Retrieving Math Information Based on Equation Detection and Recognition within Digital Images

Angel Wheelwright and Yiu-Kai Ng *

Computer Science Department, Brigham Young University, Provo, Utah, USA
Email: angelwhwrt@gmail.com (A.W.); ng@compsci.byu.edu (Y.-K.N.)
*Corresponding author

***Abstract*—In the USA, math proficiency levels these days are lower than ever before, which is problematic, since math is commonly used throughout life and math enables people to better solve problems, understand patterns, quantify relationships, and make predictions of the future. While Math Information Retrieval (MIR) as an area of study is relatively new, it is essential and provides a means to search for relevant sources of math information to those who are studying math, something which is difficult to do for people without prior knowledge of a specific math subject area they are looking for. In order to develop a robust MIR system, designers must be able to process Math Equations (MEs) to a format that the system can use, which is difficult due to various formats math information are stored in, including visual images and document texts. In solving this problem, we propose a ME extraction system that (i) applies a one shot object detector to identify math equations in digital images using an efficient neural architecture search method and (ii) employs a Sequence-to-Sequence (Seq2Seq) encoder-decoder system to recognize math equation symbols based on the Bayesian Neural Network (BNN) row encoding. The proposed system balances speed and accuracy of a Math Information Retrieval (IR) system.**

***Keywords*—image detection, image recognition, math questions, math answers**

I. Introduction

Math is a major contributor to many areas of study, and gives someone skills that (s)he can use across other subjects and different job roles. It makes a person better at solving problems. Math is also considered a universal language, since it conveys quantitative properties and values as well as how processes work, and is used in many different parts of life, such as scheduling, cooking, finances, measurements, and organization. Unfortunately, math proficiency levels for people around the world have dropped drastically, particularly in the USA. According to the National Assessment of Educational Progress [1], 12th graders in the USA are considered to be proficient in math if they have a score of 176 or higher, but the grade average in 2019, the most recent recorded year, was 150. The national average for students' math proficiency in US public schools was merely 38% in the year of 2023 [2]. To make matters worse, it is very difficult for people to locate viable sources of math information in order to learn math or familiarize themselves with an area of study or research which involves math, especially if they are not familiar with the subject area. Since understanding of math is beneficial yet so difficult to attain, having systems in place which can reliably locate, extract, and return math resources for people to use and learn from is essential.

Math Information Retrieval (MIR), a relatively new field of study which involves organizing, storing, retrieving, and evaluating math information from document repositories, has been developed to retrieve and rank math information to prospective users. With the design goal of assisting users to retrieve relevant math information, MIR systems aid users in increasing their understanding of math concepts. One of the main design issues of MIR systems, however, is to process sources of math information so that they become visible and accessible to the users. While some of these sources are just stored in textual format, others are archived in a visual format, whether this be a PDF, a physical document, or a digital image. Besides processing textual math information in math questions and answers posted by users on a social media website, such as Mathematics Stack Exchange, a sophisticated Math Information Retrieval (IR) system is expected to handle the plethora of other sources of digital math information, such as geometric figures or graphs. Doing so would improve Science, Technology, Engineering, and Mathematics (STEM) education by allowing more sophisticated search, storage, and production of math information, as well as enable automatic document digitization, which is considerably more efficient than manually digitizing documents. Extraction of math from visual sources allows online learning platforms to more easily interpret the content of user submitted images for answering questions, giving recommendations, and retrieving information, which is particularly helpful for returning relevant information to users which is either difficult to transcribe textually or more visual in nature, such as a complex multi-layer equations, graphical diagrams, and tabular data. Converting Math Equation (ME) images into textual format, however, is non-trivial.

While natural language text is arranged in relatively easy to parse lines, having characters aligned in a single

Manuscript received February 22, 2025; revised April 2, 2025; accepted April 23, 2025; published July 15, 2025.

dimension, math notation may appear both in-line with surround text or isolated from the rest of a document as an image. To further complicate matters, some math equation symbols can be used for multiple purposes. For instance, $A \bullet B$ could be referring to algebraic multiplication, matrix multiplication, or concatenation, whereas $f \circ g$ can refer to the Hadamard product with matrix multiplication or for function composition. Moreover, subscripts and superscripts alter the sizes and location of the equation symbols involved, and these symbols can be arranged in two spatial dimensions rather than one. All of these factors are critical for accurately extracting a math equation with the intended meaning. Even when math formulas are correctly extracted from images, most existing methods for equation extraction designed with accuracy in mind rather than speed, with ScanSSD-XYc being one of the only methods listed in the present that even addressed speed when it came to equation extraction systems. This is problematic, since MIR systems require both reasonable accuracy and real time speeds in order to be deployed in real world environments, and equation extraction systems would fall under similar constraints.

In solving the design problems mentioned above, we propose a MIR system that is capable of extracting digital math information by detecting and recognizing math equations in images and converting them into a usable text-based format, namely LaTeX, with high efficiency and accuracy, a contribution to the Math IR community [3]. For the proposed system we specifically focus on extracting math equations from digital or scanned documents, one of the most commonly existing and used mediums for storing math information online. The extraction model consists of two components: a detection model and a recognition model. To detect math equations in images, a Fully Convolutional One-Stage (FCOS) object detection model is adapted for identifying different kinds of math formulas in images, creating labels, and performing bounding box regression with the addition of Fast And Diverse (FAD) with Representation Sharing (RepShare) to improve model efficiency [4]. To recognize and retrieve math equations in the resulting bounding boxes as text, an encoder-decoder architecture using soft attention with Bayesian Neural Network row encoding is utilized to convert the images into LaTeX [5].

II. Literature Review

While Math Equation Detection and Recognition (MEDR) is a relatively new area of study, work has been done in this area over a decade ago. It focuses on extraction from PDFs and images that include printed or handwritten math equations.

A. Math Equation Detection Models

Even though there have been methods created to process math equations based on techniques other than Machine Learning (ML), ML is still used more frequently in recent times, with Support Vector Machines (SVM), K-Nearest-Neighbor (KNN), Convolutional Neural Network (CNN), and Long Short-Term Memory (LSTM) being the most commonly-used techniques [6–8]. Some previous work which relates to math equation detection typically utilizes some form of CNN, as this enables storing feature information and scanning for features which indicate ME locations. There are several variants of this particular method. For instance, ScanSSD and ScanSSD-Xyc slide windows over images using a CNN to select equation bounding boxes. Chu and Liu [9] separate text from the rest of the document and then employ a SVM to determine if the segmented line is a math equation E to classify E as either inline or isolated. A similar method segments the text before running a CNN for feature extraction [8]. Other techniques pair a U-Net with a CNN, or a Conditional Random Field (CRF) with a Recurrent Neural Network (RNN) [7]. Yet another approach is to use previously implemented object detectors for ME detection, such as using Faster R-CNN or Cascade Mask R-CNN [6].

