Using FCOS and an Encoder-Decoder Model to Detect and Recognize Visual Mathematical Equations

Angel Wheelwright
Brigham Young University
3361 TMCB, Computer Science Department
Provo, Utah, USA
angelwhwrt@gmail.com

Yiu-Kai Ng
Brigham Young University
3361 TMCB, Computer Science Department
Provo, Utah, USA
ng@compsci.byu.edu

ABSTRACT

Many data sources on the Internet contain math information within them, and math is used throughout daily life while being important for avenues of study and industry. Understanding math enables better problem solving, pattern comprehension, quantifying relationships, and making predictions of the future. Unfortunately, less people have proficiency in math in recent times. To make the situation worse, it is difficult to locate sources of relevant math information, particularly when the searcher has little familiarity with the subject area. Having Math Information Retrieval (IR) systems would help facilitate searches for math information and assist learners with understanding math concepts. Sadly, extracting mathematical notation in graphical representations into a standardized text-based format is a non-trivial task, since it is required to detect unique symbols and spatial arrangements of mathematical characters, as well as formula positioning in documents. Failure to correctly detecting and recognizing visual math formulas and their notation produce errors that alter the entire meaning of the resulting formulas, or simply do not have the speed needed for a real-time Math IR system. To address these problems, we have developed a combined FCOS and Image2Latex framework to detect and extract math formulas from images and translate them accurately into LaTeX in a reasonable time frame.

CCS CONCEPTS

•Computing methodologies →Artificial intelligence; •Computer Vision → *Object detection; recognition;*

KEYWORDS

Image detection, image recognition, visual math formulas, Math IR

ACM Reference format:
Angel Wheelwright and Yiu-Kai Ng. 2024. Using FCOS and an Encoder-Decoder Model to Detect and Recognize Visual Mathematical Equations. In *Proceedings of The 9th International Conference on Multimedia and Image Processing, Osaka, Japan, April 20–22, 2024 (ICMIP 2024),* 9 pages.
DOI: XXXXXXX.XXXXXXX

ICMIP 2024, Osaka, Japan

DOI: XXXXXXX.XXXXXXX

1 INTRODUCTION

Math is used in many parts of daily life, such as scheduling, cooking, finances, measurements, and organization, and many fields of study as well as industry utilize it. Indeed, math is considered a universal language because it conveys quantitative properties and values as well as how processes work, and exposure to math enables a greater capacity for problem solving, pattern comprehension, quantifying relationships, and making predictions of the future. There are many sources of information which contain math information, both offline and on the Internet. Wikipedia, Math Stack Exchange, scientific documents, textbooks and manuals all contain math interspersed with the rest of the text. In many cases, math information is stored in a textual format and typically in some form of math markup language, such as LaTeX, MathML, OpenMath, or OMDoc. Another common medium that math information is stored in are images or scanned documents, such as photos or PDFs.

Sad to say, math proficiency levels for people around the world, particularly in the United States, have dropped in recent years. In the United States, the national average math proficiency in public schools is 38% from 2023-24 [28]. According to the National Assessment of Academic Progress, 12^{th} graders in the US are considered to be proficient in math if they have a score of 176 or higher, but the grade average in 2019 was 150 [22, 23]. To make matters worse, it is difficult for people to locate viable sources of math information to either learn how to do math or familiarize themselves with an area of study or research which involves math, especially if they are not familiar with the subject. Such people would benefit from having systems in place which facilitate searching and translation of sources of math information. The area of study in Math information retrieval (IR) appears to be the answer to the problem.

Math Information Retrieval (IR) systems are a relatively new field of study, in which the systems involved organize, store, retrieve, and evaluate math information from document repositories. These systems are useful for ordinary users, as well as experts, in math or related fields to find relevant information and aid them in increasing their understanding of math concepts. In order to build a robust Math IR system, designers are expected to generate meaningful rankings on math information and recommend documents retrieved for math queries. Being able to extract math notation from images would be beneficial, since many sources of math information are in a visual storage medium, whether that be a PDF or image, or in a physical document, which would benefit from being able to be converted from a physical to digital format to be more accessible to Math IR systems and people online [24]. Converting math equation (ME) images into textual format, however,

is not a simple task. This is because natural language text is arranged in relatively easy to parse lines, with symbols aligned in a single dimension, while math notation can appear both inline with surround text or isolated from the rest of a document or image. To further complicate matters, the same ME symbol can be used for multiple purposes. For instance, the '$\cdot$' in "$a \cdot b$" could be referring to algebraic multiplication, matrix multiplication, or concatenation, and $f \circ g$ can refer to the Hadamard product with matrix multiplication or for function composition. Subscripts and superscripts alter the sizes and location of the symbols involved, and ME symbols such as this are arranged in two spatial dimensions, i.e., horizontal and vertical, rather than one dimension. All of these factors are critical for accurately extracting a ME with the intended meaning. Even when math formulas are correctly extracted from images, most existing methods for ME extraction designed with accuracy in mind rather than speed, with ScanSSD-XYc being the only method that addresses speed at all when it comes to ME extraction systems [7]. This is problematic because Math IR systems require both reasonable accuracy and real time speeds to be useful in real world contexts.

