Deep Learning Based OCR for Mathematical Expressions

By

Asad Rehman

Fall 2019-MS(CS)-09-0000319836

Supervisor

Dr.Ahmad Salman

Department of Computer Sciences

A thesis submitted in partial fulfillment of the requirements for the degree of Masters in Computer and Communication Security (MS CCS)

In

School of Electrical Engineering and Computer Science,

National University of Sciences and Technology (NUST),

Islamabad, Pakistan.

(February, 2023)

Online Printing Date & Time: Thursday, 15 June 2023 21:08:27

THESIS ACCEPTANCE CERTIFICATE

Certified that final copy of MS/MPhil thesis entitled "Deep Learning Based OCR for Mathematical Expressions" written by ASAD REHMAN, (Registration No 00000319836), of SEECS has been vetted by the undersigned, found complete in all respects as per NUST Statutes/Regulations, is free of plagiarism, errors and mistakes and is accepted as partial fulfillment for award of MS/M Phil degree. It is further certified that necessary amendments as pointed out by GEC members of the scholar have also been incorporated in the said thesis.

Signature: [illegible]

Name of Advisor: Dr. Ahmad Salman

Date: 09-Jun-2023

HoD/Associate Dean: ______________________

Date: ______________________

Signature (Dean/Principal): ______________________

Date: ______________________

Publish Date & Time: Friday, 09 June 2023 , 16:07:47

Online Printing Date & Time: Thursday, 15 June 2023 13:27:54

Approval

It is certified that the contents and form of the thesis entitled "Deep Learning Based OCR for Mathematical Expressions" submitted by ASAD REHMAN have been found satisfactory for the requirement of the degree

Advisor : Dr. Ahmad Salman

Signature: [illegible]

Date: 09-Jun-2023

Committee Member 1:Dr. Salman Abdul Ghafoor

Signature: [illegible]

09-Jun-2023

Committee Member 2:Dr. Khawar Khurshid

Signature: [illegible]

Date: 09-Jun-2023

Signature: ______________________

Date: __________________________

Publish Date & Time: Friday, 09 June 2023 , 16:07:47

Dedication

I would love to dedicate my thesis to my beloved parents, teachers, my family, and my friends who guided me throughout my degree.

Certificate of Originality

I hereby declare that this submission is my own work and to the best of my knowledge it contains no materials previously published or written by another person, nor material which to a substantial extent has been accepted for the award of any degree or diploma at NUST SEECS or at any other educational institute, except where due acknowledgment has been made in the thesis. Any contribution made to the research by others, with whom I have worked at NUST SEECS or elsewhere, is explicitly acknowledged in the thesis.

I also declare that the intellectual content of this thesis is the product of my own work, except for the assistance from others in the project's design and conception or in style, presentation, and linguistics which has been acknowledged.

Author Name: **Asad Rehman**

Signature: [illegible]

Acknowledgment

In the name of **Allah**, The Most Beneficent, The Most Merciful. All praises to **Allah** Almighty for providing me the courage to complete my thesis.
I owe my deepest gratitude to my supervisors **Dr. Ahmad Salman**. Without their continuous concern about this work, knowledgeable advice, support and encouragement this work would have been a far cry. I would also like to thank my committee member **Dr. Salman Abdul Ghafoor** and **Dr. Khawar Khurshid** for their valuable feedback, advice and comments throughout this research. I would like to thank **Muhammad Safeer-Ul-Azeem Abbasi** who helped me during the thesis and gave valuable feedback at multiple stages.
I am incredibly thankful to my **parents** for their love and continuous support throughout this journey.

Contents

List of Figures

List of Tables

Abstract

Mathematical expression recognition with the help of a deep neural network is proposed in this work. A latex sequence is generated from this model of a given mathematical expression image. An attention aggregation-based bi-directional mutual learning network is used along with high-level and low-level feature maps to make it a multiscale feature with a multiscale attention aggregation model. A multiscale feature map is used to restore the fine-grained details that use to drop by pooling operations of the Dense encoder. The model was trained and tested on CROHME dataset with different numbers of classes.

Chapter 1

Introduction

Artificial neural networks digitally impersonate the working of human neural networks. They are very helpful to solve abstract problems that require a high degree of accuracy and complex computation. Artificial neural networks can be very helpful in solving problems like image classification, image segmentation, image recognition, audio processing, computer-aided diagnostics, optical character recognition, etc. Since the artificial neural network can recognize and classify an object from an image, they have been used in optical character recognition. OCR is required because the document that has been stored in text form needs less space than pictures. Most old documents are written by hand and OCR can convert these documents easily to text. But in some scientific documents, mathematical expressions have also been handwritten. Like other text mathematical expressions also need to be digitized. Digitizing Mathematical expressions are more difficult than regular text. Normal text like English, Urdu, and character from other languages are easy to recognize by an artificial neural network. Because of their simple writing structure. But recognizing mathematical expressions is a difficult task. The reason is, in mathematical expressions along with the main script, the expression has some text written above the main script and some text written below the main script. Also, because of some special symbols, mathematical expressions cannot be recognized correctly by ordinary OCR.

1.1 Terminologies

Some standard terminologies are re-stated for ease of further reading. These terminologies include an encoder, decoder, attention model, MathML, CROHME, expiration rate, and word error rate

1.1.1 Encoder

In OCR for mathematical expression recognition an input image contain complex two-dimensional expression and their length can be very long. To extract information from this image a model called an encoder is used. This encoder comprises a convolutional neural network like VGG-Net or DenseNet. The output of that encoder is a three-dimensional feature map.

1.1.2 Decoder

It is a model with soft attention mechanism whose job is to convert features extracted by the encoder into a sequence of tokens extracted from the vocabulary. RNNs are widely used to work as a decoder. The reason for using RNNs is that they can maintain a history of the precedent prognostications and can traverse from the commencement to the cessation of sequence at arbitrary length.

1.1.3 Attention Model

In OCR, the position of the attention model is in between the encoder and decoder. The attention model helps the decoder to understand the sequence alignment. It focuses on the part of sequences that are not translated by the decoder by giving them high probabilities.

1.1.4 CROHME

Competition on Recognition of Online Handwritten Mathematical Expressions has developed a dataset that is used widely for the training of mathematical expression detection. The dataset contains, hand-written matrices, hand-written mathematical symbols, and hand-written mathematical expression datasets as well. This dataset is comprised of CROHME 2014 [1], CROHME 2016 [2] and CROHME 2019 [3] dataset. CROHME 2014 has 101 different symbols. This dataset has 8836 for training and 986 expressions for testing. CROHME 2016 introduced more expression for test purpose where as it offers the same number of expressions for training and validation as CROHME 2014 do. Mathematical expression in the training dataset of CROHME 2019 is 9993 and for validation, it offers 986 expressions. The model can be tested on 1199 expressions of the CROHME 2019 dataset

1.1.5 Expression Rate Accuracy

Expression Recognition Rate was used to check how accurately a model recognizes an expression. It is the difference between predicted expression and ground truth expression. We also calculated ExpRate accuracy from zero to three symbol-level errors.