B. Math Equation Recognition Models

The earliest math equation recognition methods typically used conventional OCR with SVMs for classification. Most methods, which were created later, utilized some form of encoder-decoder architecture, usually included a LSTM variant or attention [10]. Some of the more recent methods use Visual Transformers (ViT) as well [3]. The persistent use of encoder-decoder architectures for this problem indicates that this approach is an effective solution for this problem, and is one of the few currently existing methods which can be configured to convert images to text in a similar fashion to image captioning, which is needed for math equation recognition. The majority of these methods have some form of recurrence or attention in place to enable keeping track of sequential and spatial data, due to the importance of contextual and semantic information for recognizing equation text in images. We have found no existing work which combines Bayesian uncertainty modeling with math equation recognition, let alone combines anchor-free detection (such as FCOS) with Bayesian uncertainty modeling for math equation recognition.

III. The Proposed MIR System

In this section, we detail the design methodology of the proposed math image detection and recognition system for retrieving math information.

A. Our Math Equation Detection Model

The design goal for our math equation detection model is to recognize math equations in images with high efficiency and accuracy in order to be usable for Math IR and other real-world systems. In developing such a model, we focus on extracting math equations from images of printed scientific documents, as this is one of the most commonly-existing and widely-used mediums for math equation information stored on the Web. To detect equations in images, a Fully Convolutional One-Stage (FCOS) object detection model is adapted for identifying different math formulas in images, creating labels, and performing bounding box regression.

Most of the existing math equation detection approaches deliberately combine different methods together, and almost all of them rely on using CNNs for the task of equation detection. Our math equation detection model, however, implements FCOS, which is a one-shot object detection that utilizes a Resnet-based back bone, essentially a CNN with skip connections, combined with a Feature Pyramid Network to perform anchor free object detection [11]. All of these parts enable the network to perform with a good balance of speed and accuracy, which make FCOS ideal to use for Math IR systems. There is no current method which implements FCOS to detect math equations, and our model for detecting equations in images provides a viable method for equation detection which achieves high precise results and efficient performance.

Our math equation detection model utilizes a FCOS frame-work to identify different kinds of equations and form bounding boxes around them. The model takes images of printed documents, locates math equations within them, and forms bounding boxes around them while identifying them as embedded formulas, which are surrounded by text or isolated equations that are separate from the rest of the text. These equations might be split across lines or pages. While this approach has not been used for this kind of problem before, a related method called Faster R-CNN has been used for extracting math equation from images previously, so there is a precedent for using ML as object detectors for Math IR systems. FCOS, in particular, is both faster than most detection methods while also being lightweight, relatively new with iterated additions that provide improvements to performance, and having high accuracy. While FCOS method does not translate math equations in images into some form of markup language, it is capable of locating them and identifying whether an equation is separated from the rest of the text or split across lines. Doing so makes it easier for those equations to be extracted later and used by math equation recognition methods. Using FCOS to extract equations of different types from images achieves high accuracy and speed, which are our design goals. It also works as a reliable method as part of the overall Math IR model that can operate in real time.

1) The architecture of FCOS

In terms of function, FCOS is an anchor-free object detector which solved object detection problems in a per-pixel prediction fashion, similar to segmentation. It is primarily based off of Fully Convolutional Networks (FCN) for semantic segmentation. The model architecture has three sections as depicted in Fig. 1, the backbone, feature pyramid, and head [4]. Feature maps extracted by the backbone are fed into the Feature Pyramid Network (FPN) at different levels of scale, and the different layers feed into each other from smallest to largest [11]. This enables robustness to scale variance and also allows choosing plausible object locations at a smaller scale before narrowing down on locations on a larger scale, which is efficient. The FCOS model is using ResNet50, a Convolutional Neural Network (CNN) utilizing residual layers for the feature extraction backbone. Resnet is a kind of DNN architecture which contains skip connections that link back from later layers to earlier ones, which enables gradients to flow through them, which is helpful, since it prevents vanishing or exploding gradients that could cause the network to fail. The output of the FPN then becomes the input to a head network. The head network has two main branches, one being used for classification to predict class confidence and center-ness of the bounding, and the other for regression to predict bounding boxes [4]. The input is encoded as an image, as well as associated classes for math equations and bounding boxes within those images, while the output of training is the losses and the output of inference is predicted math equation class types and bounding for the images passed through. There are three loss functions used for the head: classification loss uses focal loss, center-ness loss uses Binary Cross-Entropy Error (BCE) loss, and regression loss uses IoU loss.

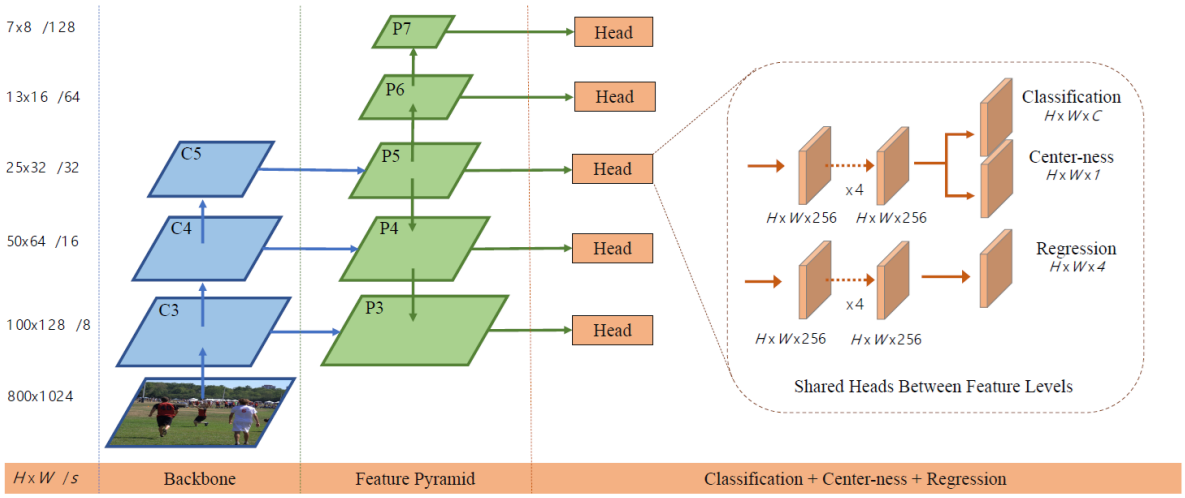


Fig. 1. The network architecture of FCOS, where C3, C4, and C5 denote the feature maps of the backbone network and P3 to P7 are the feature levels used for the final prediction. $H \times W$ is the height and width of feature maps. "/s" (s = 8, 16, ..., 128) is the downsampling ratio of the feature maps at the level to the input image.