To address the problem mentioned above, we aim to create a model that is capable of detecting and recognizing math formulas in images and converting them into a usable text-based format with high efficiency and accuracy to be usable for any Math IR system [8, 34]. For the proposed model, we specifically focus on extracting MEs from images of printed scientific documents, as this is one of the most commonly existing and used mediums for ME information stored on the Internet. To *detect* MEs in images, a fully convolutional one-stage object detection (FCOS) model [29] is adapted for identifying math formulas in images, creating labels, and performing bounding box regression [30]. To *recognize* and return MEs in the resulting bounding boxes as text, an encoder-decoder architecture, called *Image2Latex*, is utilized to convert the images into LaTeX markup language for use in other applications, since LaTeX is a commonly used markup language that is used 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 Math IR systems [6, 26, 33].

2 RELATED WORK

While math expression detection and recognition (MEDR) are a relatively new field of study, there has been related work in these fields going back over a decade, and focuses mostly on extraction from PDFs and images, and on extracting printed or handwritten MEs [16]. While there have been methods created to process ME documents with non-machine learning (ML) techniques, ML is 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, 11, 18].

Previous work which relates to ME 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, the ScanSSD and ScanSSD-Xyc slide windows over images using a CNN to select ME bounding boxes [7, 20]. Another method uses segmentation to separate text from the rest of the document, and then applies a SVM to determine if the segmented line is a ME, and the authors of [5] classify MEs as inline or isolated. Moreover, Phong, et al. [27] segment the text before running a CNN for feature extraction, while Ohyama et al. [25] and Madisetty et al. [19] pair a U-Net with a CNN and a Conditional Random Field (CRF) with a Recurrent Neural Network (RNN), respectively to detect the region of a formula. Almost all of these approaches rely on using CNNs for the task of ME detection. The detection model adapted by us, however, is Fully Convolutional One-stage Object Detection (FCOS), which is a one-shot object detection that utilizes a Resnet-based back bone combined with a Feature Pyramid Network to detect anchor free objects. All of these parts enable the network to perform with a good balance of speed and accuracy, which would make FCOS ideal to use for Math IR systems [35]. At present, there is no current method which implements FCOS to detect math formulas. Adapting this model, we provide a viable method for ME detection that offers both precise results and efficient performance.

As for ME recognition, existing approaches typically use conventional OCR with SVMs for classification [21]. Most methods that were introduced later utilize some form of encoder-decoder architecture, usually with a LSTM variant or attention included [32, 33]. Some of the more recent methods use visual transformers (ViT) as well [36]. The persistent use of encoder-decoder architectures for this problem indicates that this approach is effective for this problem, and is one of the currently existing methods that can be configured to convert images to text, something which is needed for ME 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 context for recognizing ME text in images [16].

Our model for ME recognition is based on the *Image2Latex* encoder-decoder model [10]. We have implemented a recurrence neural network and soft attention mechanism to recognize spatial information and ME symbol ordering to achieve precise results using math equations detected by FCOS. Since encoder-decoder models in general going through a single pass through the model, and this particular method uses beam search in the decoder to find the optimal output sentence, the Image2Latex model retains good speed as well as accuracy, which is beneficial to Math IR systems.

3 THE PROPOSED MATH EQUATION DETECTOR AND RECOGNIZER

In this section, we present the design of our math equation detector and recognizer.

3.1 The FCOS Math Equation Detector

Our ME detection model utilizes a Fully Convolutional One-stage Object Detection (FCOS) framework to identify different kinds of MEs and form bounding boxes around them. This model take images of printed documents, locate MEs within them, and form bounding boxes around them while identifying them as embedded formulas, which are surrounded by text, or isolated formulas, i.e., formulas that are separated from the rest of the text, as well as if they are split across multiple lines or pages. While FCOS has not been used for detecting math equations before, a related method called Faster R-CNN has been used for extracting MEs from images