1.1.6 World error rate

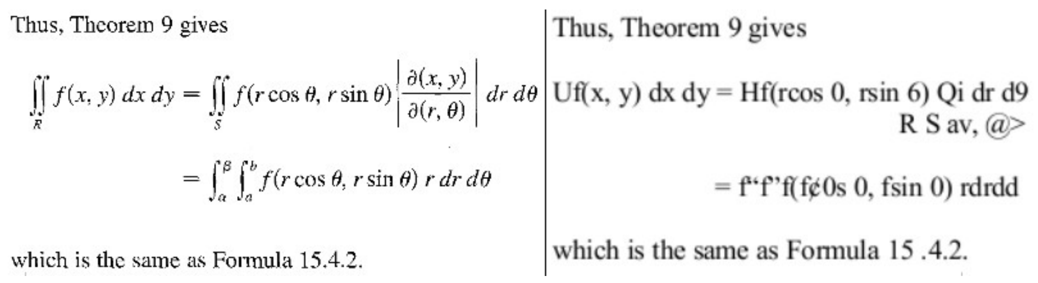
To evaluate the model at the symbol level, the word error rate (WER) is used which tells if a symbol is being substituted, deleted, or inserted into an expression or not based on ground truth.

1.2 Optical Character Recognition

In this era of technology computer has a very important role in our life. The world wide web has done the distribution and access to information very easy for everyone. This has become very useful for a common human being for understanding things that are difficult to get access to. Most of the knowledge that we have today is because of the research and experiments done in past by different people. Their finding on the related subject was available in different books. Johannes Gutenberg has done this distribution of knowledge easy after the invention of the printing press [4]. But before that, it was mostly done with books and notes written by hand. These books are widely available now in digital form on the world wide web. A lot of research has been done in this area [5]. The idea is to develop a mechanism that can convert documents irrespective of their formats, languages, and subject. In 2004 Google started working on Google books Initiative [6]. This project was based on the OCR engine developed by HP in 1985 called Tesseract [7]. The idea is to make all printed literature available online for everyone so that it can be searched easily. Due to this, Google availed to encourage further research interest in efficient and precise recognition of documents.

The process of this autonomous recognition of printed documents is called optical character recognition. The efforts for converting those books into digital form started after the invention of computers. There are some efforts made for autonomous recognition of printed documents before the invention of the computer as well. The first OCR was developed by Austrian engineer Gustav Taucschek and in 1929 he got a patent for his invention in Germany [8]. It was a mechanical device that converts pictures of letters to printed letters by matching pictures of letters with a given template. After that P.W. Handel got the patent for a similar machine in the USA in 1933 [5]. These machine requires some human intervention and can only convert one character at a time. Commercial OCR was available in the 1950s to a limited extent but they have very strict rules about the font size and structure of the documents that they can convert. But after the invention of the flatbed scanner, charge-coupled device, and Kurzweil Reading Machine in the 1970s computers can now read different documents and can accurately recognize the text. Despite the fact that they require some help in defining the text area, Kurzweil's software [9] is providing somewhat accurate results from the rest of the techniques available at that time. Calera Recognition system developed an omnifront system in the 1980s that can detect text from documents that contain pictures and columns [10].

The scientific document also contains mathematical expressions to define the problem and its solution in a more systematic way. The mathematical expression's structure and symbol used are different from the usual English language. That's why using such OCR that is used for recognizing English documents will produce garbled output as shown in figure 1.1. The reason is mathematical expressions are not just single-line text. They are comprised of multiple lines. The text written above the main text is called a superscript and the text written below the main line is called a subscript. Some other symbols such as nested fractions, matrices, etc are also part of mathematical expressions. A mathematical expression can be written anywhere in the



Thus, Theorem 9 gives

$$\iint_R f(x, y)\,dx\,dy = \iint_S f(r\cos\theta, r\sin\theta)\left|\frac{\partial(x, y)}{\partial(r, \theta)}\right| dr\,d\theta$$

$$= \int_\alpha^\beta \int_a^b f(r\cos\theta, r\sin\theta)\, r\,dr\,d\theta$$

which is the same as Formula 15.4.2.

Thus, Theorem 9 gives

Uf(x, y) dx dy = Hf(rcos 0, rsin 6) Qi dr d9
R S av, @>

= f“f’f(f¢0s 0, fsin 0) rdrdd

which is the same as Formula 15 .4.2.

Figure 1.1: On the left is an example of text that was scanned at 300 dpi from a calculus text book. To the right is the output generated by the leading open source OCR engine, Tesseract.

document. The expression written in between the normal text is called inline mathematical expression and expressions that are written in a different line from the main text are called displayed mathematical expressions. Making sure that the model is tracking the structure of expression properly is a difficult task. If a character is not recognized properly then the translated mathematical expression will be different from the expected mathematical expression. For that multiple types of research have been carried out which will be discussed in the literature review section.

1.3 Thesis Objectives

1.3.1 Research Background

Many models have been proposed in past for recognizing mathematical expressions some of which will be discussed in the literature review section. One of them has proposed bidirectional mutual learning along with an attention aggregation model [11]. The idea is to use the RNN model that will scan the feature map of the Densenet model because when the expression length is long RNN cannot translate it properly. Two RNN models will then try to decode from opposite directions and they will learn information of decoding from one another. [11] also used the attention model so that more attention is given to a such character that has not been recognized. The accuracy of this model is better than the model proposed before but there is room for increasing accuracy.

1.3.2 Problem Statement

Main issues that we have identified in the Handwritten Mathematical Expression Recognition via Attention Aggregation based Bi-directional Mutual Learning are as follow:

1. Author didn't test this model on different number of classes.
2. Mathematical expression contain different size of character. No solution is given in this model to recognize character of small size.

1.4 Contributions

To make the model perform better following contribution have been done in this regard:

1. The feature map of encoder output is concatenated with feature map obtained before last pooling layer.
2. Model is trained by changing the number of classes so that model performance can be monitored in this regard.

1.5 Thesis Organization

This thesis is organized as follow. Chapter 2 provide detailed information about the state-of-the-art methods from literature that have been proposed to defend against optical character recognition for mathematical expression. The main focus of this section is to highlight the potential issues to be resolved. Chapter 3 describes the proposed method to handle those potential issues. Results are presented at the end of proposed method. Results also provide a performance comparison of the proposed method against different OCRs. Chapter 4 summarizes all the findings and suggests the potential research improvements that can further enhance the performance of this model. And last, references are provided in chapter 5.

Chapter 2

Literature Review

There are many different methods adopted to recognize mathematical expressions. Some of these are discussed above in chapter one, introduction. Some relatively new methods that are used still today will be discussed in this section. These methods used decision trees and neural networks to detect and translate mathematical expressions. We will discuss them one by one along with their strengths and weakness.

2.1 Related Work

INFTY was introduced to convert the printed document into a structured format such as LaTeX [12]. The researcher that worked on INFTY used syntactic parsers and rule-based based structural analysis to recognize mathematical expressions. The technique used in INFTY is similar to the one used by Kacem et al [13]. Character recognition, layout analysis, manual error correction, and structural analysis are the procedures through which INFTY is recognizing a mathematical expression. It also contains a commercial OCR that can recognize and separate ordinary text from mathematical expression. One of the distinctive features of INFTY was the incorporation of the clustering technique. It was introduced to reduce misrecognition that occurs because of slight differences in shape. It breaks the expression into clusters and translates expression that is close to each other while putting them in one cluster. INFTY has shown a very good accuracy and due to high accuracy, this model was also used in InftyReader. But at some points, it requires human intervention for manual error correction. It also takes information from mathematical and linguistic grammar. When a character from a font style like greek or roman is parsed through INFTY it didn't recognize it correctly making it not suitable for the recognition of expressions written in such font style.

