited hardware resources.The current system is still sensitive to noise, where the input image needs to be cleaned and possess high contrast between the text and background.