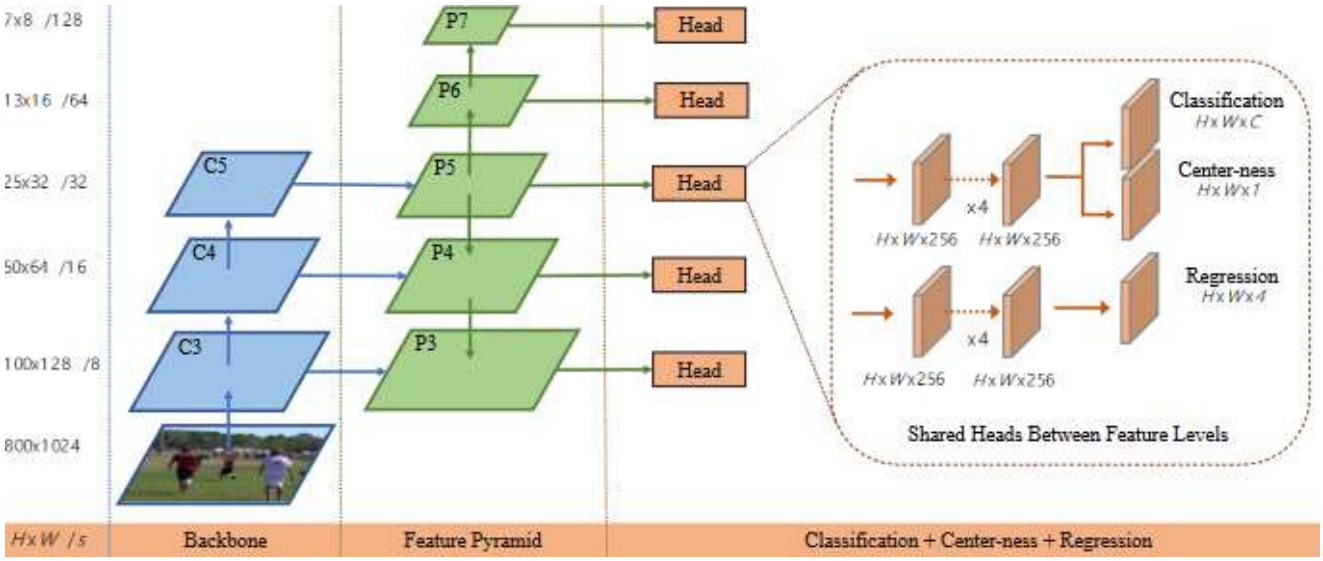


Figure 1: The network architecture of FCOS, where C3, C4, and C5 denote the feature maps of the backbone network and P3 to P7 are the feature levels used for the final prediction. $H \times W$ is the height and width of feature maps. '/s' (s = 8, 16, . . ., 128) is the downsampling ratio of the feature maps at the level to the input image [29]

previously, so there is a precedent for using it as object detectors for Math IR systems. FCOS in particular is both *faster* than most methods we have come across while also being *lightweight*, relatively new with iterated additions that provide improvements to performance, and having *high accuracy* [13]. While this method does not translate MEs in images into some form of markup language, it is capable of locating them and identifying whether the formula is separate from the rest of the text or split across lines. Doing so makes it easier for those formulas to be extracted later and used by ME recognition methods. The purpose of FCOS is to extract MEs of different types from images with high accuracy and speed so that it is a reliable approach to use as part of the overall Math IR model that can operate in real time.

In terms of function, FCOS is an anchor-free object detector which solves object detection problems in a per-pixel prediction fashion, similar to segmentation. This method is primarily based off of Fully Convolutional Networks (FCN) for semantic segmentation. The model architecture has three sections as shown in Figure 1, the backbone, feature pyramid, and head.

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 [17]. 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 more 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 which link back from later layers to earlier ones, which enables gradients to flow through them, something which is helpful, since it prevents vanishing or exploding gradients that could cause the network to fail [12]. The output of the FPN then go 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 MEs and bounding boxes within those images, while the output of training is the losses and the output of inference is predicted ME class types and bounding for the images passed through. There are three loss functions used for the head, namely classification loss uses focal loss, center-ness loss uses binary cross-entropy error (BCE) loss, and regression loss uses IoU loss.

3.2 The Image2Latex Math Equation Recognizer

In order to perform ME recognition, an existing encoder-decoder model, called *Image2Latex*, was implemented. This model is a Seq2Seq model which utilizes an encoder-decoder architecture with soft attention to translate math formula images into LaTeX markup language[1], as shown in Figure 2.

As shown in Figure 2, the encoder uses a CNN network that extracts features from the images and encodes them with spatial information, doing batch normalization so that the network runs faster with more stability. The decoder is a RNN which is composed of stacked bidirectional long short-term memory (BiLSTM) blocks integrated with a soft attention mechanism. This will operate as a language model so that the feature and spatial information in the encoder output will be translated into a LaTeX sequence. Since all parts of a ME influence the meaning and arrangement of the entire ME, and MEs can end up being rather large, a mechanism to keep track of and compare features to each other, such as attention or recurrence, is needed for this particular problem [1]. Making predictions using an encoder-decoder architecture only necessitates passing the input through the network once, so the ME recognition model will run fairly fast. Using a full-on transformer model, which is a more recent method, can produce more accurate results,

[1]Many of the more recent methods for doing ME recognition rely on some form of encoder-decoder architecture, since such a framework is effective for the purpose of converting images to text.

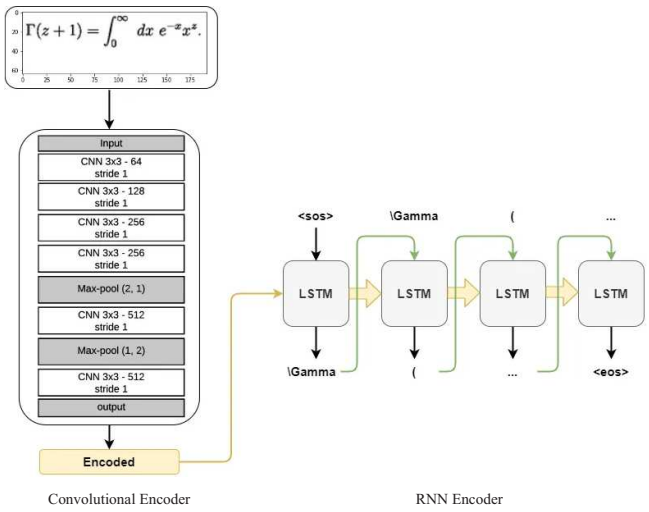


Figure 2: The network architecture of *Image2Latex*, which is composed of a convolutional encoder and RNN decoder

but transformers require more data to train properly and are computationally costly, which are not practical for Math IR [14].