2) An enhancement of detecting math equations

To ensure high efficiency of the proposed math equation detection model, Fast And Diverse (FAD) with Representation Sharing (Rep-Share) is applied to enhance the performance of FCOS. FCOS is equipped up a backbone network with FPN and two parallel subnetworks for object classification and bounding box regression [4]. The subnetworks is replaced with a search-able module that are searched for by FAD, as demonstrated in Fig. 2. The proposed searchable module FAD comprises of two groups of cells connected sequentially with a shortcut from the input of the module to that of the second group. The module outputs both object classification and bounding box prediction. The architectures and parameters are shared across different FPN levels [11]. RepShare is an acceleration method for architecture search which works by doing filter decomposition and intermediate representation sharing as shown in Fig. 3, which reduces the number of computations needed. Together, FAD and RepShare reduces memory consumption and computation time while still allowing for diverse transformations. The proposed model enables using the same parameters and computations in multiple places which accelerates searching and reduces memory consumption.

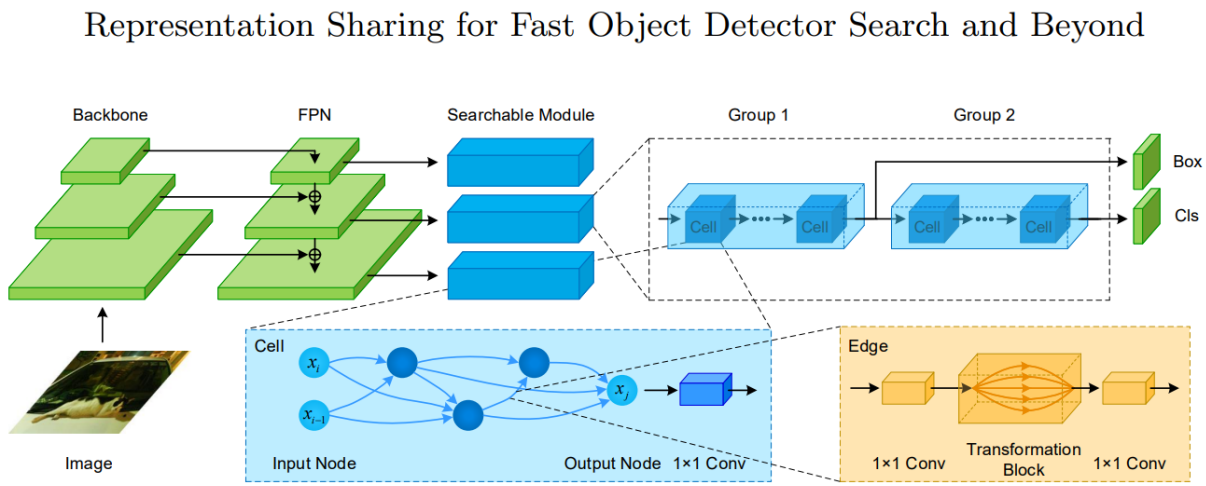


Fig. 2. Search space of FAD for one-stage object detectors. The backbone and FPN in detectors remain the same, while each FPN level is connected to a searchable module. It consists of two groups of cells, with same cell architectures within each group. In a cell, the edges connecting nodes consist of two standard 1×1 conv layers and a transformation block in between. The cell structures and the transformations are to be searched. Each edge might have different Random Fields (RFs), resulting in combinations of RFs at each node which enrich the features for capturing information of various scales.

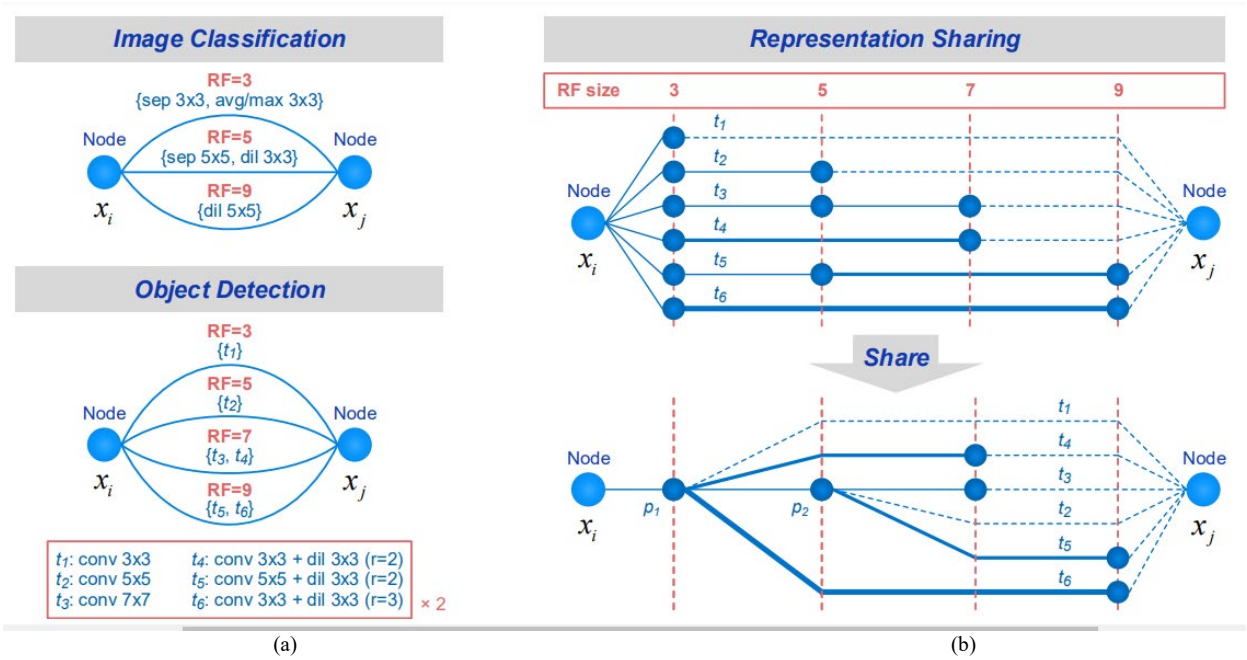


Fig. 3. Transformations and representation sharing. (a) Comparison between the transformations used for image classification and those proposed for object detection in the search space. The proposed transformations are listed at the bottom. Conv can be the standard or the depthwise separable convolution. (b) RepShare. Each sphere and solid line denotes a representation and a conv layer, respectively. First, large filters are decomposed into stacks of 3×3 filters. Second, p1 and p2 are shared across transformations. Note that the 1×1 conv layers are not shown for simplicity.