After the introduction of neural networks in the field of OCR, research work in the detection and recognition of ME was also done using neural networks. One of the efforts was done by Wataru et al [14]. In this research, Wataru has used U-Net for the detection of printed mathematical expressions. U-Net was used in the semantic segmentation of the biomedical image. Also, some competitor that participated in the document image binarization competition [15] used U-Net and their performance was better than other models used in that competition. It does not take any information from linguistic and mathematical grammar. Rather than using handcrafted rules to detect mathematical expression in this work end-to-end training has been done using a large dataset. The dataset used for training purpose was GTDB-1 and GTDB-2. This dataset contains images of textbook and journal pages. U-Net can also detect text line [16], and base line [17] and can do page segmentation better than other models used for ME detection [18]. This method can be used with any existing mathematical recognition pipeline. The main drawback of this method was it cannot detect font style on which the model was not trained on. It was only trained and tested on printed documents only and it can only detect and segment mathematical expressions from the main text and can not translate it into some other digital format.

Sequence-to-Sequence learning using Deep Learning for Optical Character Recognition [19] was introduced for the recognition of mathematical expressions and to convert them into LaTeX format. It was an end-to-end trainable method that do not use any prior information like INFTY. To better track the conversion process and mathematical structure, hard attention mechanism was also proposed. Using a feature vector of CNN position weights are calculated for hard attention. The feature vector of CNN is called the annotation vector. The convolutional neural network was used to convert images into a feature map that was the input of RNN. LSTM was used as an RNN to translate the feature map into a LaTeX token. In LSTM, a peephole was introduced for better conversion results. Peepholes are weighted connections between cell units and gates inside a memory unit. It allows every gate of a memory cell to access the current state of the cell even in the condition when the gate is not open. The dataset used for training was Image2latex-100K. 200K images of the mathematical expressions dataset was also been generated by the author for training and testing purpose. The model was only developed for printed documents and was not tested on handwritten mathematical expressions.

An approach was proposed by Wang et al in [20]. Here, encoder-decoder architecture also called sequence to sequence model was used for obtaining the LaTeX sequence of printed mathematical expression. VGG-VeryDeep CNN was used as an encoder along with LSTM as a decoder. Three main contributions of this research are mentioned below.

1. 2-dimensional semantic relationship differentiation among mathemat-

ical expressions. This was done by introducing sinusoidal positional encoding to represent information of spatial locality more prominently so that the feature map of the input image got augmented. The dimension of this encoding has the same size and dimension as the feature map.

2. Captured LaTeX sequence interrelationship using bilingual evaluation understudy score [21]. It was used to measure the prediction accuracy between the predicted and ground truth sequence. BLUE is widely used in evaluating machine translation done on natural language. The most commonly used score of BLEU is 4-gram on which this model is evaluated.

3. Elimination of exposure bias issue. This was done by providing the predicted LaTeX sequence instead of using ground truth. Since token alignment that is required in token level training no longer exists because this is a sequence level training.

IM2LATEX-100K was the dataset used for training and testing. The model has shown a very good accuracy as compared to the state-of-the-art model. But by using DenseNet the accuracy of this model can further be increased. Moreover, the dataset on which this model was trained, was comprised of printed mathematical expressions.

Watch, attend, and parse [22], a model to recognize and translate handwritten mathematical expressions using neural networks into LaTeX markup sequences. It was based on an encoder-decoder model with an attention model to understand the ME sequence so that its spatial information won't be lost. Segmentation is required in ME recognition which is done by the WAP attention mechanism. To tune the watcher, the parser also provides contextual information. It also reduces the computational cost that is required to process a mathematical expression grammar. To process large-scale handwritten mathematical expressions a deep fully convolutional network was used as an encoder (watcher). To make sure that each character of ME got properly recognized, the attention model was used. Incorrect alignment of mathematical expressions may result in over or under parsing which results in a high error rate. To resolve this issue alignment history was monitored in the attention model. A coverage vector was added in the attention model in this regard. It monitors the past alignment information so that the model gives attention to such characters that are not translated before. A feature map along with alignment information was given to the parser which is a GRU decoder to produce a LaTeX sequence based on the feature map. End-to-end training was done without any manual information insertion. The dataset used for training was CROHME 2014 and CROHME 2016. This model outperformed most of the new methods but it cannot translate complex and long mathematical expressions. Moreover, the encoder used was not suitable enough for obtaining a feature map from mathematical expression.

Zhang et al [23] used a densely connected neural network to improve the performance of the encoder. The advantage of using densely connected neural networks is they can extract more features and effective gradient propagation can be achieved even when the dataset is not very big. Due to the pooling operation in the densely connected neural network most of the fine-grained details in mathematical expressions are dropped that's why the multiscale attention model was introduced. This work is the improved version of [22]. It also translates handwritten mathematical expressions into LaTeX sequences. Multiscale attention was implemented using low-resolution feature maps and high-resolution feature maps. To get a large receptive field, a low-resolution feature map is used and to get a small receptive field high-resolution feature map is used. The decoder will then generate a LaTeX string on the basis of these feature maps. Due to the above-mentioned innovation, the model has outperformed WAP. Along with that, the decoder used is GRU. But when the length of mathematical expression is long, the translation accuracy drops due to the fact that the decoder cannot remember long mathematical expressions.

Chapter 3

ME Recognition

In short, neural networks are very helpful in detecting mathematical expressions. Using an encoder-decoder-based model, a LaTeX sequence can be generated of the corresponding image. There are some points that are not covered previously in any research. As we have seen in research done before, researchers have introduced an attention mechanism so that model can track properly the structure and sequence of mathematical expression [19], [22], [23]. Also, tremendous efforts have been done in finding which convolutional neural network can perform as an encoder [23]. But there are still some issues that are not addressed previously. Some of them are

1. High error rate while recognizing long mathematical expressions
2. Details dropped by models due to variation in the size of mathematical symbols.

In this work, we have tried to address the above-mentioned problems. We have used a recognition model mentioned in [11]. Two GRU models are used in inverse to each other to decode the feature map obtained from the encoder so that issues that are created due to long mathematical expressions can be resolved. The detail of this method will discuss in the methodology. We are introducing multi-scale feature map extraction with this existing model so that more fine feature maps can be extracted.

3.1 Methodology

For better feature extraction, we used a deep neural network model [24]. It has shown better results compared to VGGnet and other neural network models [23] when used in such scenarios. The model consists of a convolutional layer but each layer takes input from every previous layer. Therefore, the total connection in this model will be $\frac{L(L+1)}{2}$. This will help in better propagation of features, resolve the issue of vanishing of gradient, and reduce the number of parameters.