4 SOURCES OF DATA USED FOR ME DETECTION AND RECOGNITION (MEDR)

For our MEDR model, the focus is specifically on images of printed documents, since most of the math information readily available is stored in some form of document, whether it be a scientific paper or a textbook or in Wikipedia. Being able to convert document images containing MEs into text helps extracting math information in documents. As shown in Figure 3, when MEs stored in PDFs are copied and pasted in MS Word documents, the ME ends up being converted to a one-dimensional line of text comprised of the symbols in the ME that is not in the *correct order*. Moreover, subscripts and superscripts are lost and the converted formula changes the meaning of the original ME as shown in Figure 3.

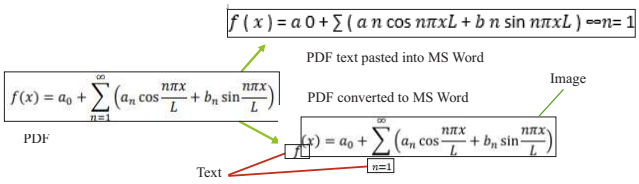
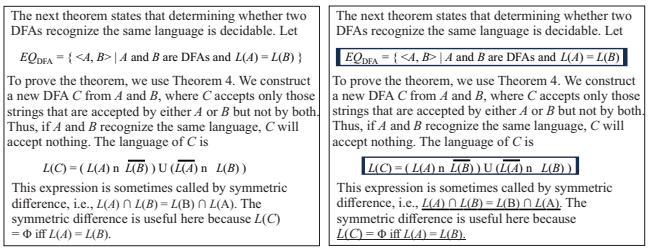


Figure 3: Math formula in a PDF document pasted in Microsoft Word and converted to Microsoft Word

For converting PDFs into Microsoft Word documents directly, the original ME ends 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 Figure 3 for an example). Neither of these results is fully comprehensible by human or computer standards. As such, in the case of MEs that are stored in a textual document format that has difficulty being converted to different mediums (e.g., PDFs), it may end up being more efficient to render them as an image (e.g., snipping tool, screenshot, photo, document conversion, etc.) and then convert those images to a math markup language, such as LaTeX, using our MEDR model.



(a) Original page (b) Highlighted bounding boxes

Figure 4: MEs in an IBEM page where inline (displayed, respectively) MEs are underlined (in boxes, respectively)

Since our MEDR model is expected to detect and recognize MEs within printed document images, the ***IBEM*** dataset was used for training the FCOS model for ME detection, as this dataset contains whole document images in scientific papers with bounding boxes and formulas defined, which more closely matches the kind of documents which are used as input into Math IR systems (see Figure 4 for a sample).

The IBEM dataset contains 160,000 formulas across 600 documents, with the labels containing bounding boxes, equivalent LaTeX formula text, as well as whether an ME is isolated, embedded, or split across lines or pages [3]. The IBEM dataset is one of the largest publicly available document image datasets available, and is composed of open-source papers from the 2003 KDD Cup [9]. All of the document page images are located in a single directory, the labels for those pages are stored in a JSON file, and there are already train, test, and validation sets defined as lists of page images to be used for these different partitions. While there are places where embedded formulas are split across lines or pages, isolated images do not have an instance where they are split. Based on this information, four classes were defined for ME detection on the IBEM dataset: *background, isolated, embedded*, and *embedded_split*. Before using this dataset, the partitions are modified to only include image references in the original partition list in which both the image and the associated annotations actually exist.

ME recognition models normally require having images of MEs with associated ME text passed in as the input, rather than a document image with MEs within them. The IBEM dataset can be adapted for this purpose, but there are already datasets that exist which contain numerous ME images and have equivalent markup language transcripts, particularly the ***im2latex*** series of datasets, which was created for use in OpenAI's image to LaTeX system [6]. The *im2latex* series of datasets is comprised of ME images and equivalent LaTeX markup transcripts (see Figure 5 for a sample).

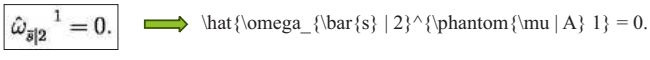


Figure 5: A sample from the im2latex-100k dataset with a ME image extracted with a bounding box and an equivalent LaTeX transcript

This series of datasets contains over 100,000 ME images and LaTeX transcripts per dataset, extracted from open-source documents in the 2003 KDD cup [9]. As these datasets have specifically been created with image to LaTeX models in mind, and are commonly used for ME recognition, these datasets were used for

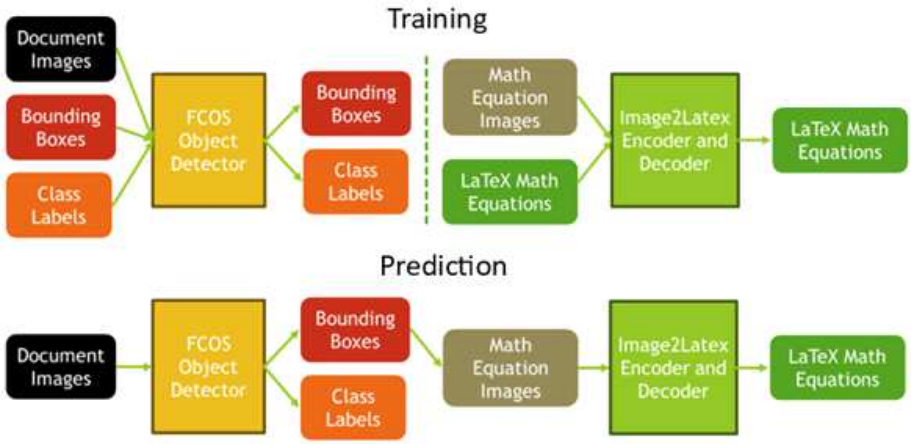


Figure 6: The inputs and outputs of the FCOS and Image2Latex models and how they feed into each other in the prediction phase, taking in document images as input and getting LaTeX MEs as output for the combined model. For training, document images, bounding boxes, and class labels from the IBEM dataset are passed through the FCOS model, while ME images and LaTeX transcripts are through the Image2Latex model