The optimizer, which is used for equation detection, is Adam with a set learning rate, since Adam is an effective optimizer that is commonly used and works well for a wide variety of problems. Precision, recall, and Mean Average Precision (mAP) serve as the error metrics. Frames Per Second (FPS) is then applied to measure the speed of the models. However, defining real time speed in a computer vision context is something which varies depending on the situation. Typically, having real time speed is defined as an algorithm processing input at the same rate of the source supplying the images.

Example 1: Consider the question, "Expressing Ramanujan τ function as Cauchy product of divisor function", extracted from Math Stack Exchange, a prominent online math forum, as shown in Fig. 4 in which all the embedded and isolated math equations are boxed and highlighted. Our detection model, which accurately detects all of the math equations embedded in the textual content, shows that it precisely extracts bounding boxes of math equations in a document.

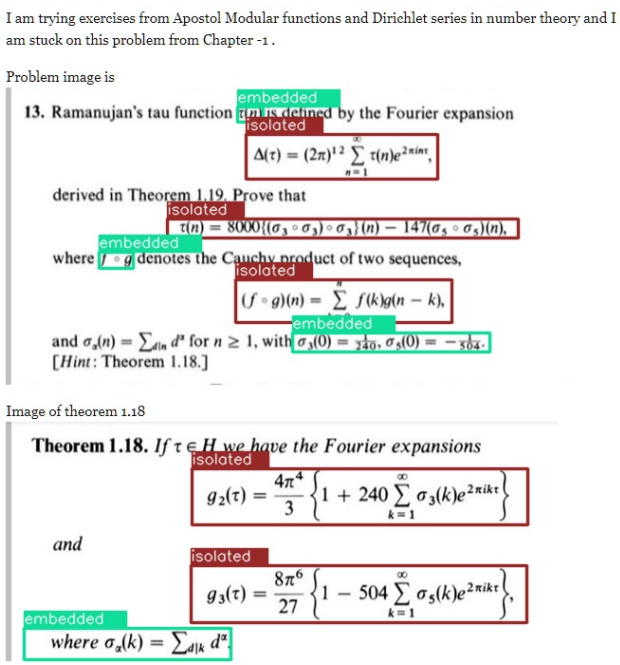


Fig. 4. A sample math question posted by a Math Stack Exchange user, with (a) embedded (inline with the textual content) in green bounding boxes and (b) isolated math equations in red bounding boxes, labeled and highlighted.

B. Our Math Equation Recognition Model

Our MIR system is capable of recognizing mathematical formulas in images and converting them into a usable text-based format with high efficiency and accuracy. In designing the math equation recognition model of our MIR system, we focus on extracting equations from images exposed by our detection model. To recognize and retrieve equations in the resultant bounding boxes as text, an encoder-decoder architecture, called Image2Latex, was utilized to convert math images into LaTeX markup language, which was designed for various applications [5, 12]. Along with the original method, a version of the Image2Latex model utilizing Bayesian Neural Networks (BNNs) was also implemented for comparison, since BNNs have been shown to decrease overfitting, better handle uncertainty, and more effectively handles smaller dataset sizes, which is helpful for the relatively small amount of data present for this problem. At present, since none of the existing image recognition models has incorporated BNNs into them, such a model comparison is a good indicator for how well BNNs work for image recognition and other computer vision tasks. The resultant model performs accurately while accounting for a smaller data set and uncertainty within the model.

Our model for math equation recognition enhances the Image2Latex Seq2Seq encoder-decoder model [12]. Image2LaTeX has been adapted to detect math equations and implements a Recurrence Neural Network (RNN) and soft attention mechanism to keep track of spatial information and symbol ordering to achieve more precise results. The model is capable of translating math formula images into LaTeX markup language, since LaTeX, which has been utilized to produce scientific papers, is fairly compact when it comes to representing math formulas, and already has existing open-source methods that can convert LaTeX to other markup languages, which is good to use for MIR systems. Since encoder-decoder models in general do a single pass through the model, and this particular method uses beam search in the decoder to find the optimal output sentence, the base Image2Latex model retains good speed and accuracy, which is beneficial for the use of MIR systems [12].

The encoder uses a CNN network to extract features from the images and encodes them with spatial information using a row encoder while doing batch normalization so that the network runs faster with more stability [12]. The decoder is a RNN which is composed of stacked Bidirectional Long Short-Term Memory (BiLSTM) blocks integrated with a soft attention mechanism [5]. Both the encoder and decoder architectures are shown in Fig. 5. Along with this model a version of the encoder-decoder model was produced which incorporates BNNs into the architecture, specifically by replacing the BiLSTM blocks in the RNNs for the row encoding with Bayesian LSTMs. The recognition model operates as a language model so that the feature and spatial information in the encoder output are translated into a LaTeX sequence. Since all parts of a math equation influence the meaning and arrangement of the entire equation, and math equations end up being rather large, having a mechanism to keep track of and compare features to each other, such as attention or recurrence, is needed for this particular problem. Most modern methods for recognizing math equations rely on some form of encoder-decoder architecture, since this particular framework is more effective for the purpose of converting images to text. Using a full-on transformer model can end up producing more accurate results, but transformers require more data to train properly and are

computationally costly, which is not good to use for Math IR. In comparison, generating predictions using the encoder-decoder architecture only necessitates passing the input through the network once, which is significantly faster than a full-on transformer architecture.