A schematic representation of a DenseNet is shown in 3.1. The model contains Dense Blocks. These dense blocks are interconnected using transition layers. Dense Blocks contain multiple Bottleneck layers. The composition of the Bottleneck layers is Batch Normalization followed by the ReLU activation layer followed by 1x1 convolutional layer followed by Batch Normalization followed by ReLU activation layer and finally 3x3 convolution. To improve the computational efficiency 1x1 convolution layer is used before the 3x3 convolution. The transition layer used starts with batch normalization. After that 1x1 convolution is applied and in the end, 2x2 average pooling is applied. After every layer, the feature increases in DenseNet. This is a hyperparameter called growth rate. If input feature map is k_o the the input of *lth* layer would be $ko + k(l - 1)$. This k is the growth rate in DenseNet. All layers in DensNet are connected with subsequent layers. This interconnection of the layer will provide better propagation of the gradient. The

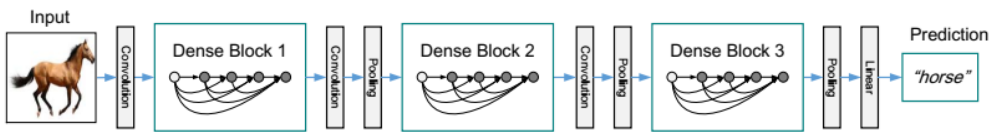


Figure 3.1: Visual representation of DenseNet

Densenet we used consists of three Dense blocks. After each Dense Block, an average pool is applied to the output feature map. The average pool will decrease the size of the feature map. That is the reason why the output feature map of the last Dense Block will have low resolution. Some of the characters or symbols that are visible in the input image won't be present in this low-resolution feature map. The model will not decode those characters or symbols. To resolve this issue we concatenated a feature map obtained before the second average pool layer. We pass it through a kernel and resize this feature map to match the size of the low-resolution feature map. The image of our model is illustrated in figure 3.2. The architecture of the model used in this research is mentioned in table 3.1. The output of the encoder will be a three-dimensional feature map H x W x D. It is an M dimensional vector i.e. $\mathbf{a} = a_1, a_2, a_3, ..., a_M$. Here a_i belongs to $\mathbb{R}^{\mathbb{D}}$ and $M = Hx\,W$

The attention model we used is called the attention aggregation model. The one used in [11]. A block diagram of AAM is shown in figure 3.3. Due to the fact that AAM consists of small as well as large kernels, it will focus on local as well as global features. Covergance-based attention will help the model find alignment information. This is done by giving a high probability to an area that is not been decoded in past. Variables like hidden state coverage attention and feature map allow AAM to calculate weights of current attention. With these parameters, it also computes context vectors. This attention model was inspired by the Inception model [25]. This AAM model uses feature map $\mathcal{F}$, coverage attention β_t, and hidden states $\hat{h}_t$ to com-

Layers	DenseNet
Convolution	7 $\times$ 7 conv, stride 2, padding 3
Dense Block (1)	$\begin{pmatrix} 1x1conv \\ 3x3conv \end{pmatrix} x16$
Transition Layer (1)	1 x 1 conv, stride 1
Dense Block (2)	$\begin{pmatrix} 1x1conv \\ 3x3conv \end{pmatrix} x16$
Transition Layer (2)	1 x 1 conv, stride 1
Dense Block (3)	$\begin{pmatrix} 1x1conv \\ 3x3conv \end{pmatrix} x16$
Kernel for resizing high dimension feature map	2 x 2 conv, stride 2

Table 3.1: DenseNet architecture

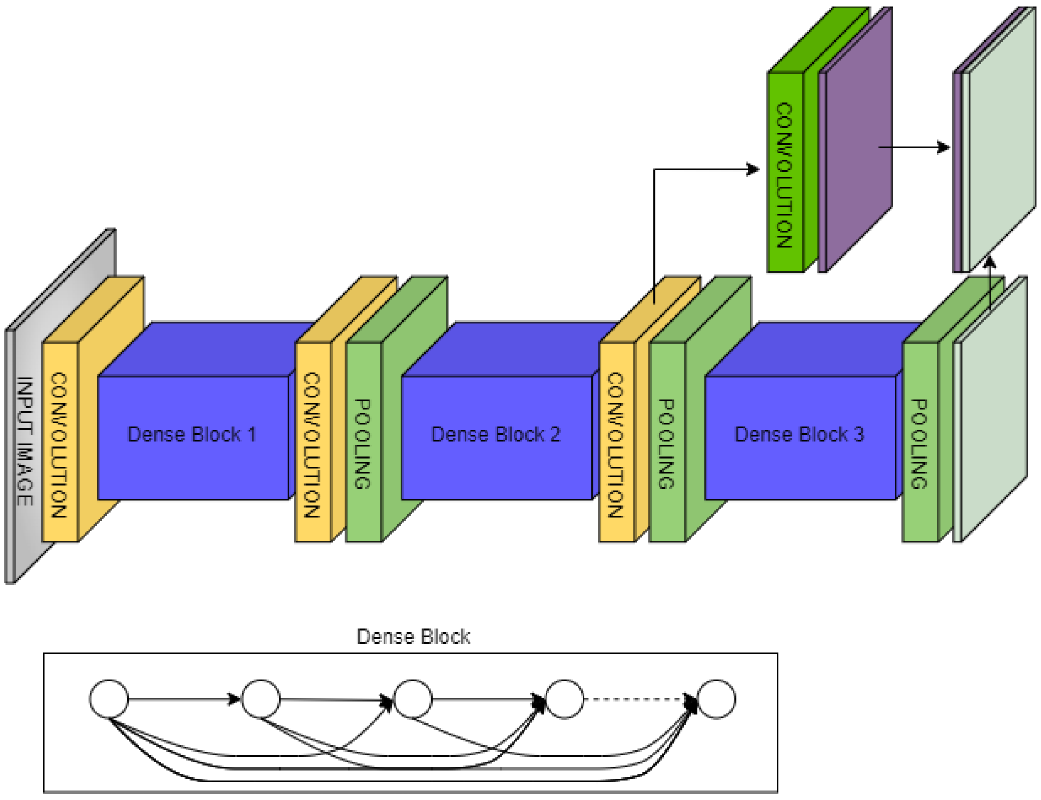


Figure 3.2: Model Proposed in this research using DenseNet

pute the small and large scale attention. Using these attentions, it calculates weights of current attention α_t and context vector c_t. The context vector can be calculated as

$$c_t = \sum_{i=1}^{M} \alpha_{t,i} a_i$$

α_t is the score of attention at step l. To calculate the attention of current attention α_t is used such that

$$\alpha_t = v_a^T \tanh\left(W_{\hat{h}} \hat{h}_t + U_f \mathcal{F} + W_s A_s + W_l A_l\right)$$

$W_{\hat{h}}$, W_l and W_s are trainable weight matrix. $\hat{h}_t$ is GRU's hidden state and A_s and A_l can be computed as:

$$A_s = U_s B_t, A_l = U_l B_t$$

In the above equations, U_s and U_l are convolutions of large and small sizes of the kernel. The size of those kernels can be 5 and 7 respectively. Here, β_t which is coverage attention is the summation of all previous probabilities of attention. β_t is initialized as zero vector and then computed by

$$\beta_t = \sum_{l=1}^{t-1} \alpha_l$$

Previously in WAP [22] and HMER [23], the decoding is done only in one direction. This is the reason the model cannot decode long mathematical expressions in those models. That's why we adopted the technique mentioned in [11] where two unidirectional decoder blocks are used. The working of both of the blocks is the same but the difference is in direction of which they are decoding. One block is decoding a mathematical expression from Left to right and the other block decodes it from right to left. Eventually, both blocks will learn from each other information to overcome long-term dependency. A block diagram of the decoder is shown in figure 3.4.