training the Image2Latex model for this problem, specifically the im2latex-100k and im2latex-230k datasets [6]. These datasets contain approximately 100k and 230k ME images and LaTeX transcripts respectively, and are arranged so that there is a single image repository, a JSON file which contains the vocabulary used for the LaTeX formulas, and separate CSV files for partitions to be used for training, testing, and validation. The reason we have chosen these two datasets is that the Image2Latex model is already setup to be able to use the 100k dataset that is commonly used for ME recognition training, and the 230k dataset is the most recent and largest dataset in the im2latex series, which provides more data for our MEDR model to learn from.

The target for training is 100 epochs for the ME detection and ME recognition models, but if a model starts overfitting or if the loss starts leveling out, then no more training is needed. The training and prediction process for these models is shown in Figure 6.

The optimizer which was used for the ME detection phase was *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 [15]. For the ME recognition phase Adam with weight decay (AdamW) is the optimizer used [10]. Training is done on a NVIDIA A100-SXM4-80GB GPU for both models.

5 PERFORMANCE MEASURES

For ME detection, *precision, recall*, and *mean average precision* (*mAP*) are used as the error metrics. *Bilingual evaluation understudy* (*BLEU*), *edit distance*, and *exact match* for both the textual math formulas and their resulting images are applied to measure recognition effectiveness [34]. While there are not any currently existing benchmarks for printed ME detection and recognition, particularly in terms of MEs rather than math symbols, the performance measures can be compared against some of the other models performing similar tasks, as many of them use the same or similar metrics with their models [27, 32]. *Frames per second* (*FPS*) is used to measure the speed of the models.

Defining real time speed in a computer vision context, however, 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. For camera and video processing, this is usually around 30 frames per second (FPS), but 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 [31]. The images in the ICDAR 2015 dataset are 720 pixels wide and 1280 pixels high. 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. Since most ME detection methods focus on increasing *accuracy*, there are little to no recorded efficiency measurements in terms of speed that can be found. However, while natural language text detection is not the same as ME detection, the tasks are similar enough to be used as a feasible target and benchmark in terms of *speed*. Assuming that the speed of processing images is proportional to the image size, an equivalent real world speed using ME detection on the IBEM dataset would be approximately 2.76 FPS, with an FPS of 4.10 being considered state-of-the-art. For ME recognition, on the ICDAR 2013 dataset having a speed of 20 FPS is considered state of the art, with the next best state of the art result topping out at 5.66 FPS [2]. As such, an FPS of 5.66 is used as a benchmark for a real time speed with a ME recognition model. It is the prediction speed which is used to assess whether the 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.

6 EXPERIMENTAL RESULTS

After training the FCOS and Image2Latex models, these models are capable of learning from the data and return viable results. As seen in Figure 7, the *mAP* values started out low, then spiked up and plateaued almost immediately, with a similar outcome with the *mean interpolated precision* and *mean interpolated recall*. In the same time frame, the *loss* started out larger with a high rate of change and then started to level out around epoch 30 with a *loss* of around 0.65, which demonstrates that the model was able to learn during the training phase.