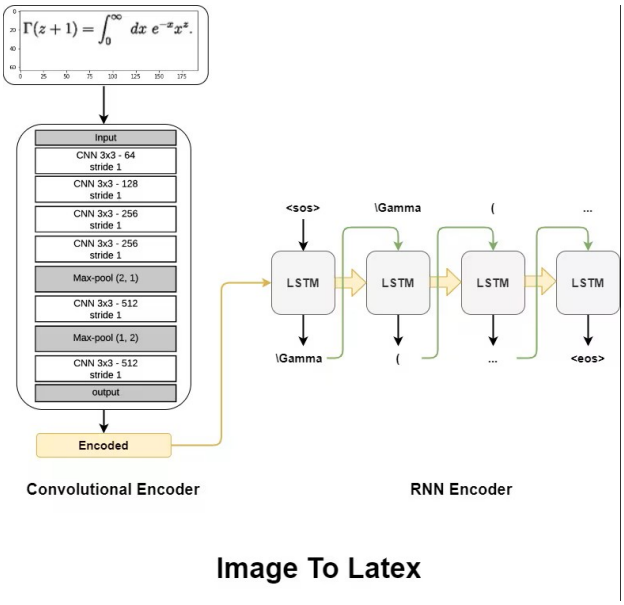


Fig. 5. The network architecture of Image2Latex, which is composed of a convolutional encoder and a RNN decoder.

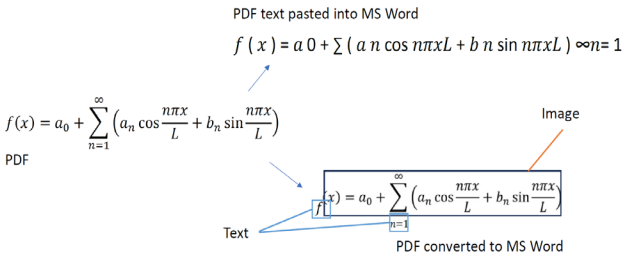


Fig. 6. The math equation shown was originally located in a PDF document (as shown on the left of the image) and then was (a) pasted into a Microsoft Word document (as shown at the top of the image), and (b) the PDF was converted into a Microsoft Word document (as shown at the bottom of the image). In contrast with the original PDF medium, in either adaptation information was lost or distorted, with (a) the equation on top all being inline without spatial information and (b) the equation on the bottom having part of the text as an image, with the rest as text without full alignment with the image.

Our equation recognizer model focuses specifically on images of printed documents, a typical storage form. As shown in Fig. 6, when math equations stored in PDFs are copied and pasted in Microsoft Word documents, the math equation ends up being converted to a one-dimensional line of text comprised of the symbols in the equation that is not always in the correct order. Any subscripts or superscripts are lost, which changes the meaning of the converted equation to be different from the original. For converting PDFs into Microsoft Word documents directly, the equation can end up being converted to an image within the file with some of the math symbols left around the edge of the image as text (see Fig. 6). Neither of these results is fully comprehensible by human or computer standards. As such, in the case of math equations that are stored in a textual document format that has difficulty being converted to different mediums, e.g., PDFs, it can be more efficient to render them as an image, such as snipping tool, screenshot, photo, etc., and then convert those images to a math markup language such as LaTeX using our MER model.

IV. Experimental Results

We analyze the performance of the proposed detection and recognition models. The codes of the models and datasets used for evaluation of the proposed model, can be found in https://github.com/Zenos5/PMEDR/tree/main/image2latex

A. The Datasets

Math equation recognition models normally require having images of equations with associated text passed in as the input. To verify the performance of our detection and recognition models, we used the im2latex series of datasets, which contain numerous math equation images with equivalent markup language transcripts. (See a sample equation image and its transcript as shown in Fig. 7.) This series of datasets contains over 100,000 math equation images and LaTeX transcripts per dataset, extracted from open-source documents in the 2003 KDD cup. As these datasets have specifically been created with image to LaTeX models in mind and are commonly used for math equation detection and recognition, they were used for training our FCOS and encoder-decoder models for equation detection and recognition, specifically for the im2latex-100K and im2latex-230K datasets. The former contains approximately 100K, whereas the latter includes close to 230K images and LaTeX transcripts respectively. These datasets are arranged so that there is a single image repository, a JSON file which contains the vocabulary used for the LaTeX formulas and separates CSV files for partitions to use for training, testing, and validation. The two datasets were chosen, since the base Image2Latex model has already been setup to use the 100K dataset and is widely used for math equation recognition training, and the 230K dataset is the most recent and largest dataset in the im2latex series, which provides more data for our equation recognition model to learn from.

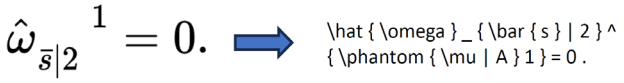


Fig. 7. A sample from the im2latex-100k dataset with a math equation image extracted with a bounding box and its corresponding LaTeX transcript.

While these are decent sized corpuses of data to work with for printed math equations, they are still a relatively small dataset in general. There is also a lack of variation in the data transcribed, since all of the images included are taken from scientific documents, which have standardized formats, sizes, spacing, and coloration. Without the inclusion of more varied data, model knowledge would not be sufficient for the kind of input distribution found in real world tasks. These datasets are

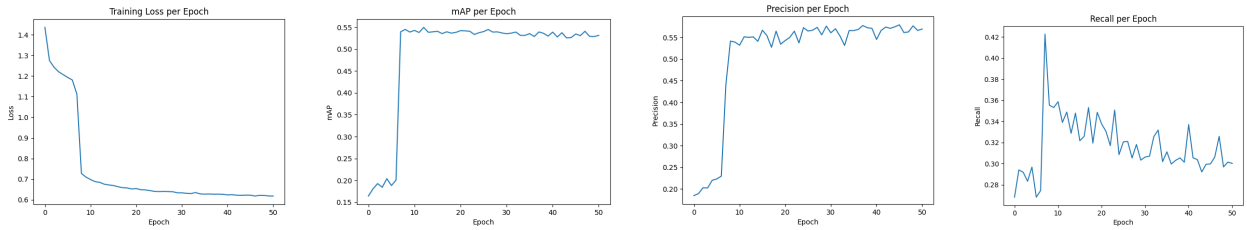
also specifically for printed and digitized documents, and do not include handwritten examples. The model is not trained for handwritten math equation detection and recognition and would need to be trained further on handwritten data for that particular task.

B. Evaluation on Our Detection Model

To verify the novelty of the proposed math equation detection model, we conducted a performance evaluation on our detection approach based on different quantitative measures and its processing speed.