As we mentioned earlier each unidirectional decoder block is GRU, the same that has been used in WAP [22]. A mathematical representation of this block is given below.

$$\hat{h}_t = GRU_1(h_{t-1}, E\overrightarrow{y}_{t-1})$$

$$h_t = GRU_2(\hat{h}, c_t)$$

Where h_t is the current hidden state of GRU, E is the embedding matrix, and $\overrightarrow{y}_{t-1}$ previous prediction output. $\langle sos \rangle$ and $\langle eos \rangle$ are added in the

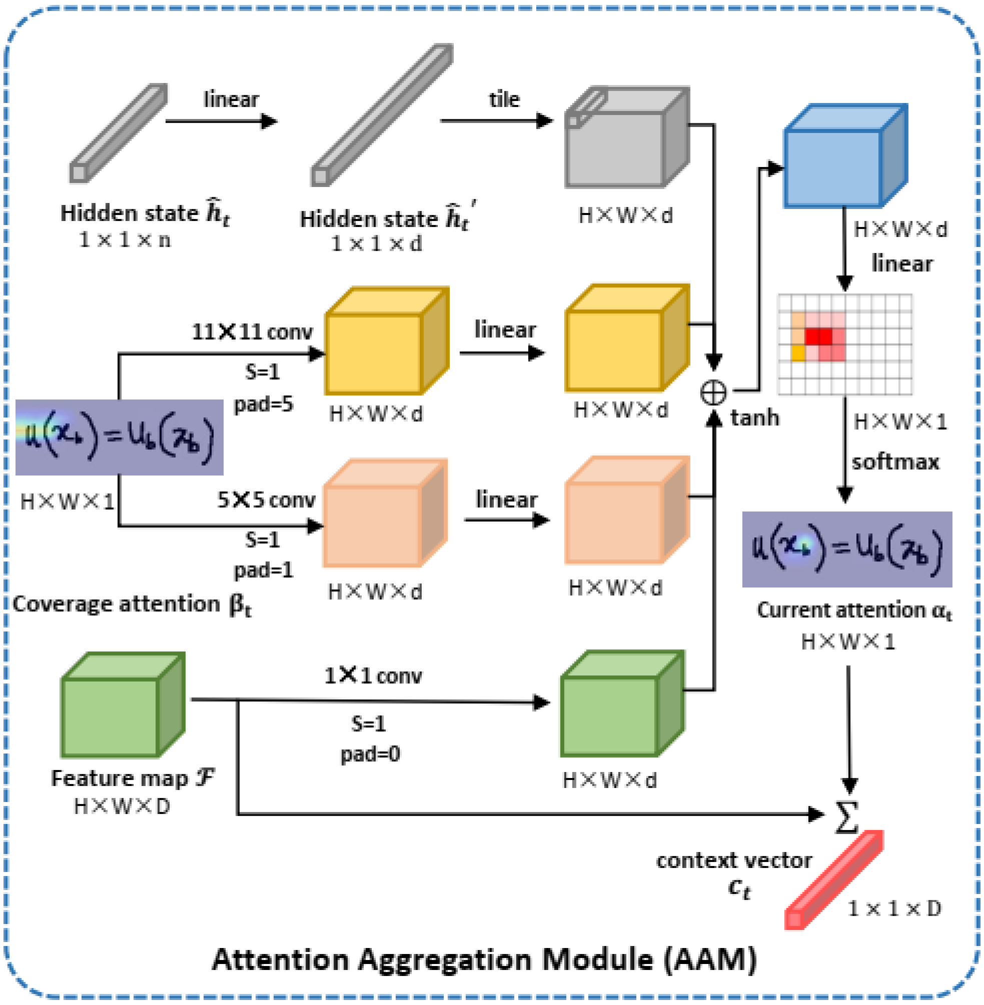


Figure 3.3: Attention Aggregation Model

LaTeX string label to represent the start and end of the string respectively. If the targeted string is $Y = \{Y_1, Y_2, Y_3, ..., Y_T\}$ then we denote left to right LaTeX string as $\overrightarrow{Y_{l2r}} = \{\langle sos\rangle, Y_1, Y_2, Y_3, ..., Y_T, \langle eos\rangle\}$ and right to left string as $\overleftarrow{Y_{r2l}} = \{\langle eos\rangle, Y_T, ..., Y_3, Y_2, Y_1, \langle sos\rangle\}$. The probability is calculated for each symbol in the L2R direction sequence of mathematical expression as follows:

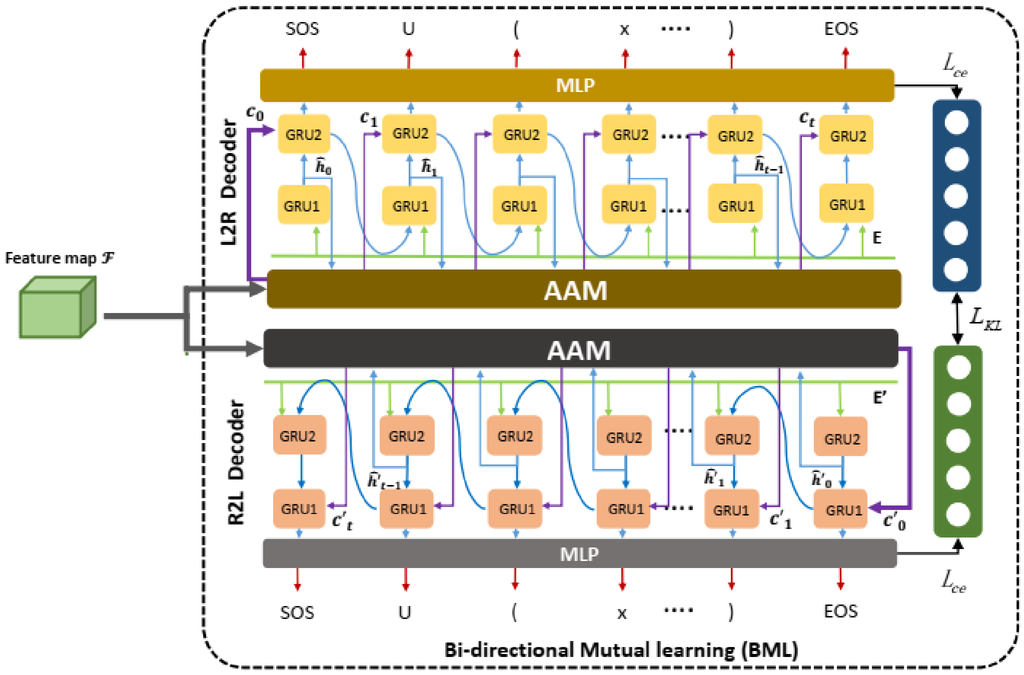


Figure 3.4: Decoder Block Diagram

$$p(\overrightarrow{y}_t \mid \overrightarrow{y}_{t-1}) = W_o max(W_y E \overrightarrow{y}_{t-1} + W_h h_t + W_t c_t)$$

For R2L it can be computed as:

$$p(\overleftarrow{y}_t \mid \overleftarrow{y}_{t-1}) = W'_o max(W'_y E' \overleftarrow{y}_{t-1} + W'_h h'_t + W'_t c'_t)$$

$W_o \in \mathcal{R}^{K\times d}$, $W_y \in \mathcal{R}^{d\times n}$, $W_h \in R_{d\times n}$ and $W_t \in R^{d\times D}$ are trainable weight matrices. Let d denote the attention dimension, K denote the number of LaTeX symbols and n denote the GRU dimension.