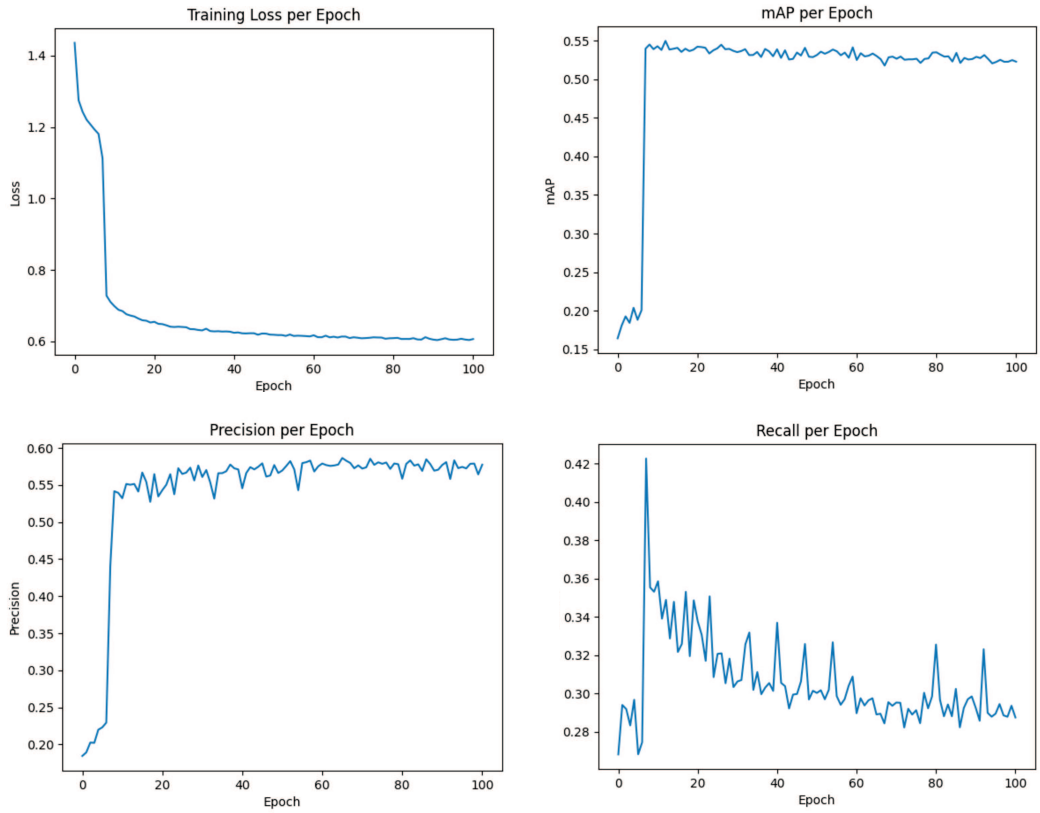


Figure 7: Graphs depicting the *training loss, mAP, mean interpolated precision*, and *mean interpolated recall* scores over 100 epochs

Higher *precision* means that an algorithm retrieves more results that are relevant than non-relevant ones, and high *recall* indicates that an algorithm extracts a higher proportion of the relevant results. However, there is a tradeoff between precision and recall, so raising one usually lowers the other. Since *mAP* computes the *average precision*, which measures the area under the precision-recall curve, it is influenced by both *precision* and *recall*. For the validation metrics when training, the precision values get *higher* over time, while the recall values *shrink*, which is reflected in the mAP graph depicted in Figure 7. It makes sense that the *precision* values get *higher* over time, since the model would have learned from the training set over time to return more *accurate* results. The fact that the *recall* gets *lower* may simply reflect the tradeoff between precision and recall, and specifically may be indicating that the *longer* the model trains the tendency is for the model to select *more* specific prediction, which would return *less* ME predictions overall, but those predictions would be *more precise*. Table 1 shows that at the interpolated precision and recall at 1, 5, and 10, respectively, the *precision* is over 0.9, which indicates greater than 90% *precision*, while *recall* is around 10% to 15% for the first 10 predictions. The reason that the recall is so *low* is that there is a *large* number of *correct* ME equations to retrieve and only a limited number are returned, but the *high precision* and the fact that the precision and recall are higher with more results returned indicate that this model performs *well* for ME detection, as it reliably returns *accurate* predictions over multiple images.

Table 1: Interpolated precision and recall for FCOS for the first retrieved ME prediction, the first five predictions, and the first 10 predictions

Measures	@1	@5	@10
Precision	0.9056	0.9894	0.9941
Recall	0.0920	0.1202	0.1678

6.1 Performance Using the im2latex-100k Dataset

Some observations can be made about the results as depicted in Figures 8a and 8b, which show some of the predictions made for ME bounding boxes and classifications on the IBEM dataset with the ME detection model. The predictions on the images in Figure 8a are all correct, whereas the image prediction in Figure 8b is mostly correct, but is missing one of the isolated images and misclassifies one of the *embedded_split* MEs as just an embedded ME. This pattern continues with the rest of the predictions generated by the model, with all or almost all of the bounding boxes and ME classification being correct with just a few missing or misclassified. Most of these misclassifications seem to occur with the *embedded_split* class, so more accurate classification could be achieved by taking

out that class altogether and adding those MEs to the *embedded* class. If split MEs do need to be identified, then their locations in the document images can be used as an indicator. Part of the reason that not all of the MEs are being returned may also be that the *threshold* used for counting a detected ME as valid may be too *strict*, so it is possible the model would detect a greater number of correct MEs, though changing the threshold to be too *low* may also enable invalid detections to be returned as predictions more often.

For ME recognition, after training on the im2latex-100k dataset for 200 epochs, the model did not perform as well as the FCOS model did. Table 2 shows the results of passing the im2latex-100k test set through the Image2Latex model. BLEU returned approximately 17%, and the percentage of exact matches returned was 0.6%. This means that there were almost no MEs which were returned that was the exact same as the ground truth MEs. Edit distance performed a bit better, only needing about 2 edits on average to change the ME predictions into the ground truth MEs. Having a loss score of 1.6256 is not terrible, but it is not good either. While the ME recognition model was able to learn some things from the training set, it likely was not enough data for the model to learn effectively for this task.

Table 2: Various performance measures for passing im2latex-100k through the Image2Latex model

Measures	Scores	Measures	Scores
BLEU	0.1703	Edit Distance	1.7033
Exact Match	0.0060	Loss	1.6256

6.2 Improved Performance Using the im2latex-230k Dataset

Some of the prediction problems which the Image2Latex model encountered are depicted in Figure 9a. The predicted ME images were rendered from the LaTeX transcript predictions. The first predicted ME is mostly correct, having most of the same symbols, but is missing some of the symbol's present in the ground truth, such as the closing bracket that is either missing or mis-written as other letter symbols. For the second prediction, the first half of the predicted ME was correct, but after the equal sign, the same symbol is repeated over and over again. These problems result from not seeing enough training examples to account for the MEs being passed through the network. There have been cases previously where ME recognition models were not able to process MEs which are longer than the MEs being used in the training set, so this may be a result of the recurrence or attention mechanisms.