1) Precision, recall, mAP, and training scores

For the detection model, training for 50 epochs was sufficient to obtain viable results and observe trends with the data. As depicted in Fig. 8, while the training loss continuously decreased, with greater change in loss at the start, the mAP values started out low, then spiked up and plateaued almost immediately with a slight downward trend, with a similar outcome with the mean interpolated precision and mean interpolated recall. The main spike occurs at epoch 10, with the training loss, mAP and mean interpolated precision quickly leveling out, whereas the mean interpolated recall starting to decrease.



(a) (b) (c) (d)

Fig. 8. The (a) training loss, (b) mAP, (c) mean interpolated precision, and (d) mean interpolated recall scores over the course of 50 epochs. A lower training loss indicates a better result, while a higher mAP, mean interpolated precision, and mean interpolated recall indicates a better score.

Various accuracy measures determine how often a model predicts the outcome correctly relative to the total number of predictions. Precision is the fraction of relevant instances among the retrieved instances, whereas recall is the portion of relevant instances that were retrieved. Since precision and recall are linked, raising one usually lowers the other. mAP is the mean of the average precision that measures the area under the precision recall curve and is influenced by both precision and recall.

As shown in Fig. 8 with the validation metrics when training, the precision values is higher over time while the recall values shrink, and is reflected in the mAP graph. The model is training to reduce error and increase accuracy, so it makes sense that the precision values get higher over time, since the model would have learned during training to return more accurate results. The fact that the recall value decreases once it reaches a certain point is due to the fact that the longer the model trains the more it tends to select more specific predictions, which would result in more precise predictions but miss viable predictions which the model is not as confident in.

As shown in Table I, the interpolated precision and recall at 1, 5, and 10 the precision is over 0.85, which indicates a precision of over 85% with the highest precision being 99% with 10 predictions, while recall is around 60% to 70% for the first 10 predictions. The reason why the recall is not higher is because there is a large number of correct math equations in the corpus that can be retrieved. The high precision and mAP indicate that our detection model performs very well for math equation detection, as it reliably returns accurate predictions over multiple images.

TABLE I. INTERPOLATED PRECISION AND RECALL FOR OUR DETECTION MODEL FOR THE FIRST RETRIEVED EQUATION PREDICTION, THE FIRST FIVE PREDICTIONS, AND THE FIRST 10 PREDICTIONS

Measures	@1	@5	@10
Precision	0.8562	0.9739	0.9932
Recall	0.5871	0.6372	0.6688

2) Processing speed

In terms of processing speed, our math equation detection model runs at approximately 16.930 FPS, although applying the model for extracting the bounding boxes to be used for the equation recognition model alongside error metric calculations runs at approximately 6.883 FPS during the prediction phase. As mentioned earlier, defining real time speed in a computer vision context is something which varies depending on the situation, and for doing text detection, when training on the ICDAR 2015 dataset an FPS rate of 8.9 was considered better than most state-of-the-art results, with 13.2 being the highest FPS a model achieved [10]. The images in the ICDAR 2015 dataset are 720 pixels wide and 1280 pixels high; however, in comparison, images in the IBEM dataset are 1447 pixels wide and 2048 pixels high, 3.22 times the size of images in the ICDAR 2015 dataset [13]. While natural language text detection is not the same as math equation detection, the tasks are similar enough to be used as a feasible target and benchmark in terms of speed, as most math equation detection methods do not list what speed the models ran at. Realistically, math equation detection is more complicated than natural language text, so the state-of-the-art speeds are likely significantly slower than general text detection in any event. Assuming that the speed of processing images is proportional to the image size, an equivalent real-world speed using math equation detection on the IBEM dataset

would be approximately 2.76 FPS, with an FPS of 4.10 being considered state-of-the-art. FCOS with Image2LaTeX runs significantly faster than these speeds, at least 2.0 FPS faster of the state-of-the-art speed even in the slowest case. As such, our MEDR model meets and exceeds the goal of operating at real-time speeds in detecting math equations in visual images, at least in comparison to general text detection.

C. Evaluation on Our Recognition Model

To verify the merit of the proposed math equation recognition model, we conducted a performance evaluation on the model. After completing training on the im2latex-100K dataset using 43 epochs, the proposed model performed as well as anticipated. Interestingly, there was little to no difference in terms of precision and accuracy in regards to training with and without using BNNs based on the im2latex-100K dataset. When using BNNs, training on the im2latex-230K dataset yields a slight decrease in accuracy and precision, though only around a 0.02 difference at most for the different metrics. For both datasets, using BNNs improved the speed of the model by approximately 6–10%, which is a significant speedup. More importantly, the semantic meaning of the formulas was very close or matched exactly. As shown in Fig. 9, all of the generated formulas were near identical except one missing/mismatched symbol and accenting (due to symbol similarity or lack of exposure during training). In a few cases, extra repeating patterns were added to the end of a math equation, which is a common problem with math equation recognition, text recognition, and LLMs in general, as text generators have a tendency to repeat sequences of words/symbols, especially if the pattern already showed up multiple times or if the model loses track of its place in the sequence. We have prevented excess repetition using the repetition dropout and with synthetic data technique to prevent formulas with repeating patterns to improve the overall effectiveness.

While training the math equation recognition model, Genthial *et al*. [12] and Loshchilov *et al*. [14] adopt Adam with Weight decay (AdamW) as the optimizer. For training, our recognition model adopted the same idea and was configured to stop training once the validation loss stopped decreasing. For evaluation, Bilingual Evaluation Understudy (BLEU), edit distance, loss, and exact match for math equations were used to measure the effectiveness of the encoder-decoder model. As for efficiency, having real-time speed is desirable. For camera and video processing, this is usually around 30 FPS. For math equation recognition, Anand *et al*. [15] claim that on the ICDAR 2013 dataset[1] [16] achieving a speed of 20 FPS is considered state-of-the-art achievement, with the next best state-of-the-art result topping out at 5.66 FPS. As such, an FPS of 5.66 can be used as a benchmark for a real time speed with a math equation recognition model. Specifically, the 12 FPS prediction speed of our recognition model is fast enough to run on real-world systems rather than the training speed, as training can be done offline, but prediction is done in sync with user input.