The main purpose is to decrease the difference between the probabilities calculated by each uni-directional decoder. To quantify the difference Kullback-Leibler (KL) loss is used. to give more information soft probability is used that is been generated by the model. The soft probability for L2R branch can be computed by

$$\sigma(\overrightarrow{Z}_{i,k}, S) = \frac{\exp(\overrightarrow{Z}_{i,k}/S)}{\Sigma_{j=1}^{K} \exp(\overrightarrow{Z}_{i,k}/S)}$$

Where S is the parameter to generate soft labels. The KL difference of L2R and R2L probabilities can be calculated as:

$$KL = S^2 \sum_{i=1}^{T} \sum_{j=1}^{K} \sigma(\overrightarrow{Z}_{i,k}, S) \log \frac{\sigma(\overrightarrow{Z}_{i,k}, S)}{\sigma(\overleftarrow{Z}_{T+1-i,j}, S)}$$

To make sure that another branch is contributing equally in probability distribution along with ground truth, S^2 is inserted in the above equation. $\overleftarrow{Z}_{T+1-i,j}$ is log of R2L probability and $\overrightarrow{Z}_{i,k}$ is log of L2R probability.

The loss function is computed using cross entropy losses of left-to-right and right-to-left decoder branches. The formula for this loss is

$$L = {L_{ce}}^{l2r} + {L_{ce}}^{r2l} + \lambda L_{KL}$$

A hyperparameter λ is used to balance KL divergence loss and loss of recognition. In the above equation cross entropy for each branch of the decoder can be defined as

$${L_{ce}}^{l2r} = \sum_{i=1}^{T} \sum_{i=1}^{K} - Y_{i,j} \log(\overrightarrow{y}_{i,j})$$

$${L_{ce}}^{r2l} = \sum_{i=1}^{T} \sum_{i=1}^{K} - Y_{i,j} \log(\overrightarrow{y}_{T+1-i,j})$$

Consider that the LaTeX sequence generated by decoder is $\overrightarrow{Y_{l2r}} = \{\langle sos \rangle, Y_1, Y_2, Y_3, ..., Y_T, \langle eos \rangle\}$. The length of this LaTeX sequence is T, then the one-hot ground-truth label at any time step i can be $Y_i = \{x_1, x_2, x_3, ..., x_K\}$ where $x_i \in \{0, 1\}$. k_th symbol's softmax probability can be computed as

$$\overrightarrow{y_{i,j}} = \frac{\exp(\overrightarrow{Z_{i,k}})}{\sum_{j=1}^{K} \exp \overrightarrow{Z_{i,k}}}$$

3.2 Results

Our proposed model is end-to-end trainable without the use of predefined expression grammar. The objective of the training is to maximize the probability of a predicted word. The process of calculating probability is described in the previous section. Cross-Entropy is used as an objective function.

A block diagram of the dense Encoder that we used is shown in figure 3.1. The dimension of GRU decoder is 256 and the embedding dimension is set to 256. Internal covariant shifts are reduced by training the model using batch normalization. Adalta optimizer is used for optimization purposes with learning rate decay enable. This learning rate starts from 1 and whenever the word error rate won't reduce for 15 epochs, the learning rate will be halved from its original value. Over-fitting is also a major issue in neural networks which was prevented here using dropout inside dense blocks and transition layers. The dropout value used to be 20 percent. The weights are initialized using Kaiming Initialization [26] so that non-linearity induced due to the activation function can be handled. This initialization method works well with ReLU activation. We test and validate our model on a dataset that is available online for the public called CROHME [3]. This dataset includes different datasets that were received from participants. Over the year its size and complexity have increased. The number of mathematical expressions in CROHME 2014 dataset was 8836 and it comprises 86000 symbols. CROHME 2014 dataset also contains 986 mathematical expressions. The number of symbols in the test dataset was 6000. The number of different mathematical symbols was 101 in CROHME 2014 dataset. The training dataset of CROHME 2016 dataset was the same as the dataste of CROHME 2014 dataset. The University of Nantes labeled the test dataset of CROHME 2016. CROHME 2019 dataset was also used for testing our proposed model. It contains 1199 mathematical expressions.

Usually, such expressions are stored in InKML file format. InKML file format contains information on points that makes the set of trace and ground truth of mathematical expression. The sequence in which that label is written can be a LaTeX or MathML. Some other information like handedness (right or left) gender and age of the writer and identification is also given in the file but this is irrelevant to this work.

Expression rate is used to evaluate how accurately a mathematical expression is translated into its LaTeX sequence. The predicted LaTeX sequence is compared with the ground truth provided in InkML file. We use not only the Expression rate but also the evaluated model up to a two-word error rate. Another type of evaluation method called world error rate is used for the evaluation of OCRs. In this method errors like insertion, substitution, and deletion can be calculated. These errors can be understood by an example. If the target sequence is $x + y = z$ and the predicted sequence is $x+ = z$ we consider it a deletion error and if the predicted sequence is $x + +y = z$ we call it an insertion error. An error is a substitution if the output sequence is $x+x = z$. The word error rate can be calculated using the following formula:

$$WER = \frac{N^W_{sub} + N^W_{del} + N^W_{ins}}{N^W} = \frac{N^W_{sub} + N^W_{del} + N^W_{del}}{N^W_{sub} + N^W_{del} + N^W_{cor}}$$

In the above equation N^W_{sub} represents the number of characters being substituted, N^W_{del} represents the number of characters being deleted and N^W_{ins} represents the number of characters being inserted in a predicted output.

N_{cor}^{W} is the total number of correctly predicted characters and N^{W} is the total number of characters in label mathematical sequence.

In figure 3.5, attention visualization of the attention aggregation model is shown. The attention block will try to create a relationship among the given mathematical expression by learning the location of the character. Relationships between symbols that are written on the middle line are easy to learn but relationships with superscripts and subscripts are not easy to learn. Then attention block must focus on the top right corner to detect superscripts and the bottom left corner to focus on subscripts. In figure 3.5 we have shown step by step how our model will visualize mathematical expression. The expression is $4^2 + 4^2 + \frac{4}{4}$. Whenever it encounters a superscript it automatically put $"\hat{}"$ in the equation. Not only that our model learns arrangements that unequivocally compare with human instinct when it encounters special symbols like '?', '*', '%', etc.