Training Image2Latex on the im2latex-230k dataset, however, had very different results as shown in Table 3. The results, as shown in the table, are significantly better than when trained on the im2latex-100k dataset as shown in Table 2. BLEU returned approximately 75% correspondence to the outputs, and the percentage of exact matches returned was 11%. While there are still around 90% MEs generated that are not completely correct, this is significantly better than barely any of them matching, as shown in Table 2. Edit distance using the im2latex-230k dataset performed a significantly better than im2latex-100k, only taking about 0.2 edits on average to change the ME predictions into the ground truth MEs. Having a loss score of 0.3 is actually really good. The significantly enhanced result is due to the fact that the model was set

Table 3: Various performance measures for passing im2latex-230k through the Image2Latex model

Measures	Scores	Measures	Scores
BLEU	0.7454	Edit Distance	0.2189
Exact Match	0.1090	Loss	0.3006

to stop training when the training loss stopped decreasing. It is also possible the model is overfitting to the training data, in which case training for less time would have been better for performance. Despite this, the model does have solid performance for ME recognition, generating MEs which are completely match the ground truth values or are close, as shown in Figure 9b, though there are still a few cases where characters end up in a loop.

6.3 Processing Speed

In terms of proceeding speed, Table 4 shows that the ME detection model with FCOS runs at around 6 FPS during the prediction phase, and Image2Latex achieves around 10 FPS. As stated previously, an FPS of 2.76 is a good real time speed for image text detection, and 5.66 FPS is a favorable real time speed for text recognition. FCOS with Image2LaTeX runs significantly faster than these speeds. As such, the MEDR model meets the design goal of operation at real time speeds, and performs significantly faster in terms of general text detection and recognition.

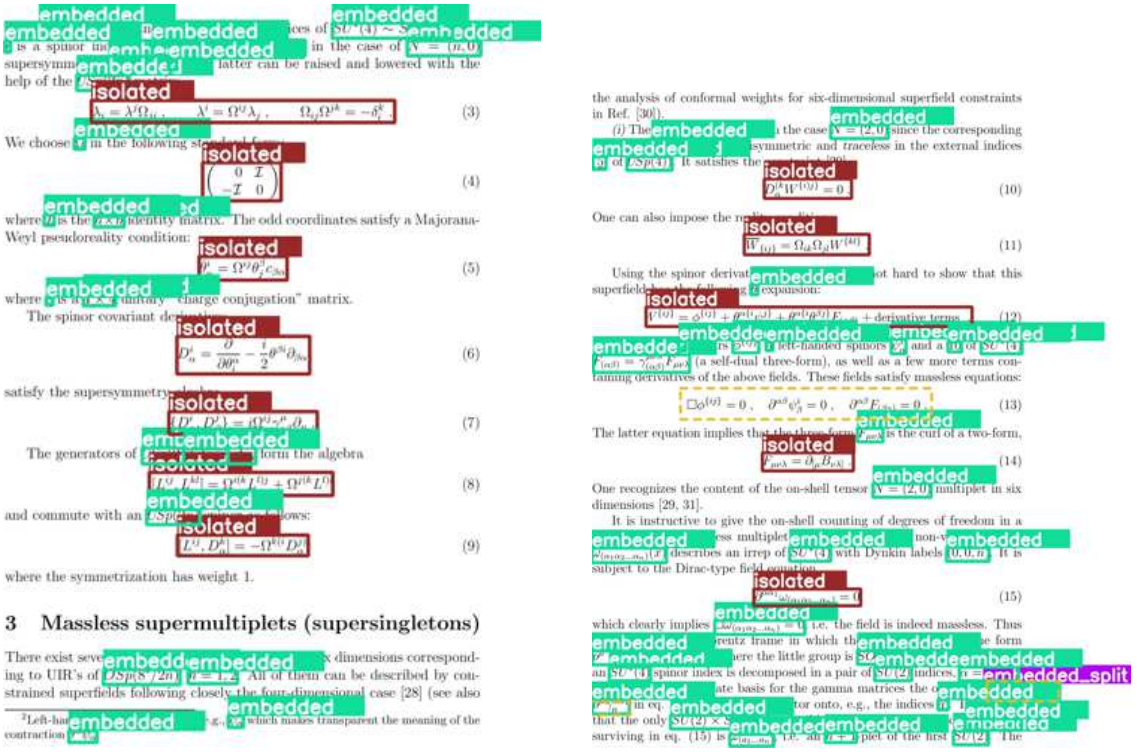
Table 4: Approximate FPS for the ME detection and ME recognition models for the prediction phase

Model	Processing Speed	Model	Processing Speed
FCOS	~6.007 FPS	Image2Latex	~10.097 FPS

7 CONCLUSION

The novel use of FCOS for ME detection turned out producing ME predictions with ideal speed and accuracy. For ME recognition, the Image2Latex model ended up producing admirable results in terms of accuracy and speed, even if it does not outperform current state of the art. Using these two models together yields a system capable of converting images containing MEs and translating the MEs directly to LaTeX markup. Using more data samples for training resulted in better predictions. Since the MEDR model produces good results at real time speeds, especially with the detection portion, this model is valuable in Math IR methods and other platforms.