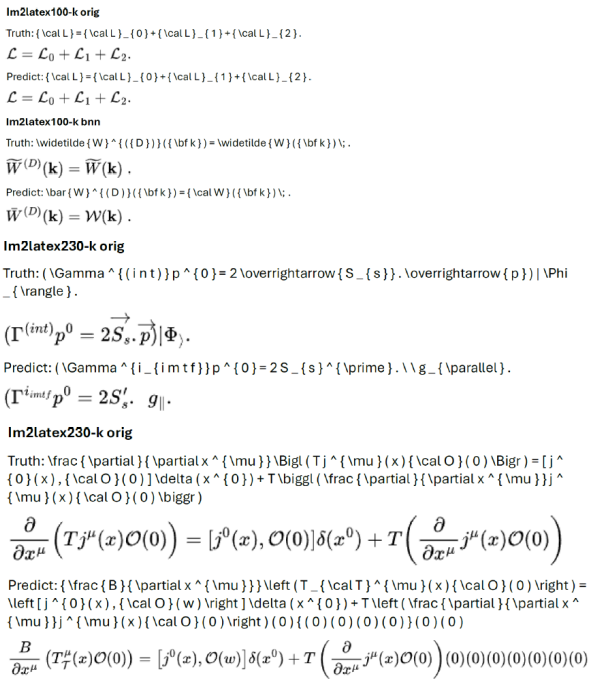


Fig. 9. Predicted math and LaTeX equations/images for im2latex-100K/230k with(out) BNNs, including (i) an exact match, (ii) mostly the same, with some missing or miswritten symbols, and (iii) mostly the same except for repeated symbols.

In terms of accuracy of the equation prediction, Table II shows the results on the 100K and 230K im2latex datasets with both the original model and with BNNs incorporated into the model. While not perfect, the results are a lot better than what might be suggested at face value. BLEU is intended to convey how close a computer-generated text is to the human-translated reference text, and the closer BLEU is to 1.0, the better the translation is [17]. However, it is near impossible to achieve a score of 1.0 in reality, and in general a value greater than 0.3 is considered a good score [17]. All of the experiments ran on the recognition model result in BLEU scores of over 0.63, which is exceptional in comparison.

TABLE II. RESULTS OF OUR RECOGNITION MODEL USING THE IM2LATEX-100K AND IM2LATEX-230K DATASETS WITH BOTH THE ORIG(INAL) MODEL AND WITH BNNS INCORPORATED INTO THE MODEL

Dataset	BLEU	Edit Distance	Loss	Exact Match	FPS
100K Orig	0.642	0.291	0.121	0.651	11.002
100K BNN	0.642	0.291	0.121	0.651	12.148
230K Orig	0.646	0.333	0.053	0.441	10.677
230K BNN	0.629	0.328	0.037	0.453	11.365

The edit distance conveys how many edits need to be made on average in order to convert from the predicted results to the actual math equations. The fact that only 0.333 edits need to be made at most, with 0.291 edits

[1]The ICDAR 2013 dataset consists of 229 training images and 233 testing images, with word-level annotations provided. It is the standard benchmark dataset for evaluating near-horizontal text detection.

being the lowest average edit distance score, indicates that very few corrections to the predictions were needed, which demonstrates that the predictions of math equations are usually quite accurate, or at least semantically and syntactically similar. As for the loss, our recognition model uses standard cross entropy loss which was calculated with predicted and true math equations, with lower loss indicating better results. The loss on the test sets ranged from 0.037 to 0.121, which is exceptionally good for a recognition problem and indicates that the model did not end up overfitting to the training data. Overall, our predictions were accurate and comparable with true math equation markups.

As for the exact matching, having a score of 1.0 would indicate that all of the predictions were identical to the ground truth math equations, and a score of 0 would indicate that none of the predictions matched the ground truth equations. Our trained math recognition model achieved 65.1% matched at the highest, with the lowest exact match percentage being 44.1%. This is a good score for a recognition model, since it is difficult to generate predictions which exactly match a math equation, as there are multiple ways to write the same equation and symbols which look really similar but are actually different. Only around 10% at most of the generated math equations completely matched the target math equations [17].

Our math equation recognition encoder-decoder model meets the design goal for converting equation images to LaTeX, having high processing speed and accuracy. It turns out that incorporating BNNs into the Image2latex model has not shown significant improvement in accuracy, which is not a surprise, since our recognizer stops with validation checking before it reaches the point where it overfits to the dataset. However, the processing speed of our recognition model based on *FPS* is improved using BNNs.

V. Conclusion

Being able to reliably extract math equations from images would allow more specification to be given in searches for math source and provide greater access to relevant math information to be used by both math learners and experts alike. Since numerous sources of math information are either in electronic files, physical documents, or are contained in photos or digital images, if we are unable to convert these sources into a text-based format, conventional Math IR systems would fail to utilize the embedded information. The side effect is that many viable sources of math information would be left out from the users. In addition, the process of transcribing math equations in images to text manually is a tedious and time-consuming. As such, being able to solve these two major problems with an automated tool would give users access to more information to learn from and provide more information for Math IR systems to utilize quickly with less effort to help users search relevant information.

We have proposed a math equation detection and recognition model, denoted MEDR. MEDR composes of a one-shot object detector enhanced with the use of architecture search and representation sharing and a Seq2Seq BNN encoder-decoder method [18] with repetition mitigation and soft attention in order to detect and recognize math equations contained in digital images. MEDR adopts FCOS, FAD, and BNNs for the given tasks on math equation detection and recognition. An empirical study on MEDR using the document image im2latex datasets has demonstrated that the proposed model has viably achieved promising results in terms of processing speed and accuracy. The significant contribution of MEDR is its ability to work with a variety of images and reliably and swiftly return the associated LaTeX markup transcripts which existing Math IR systems are keen to include in their systems.

For future work, we consider adding data augmentation during training to enable MEDR to generalize to a wider range of data, which would be beneficial for the different kinds of images appeared in Math IR models, especially Question-Answer (QA) systems. Specifically, rotations, resolution, scale, text fonts, and color tones are good transformations to be included in training to account for variation in the input images. Compiling a larger dataset with both printed and handwritten mathematical texts would improve coverage of varied input data and increase the utility and accuracy of the MEDR model. Doing computational complexity analysis, training and comparison with other MIR models on equivalent data and running a case study would give additional insights into the performance of MEDR from a complexity, equivalency and user perspective. To further improve the efficiency of the detector of MEDR, we would like to incorporate various approaches with the base FCOS model to accelerate searching and reduce memory consumption [19]. As for the recognition model, incorporating changes to prevent excessive repetition using repetition dropout or creating synthetic data to use to penalize repetitions in the model should help prevent formula predictions with continuously repeating patterns [20]. These modifications allow for reaching faster speeds, using less computations, having better generalization, and attaining better precision and accuracy.