We first train and test on CROHME 2014 and 2016 datasets. The number of symbols in that dataset was 111. After that, we increased the dataset by including more expressions in it. Now, the number of symbols is 131, and the number of expressions is also increased in the training, validation, and test dataset. The model is optimized with the help of an Adadelta optimizer. The learning rate of this optimizer is set to 1. For a different number of classes, its learning rate changes differently whenever the WER does not change for a long time. λ is set to 0.5 in the loss function. After concatenating two feature maps i.e. high resolution and low-resolution feature maps, the size of the feature map from the encoder is 1284. The growth rate for the encoder is set to 24.

Expression Recognition Rate was used to check how accurately a model recognizes an expression. It is the difference between predicted expression

System	Classes	ExpRate	ExpRate ≤ 1	ExpRate ≤ 2	WER
ABM	111	52.8398	72.1095	79.0061	11.9771
Our Model	111	**54.3611**	**72.8195**	**79.8174**	**11.5997**
ABM	131	49.6599	57.5510	65.5782	20.2491
Our Model	131	48.8435	**58.7755**	**67.3469**	**18.4602**

Table 3.2: Comparision with ABM

and ground truth expression. We also calculated ExpRate accuracy from zero to three symbol-level errors. To evaluate the model at the symbol level word error rate (WER) is used which tells if a symbol is being substituted, deleted, or inserted into an expression or not based on ground truth. The comparison of our model with ABM is given in Table 3.2.

3.3 Discussion

From the research done in [11], we have seen that whenever the number of tokens increases the expiration rate accuracy decrease. This can also be seen in table 3.2. By concatenating low and high feature maps in the encoder, we not only increased the expiration rate accuracy of the model with 111 classes but also increased the accuracy of the model with 131 classes.

By comparing our model with ABM, it can be seen that our model is detecting small details that ABM cannot detect as shown in figure 3.6. In this example a super script was not detected by ABM but this super script was detected by our model. Similarly, we have also try to resolve the issue of long term dependency. Image shown in figure 3.7 was recognized by ABM as

$$\frac{2}{\sqrt{3}-1} \times \frac{\sqrt{3}+1}{\sqrt{3}+1} = \frac{2\sqrt{3}+1}{3-1} = \sqrt{3}+1$$

The same image was recognized by our model as

$$\frac{2}{\sqrt{3}-1} \times \frac{\sqrt{3}+1}{\sqrt{3}+1} = \frac{2(\sqrt{3}+1)}{3-1} = \sqrt{3}+1$$

but the ground truth for that image is

$$\frac{2}{\sqrt{3}-1} \times \frac{\sqrt{3}+1}{\sqrt{3}+1} = \frac{2(\sqrt{3}+1)}{3-1} = \sqrt{3}+1$$

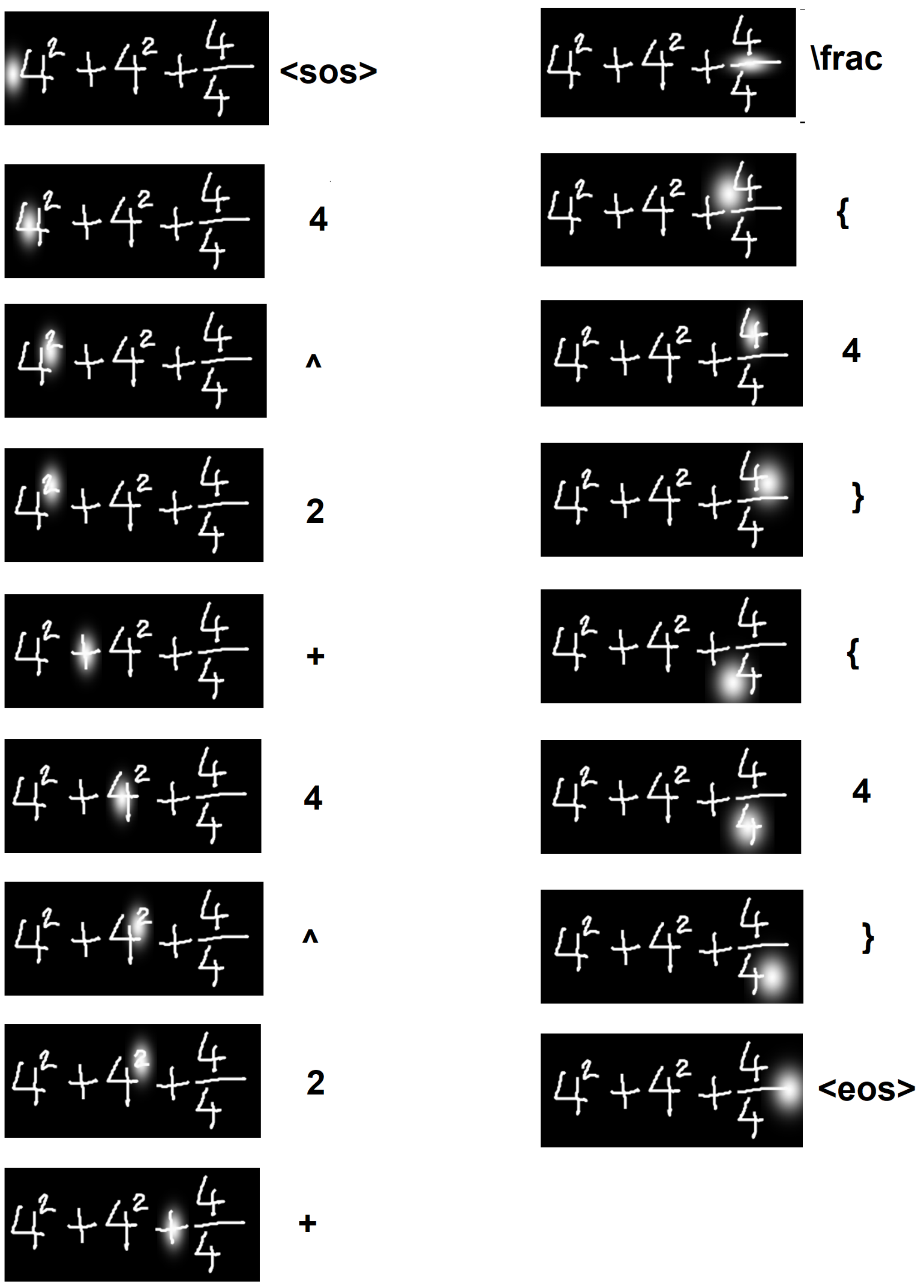
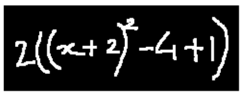


Figure 3.5: attention visualization of LaTeX expression “4ˆ2+4ˆ2+\frac{4}{4}”



Recognized by ABM: 2 (x + 2) ^ { 2 } - 4 + 1)
Recognized by Our Model: 2 ((x + 2) ^ { 2 } - 4 + 1)
Groud Truth: 2 ((x + 2) ^ { 2 } - 4 + 1)

Figure 3.6: Recognition comparison with ABM

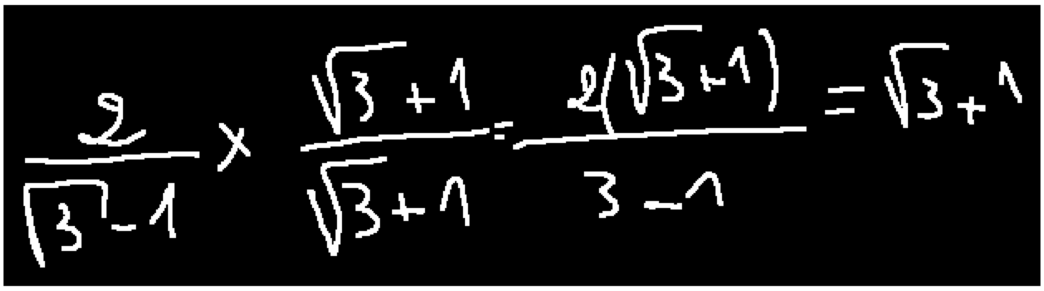


Figure 3.7: Recognition comparison with ABM

Chapter 4

Conclusion and Future Work

4.1 Summary

In this study, we have increased the accuracy of ABM by including an extra branch in the encoder. This branch has a high-resolution feature map that we concatenated with low-resolution feature map of encoder output. We have noticed that by increasing the LaTeX classes the performance of ABM decreases so we tested our model by increasing the number of LaTeX classes and comparing it with the state-of-the-art ABM model.