Appendix A: Hardware Specification

Fig. A1 depicts the hardware specifications that include the GPU model and CUDA version, which is produced using the “nvidia-smi” command.

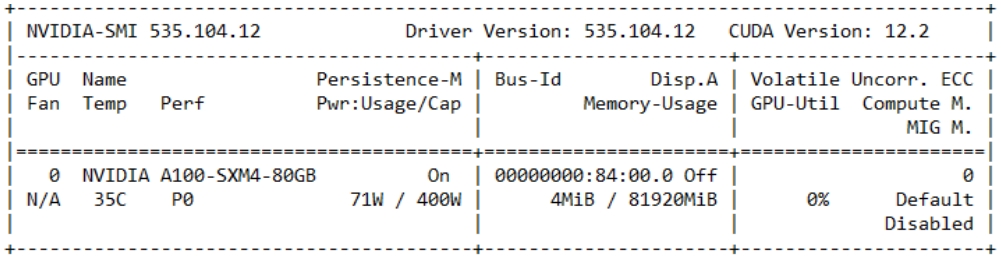


Fig. A1. The hardware specifications that include the GPU model and CUDA version.

Appendix B: Dataset Location and License

A. *im2latex-100k*

The im2latex-100K dataset is located at https://doi.org/10.5281/zenodo.56198

License: CC0 1.0 Universal

B. *im2latex-230k*

The im2latex-230K dataset is located at https://doi.org/10.5281/zenodo.7738969

License: Creative Commons Attribution 4.0 International

C. *IBEM*

The IBEM dataset is located at https://doi.org/10.5281/zenodo.4757865

License: Creative Commons Attribution 4.0 International

Conflict of Interest

The authors declare no conflict of interest.

Author Contributions

Angel Wheelwright developed the methodology of the paper, wrote each section included, and conducted the empirical study of the project; Yiu-Kai Ng involved with the design of the proposed detection and recognition model and reviewed and edited the submitted manuscript; all authors had approved the final version of the paper.

Acknowledgment

The authors wish to thank the anonymous reviewers of the submitted manuscript with constructive comments that enabled us to further enhance its content, structure, and presentation.

References

[1] National Center for Educational Statistics. The NAEP Mathematics Achievement Levels by Grade. [Online]. Available: https://nces.ed.gov/nationsreportcard/mathematics/achieve.aspx

[2] Public School Review. (2023). Average Public School Math Proficiency.[Online]. Available: publicschoolreview.com/average-math-profiency-stats/national-data

[3] M. Zhou, M. Cai, G. Li, and M. Li, "An end-to-end formula recognition method integrated attention mechanism," *Mathematics*, vol. 11, no. 1, 177, 2022.

[4] P. Chandra and S. Rath. (2022). FCOS-anchor free object detection explained. [Online]. Available: learnopencv.com/fcos-anchor-free-object-detection-explained/

[5] Z. Wang and J. Liu, "Translating math formula images to LaTeX sequences using deep neural networks with sequence-level training," *IJDAR*, vol. 24, no. 1, pp. 63–75, 2021. doi: 10.1007/s10032-021-00378-0

[6] K. Hashmi, A. Pagani, M. Liwicki, D. Stricker, and M. Afzal. "Cascade network with deformable composite backbone for formula detection in scanned document images," *Appl. Sci*., vol. 11, no. 16, 7610, 2021. doi: 10.3390/app11167610

[7] S. Madisetty, K. Maurya, A. Aizawa, and M. Desarkar, "A neural approach for detecting inline mathematical expressions from scientific documents," *Expert Syst*., vol. 38, no. 4, e12576, 2021. doi: 10.1111/exsy.12576

[8] B. Phong, T. Hoang, and T. Le, "An end-to-end framework for the detection of mathematical expressions in scientific document images," *Expert Syst*., vol. 39, no. 1, e12800, 2022. doi: 10.1111/exsy.12800

[9] W. Chu and F. Liu, "Mathematical formula detection in heterogeneous document images," in *Proc. TAAI*, pp. 140–145, IEEE, 2013.

[10] G. Tong, M. Dong, and Y. Song, "A real-time and effective text detection method for multi-scale and fuzzy text," *Real-Time Image Process*., vol. 20, no. 1, 13, 2023. doi: 10.1007/s11554-023-01364-y

[11] T. Lin, P. Dollar, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proc. IEEE CVPR*, 2017, pp. 2117–2125.

[12] G. Genthial and R. Sauvestre. (2016). Image to Latex. [Online]. Available: https://api.semanticscholar.org/CorpusID:229712014

[13] D. Karatzas, L. Gomez-Bigorda, A. Nicolaou, S. Ghosh, A. Bagdanov, M. Iwamura, J. Matas *et al*., "ICDAR 2015 competition on robust reading," in *Proc. ICDAR*, IEEE, 2015, pp. 1156–1160.

[14] I. Loshchilov and F. Hutter, "Decoupled weight decay regularization," in *Proc. ICLR*, 2017, pp. 1–8.

[15] S. Anand, S. Susan, S. Aggarwal, S. Aggarwal, and R. Singla, "Scene text recognition in the wild with motion deblurring using deep networks," in *Proc. CVIP*, Springer, 2021, pp. 93–103.

[16] D. Karatzas, F. Shafait, S. Uchida, M. Iwamura, L. Bigorda, S. Mestre, J. Mas *et al.* (2013). ICDAR 2013 Dataset. [Online]. Available: paperswithcode.com/dataset/icdar-2013

[17] R. Ashraf. (2023). Demystifying the BLEU metric: A comprehensive guide to machine translation evaluation. [Online]. Available: https://www.traceloop.com/blog/demystifying-the-bleu-metric

[18] M. Song, Z. Chen, P. Niu, and E. Haihong, "FPSeq: Simplifying and accelerating task-oriented dialogue systems via fully parallel sequence-to-sequence framework, in *Proc. ICTAI*, 2019, pp. 1195–1202.

[19] Y. Zhong, Z. Deng, S. Guo, M. Scott, and W. Huang, "Representation sharing for fast object detector search and beyond," in *Proc. ECCV*, Springer, 2020, pp. 471–487.

[20] H. Li, T. Lan, Z. Fu, D. Cai, L. Liu, N. Collier, T. Watanabe, and Y. Su, "Repetition in repetition out: Towards understanding neural text degeneration from the data perspective," *Adv. Neural Inf. Process. Syst.*, vol. 36, pp. 1–16, 2024.