4.2 Future Work

One of the possible future directions is related to the type of dataset used in this work. We have used the CROHME dataset. By including more LaTeX tokens of mathematical expression the model can be trained and more mathematical characters can be recognized. Some of the possible roads ahead can be:

- Use this model with a mathematical detection model like [14] and use it with the normal OCR model so that it can not only convert normal text but also can extract ME from documents like research papers and convert them into LaTeX sequence.

- Using models like LSTM as a decoder or any other RNN network that can better extract text from feature maps can be used to increase the performance. Similarly using an encoder that can better extract feature maps from input images can also be very helpful in achieving better accuracy.
- There are some other attention models that were proposed recently [20]that can better focus on the untranslated text. Using those as an attention model can be very helpful in this regard.

Bibliography

[1] H. Mouchère, C. Viard-Gaudin, R. Zanibbi, and U. Garain, "Icfhr 2014 competition on recognition of on-line handwritten mathematical expressions (crohme 2014)," in *2014 14th International Conference on Frontiers in Handwriting Recognition*, pp. 791–796, 2014.

[2] H. Mouchère, C. Viard-Gaudin, R. Zanibbi, and U. Garain, "Icfhr2016 crohme: Competition on recognition of online handwritten mathematical expressions," in *2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR)*, pp. 607–612, 2016.

[3] M. Mahdavi, R. Zanibbi, H. Mouchere, C. Viard-Gaudin, and U. Garain, "Icdar 2019 crohme + tfd: Competition on recognition of handwritten mathematical expressions and typeset formula detection," in *2019 International Conference on Document Analysis and Recognition (ICDAR)*, pp. 1533–1538, 2019.

[4] J. R. Bruce, *Mathematical expression detection and segmentation in document images.* PhD thesis, Virginia Tech, 2014.

[5] H. F. Schantz, *History of OCR, optical character recognition.* Recognition Technologies Users Association, 1982.

[6] L. Vincent, "Google book search: Document understanding on a massive scale," in *Ninth International Conference on Document Analysis and Recognition (ICDAR 2007)*, vol. 2, pp. 819–823, IEEE, 2007.

[7] S. V. Rice, F. R. Jenkins, and T. A. Nartker, "The fourth annual test of ocr accuracy," tech. rep., Technical Report 95, 1995.

[8] Heather, "Optical character recognition (ocr) explained — everything you need to know."

[9] A. Kleiner, "A description of the kurzweil reading machine and a status report on its testing and dissemination," *Bull Prosthet Res*, vol. 10, no. 27, pp. 72–81, 1977.

[10] M. Bokser, "Omnidocument technologies," *Proceedings of the IEEE*, vol. 80, no. 7, pp. 1066–1078, 1992.

[11] X. Bian, B. Qin, X. Xin, J. Li, X. Su, and Y. Wang, "Handwritten mathematical expression recognition via attention aggregation based bi-directional mutual learning," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 36, pp. 113–121, 2022.

[12] M. Suzuki, F. Tamari, R. Fukuda, S. Uchida, and T. Kanahori, "Infty: an integrated ocr system for mathematical documents," pp. 95–104, 11 2003.

[13] A. Kacem, A. Belaid, and M. B. Ahmed, "Extrafor: automatic extraction of mathematical formulas," in *Proceedings of the Fifth International Conference on Document Analysis and Recognition. ICDAR'99 (Cat. No. PR00318)*, pp. 527–530, IEEE, 1999.

[14] W. Ohyama, M. Suzuki, and S. Uchida, "Detecting mathematical expressions in scientific document images using a u-net trained on a diverse dataset," *IEEE Access*, vol. 7, pp. 144030–144042, 2019.

[15] I. Pratikakis, K. Zagoris, G. Barlas, and B. Gatos, "Icdar2017 competition on document image binarization (dibco 2017)," in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, vol. 1, pp. 1395–1403, IEEE, 2017.

[16] G. Renton, C. Chatelain, S. Adam, C. Kermorvant, and T. Paquet, "Handwritten text line segmentation using fully convolutional network," in *2017 14th IAPR International Conference on Document Analysis and Recognition (ICDAR)*, vol. 5, pp. 5–9, IEEE, 2017.

[17] M. Fink, T. Layer, G. Mackenbrock, and M. Sprinzl, "Baseline detection in historical documents using convolutional u-nets," in *2018 13th IAPR International Workshop on Document Analysis Systems (DAS)*, pp. 37–42, IEEE, 2018.

[18] S. Capobianco, L. Scommegna, and S. Marinai, "Historical handwritten document segmentation by using a weighted loss," in *Artificial Neural Networks in Pattern Recognition: 8th IAPR TC3 Workshop, ANNPR 2018, Siena, Italy, September 19–21, 2018, Proceedings 8*, pp. 395–406, Springer, 2018.

[19] V. Mishra and D. Kaur, "Sequence-to-sequence learning using deep learning for optical character recognition (ocr)," in *2018 International Conference on Computational Science and Computational Intelligence (CSCI)*, pp. 324–329, 2018.

[20] Z. Wang and J.-C. Liu, "Translating math formula images to latex sequences using deep neural networks with sequence-level training," *International Journal on Document Analysis and Recognition (IJDAR)*, vol. 24, no. 1-2, pp. 63–75, 2021.

[21] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pp. 311–318, 2002.

[22] J. Zhang, J. Du, S. Zhang, D. Liu, Y. Hu, J. Hu, S. Wei, and L. Dai, "Watch, attend and parse: An end-to-end neural network based approach to handwritten mathematical expression recognition," *Pattern Recognition*, vol. 71, pp. 196–206, 2017.

[23] J. Zhang, J. Du, and L. Dai, "Multi-scale attention with dense encoder for handwritten mathematical expression recognition," in *2018 24th international conference on pattern recognition (ICPR)*, pp. 2245–2250, IEEE, 2018.

[24] G. Huang, Z. Liu, L. Van Der Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 4700–4708, 2017.

[25] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 1–9, 2015.

[26] K. He, X. Zhang, S. Ren, and J. Sun, "Delving deep into rectifiers: Surpassing human-level performance on imagenet classification," in *Proceed-*

ings of the IEEE international conference on computer vision, pp. 1026–1034, 2015.