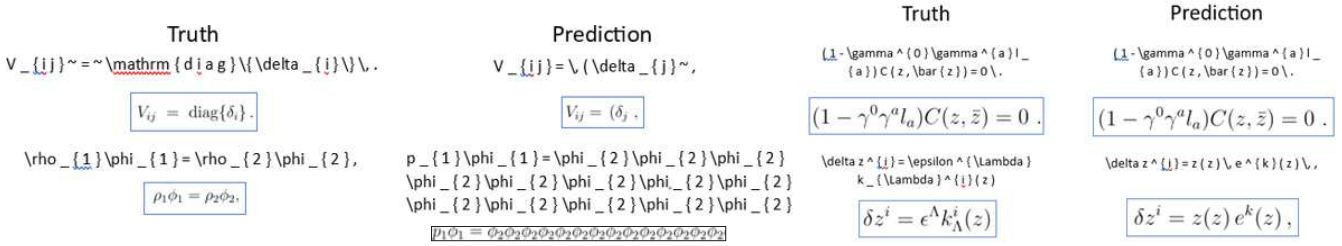
Being able to reliably extract MEs from images would allow more specified math results obtained in regards to math source searches and give greater access to relevant math information for use of both math learners and experts alike. There are a lot of sources of math information that are either in scanned documents, physical documents, or contained in images, and if these sources are unable to be converted into a text-based format, conventional Math IR systems will not be able to utilize these information sources. Neglecting to take into account visual sources of math information with Math IR systems would leave out many viable sources of information that ordinary users and experts can learn from or are directly relevant to what they are working on. As such, being able to convert these sources of information to a viable format would give users access to more information to learn from and provide more documents for Math IR systems to utilize to help provide users with more relevant information to what they are searching for.



(a) Fully-corrected Predictions

(b) Partially-corrected Predictions

Figure 8: Two samples from the results of training on the FCOS model for 100 epochs and generating predicted ME bounding boxes from document images passed through the model. The images in Figure 8a have all of the bounding boxes and ME class labels correctly identified, whereas the prediction in Figure 8b has failed to locate one of the isolated MEs in the center and only located one half of the split ME near the bottom of the page, which are shown in the dashed bounding boxes



(a) ME images and LaTex transcripts from im2latex-100k

(b) ME images and LaTex transcripts from im2latex-230k

Figure 9: Two examples of ground truth ME images and LaTeX transcripts from the im2latex-100k and im2latex-230k datasets, respectively and the prediction errors made on those same ME images

REFERENCES

[1] R. Aggarwal, S. Pandey, A. Tiwari, and G. Harit. 2022. Survey of Mathematical Expression Recognition for Printed and Handwritten Documents. *IETE Technical Review* 39, 6 (2022), 1245–1253.

[2] S. Anand, S. Susan, S. Aggarwal, S. Aggarwal, and R. Singla. 2021. Scene Text Recognition in the Wild with Motion Deblurring Using Deep Networks. In *Proceedings of the 5th International Conference on Computer Vision and Image Processing (CVIP)*. Springer, 93–103.

[3] D. Anitei, J. Sánchez, J. Benedí, and E. Noya. 2023. The IBEM Dataset: A Large Printed Scientific Image Dataset for Indexing and Searching Mathematical Expressions. *Pattern Recognition Letters* 172 (August 2023), 29–36. Issue C.

[4] P. Chandra and S. Rath. 2022. FCOS- Anchor Free Object Detection Explained. https://learnopencv.com/fcos-anchor-free-object-detection-explained/.

[5] W. Chu and F. Liu. 2013. Mathematical Formula Detection in Heterogeneous Document Images. In *Proceedings of 2013 Conference on Technologies and Applications of Artificial Intelligence*. IEEE, 140–145.

[6] Y. Deng, A. Kanervisto, J. Ling, and A. Rush. 2017. Image-to-Markup Generation with Coarse-to-Fine Attention. In *Proceedings of International Conference on Machine Learning (ICML'17)*. PMLR, 980–989.

[7] A. Dey and R. Zanibbi. 2021. ScanSSD-XYc: Faster Detection for Math Formulas. In *Document Analysis and Recognition–ICDAR 2021 Workshops*. Springer, 91–96.

[8] S. Gao and Y.-K. Ng. 2023. Recommending Answers to Math Questions Based on KL-Divergence and Approximate XML Tree Matching. In *Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region (SIGIR-AP)*. ACM, 21–31.

[9] J. Gehrke, P. Ginsparg, and J. Kleinberg. 2003. Overview of the 2003 KDD Cup. *ACM SIGKDD Explorations Newsletter* 5, 2 (2003), 149–151.

[10] G. Genthial and R. Sauvestre. 2016. Image to Latex.

[11] K. Hashmi, A. Pagani, M. Liwicki, D. Stricker, and M. Afzal. 2021. Cascade Network with Deformable Composite Backbone for Formula Detection in Scanned Document Images. *Applied Sciences* 11, 16 (2021), 7610.

[12] K. He, X. Zhang, S. Ren, and J. Sun. 2016. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 770–778.

[13] G. Hyams and D. Malowany. 2020. The Battle of Speed vs. Accuracy: Single-Shot vs Two-Shot Detection Meta-Architecture. https://clear.ml/blog/the-battle-of-speed-accuracy-single-shot-vs-two-shot-detection.

[14] S. Islam, H. Elmekki, A. Elsebai, J. Bentahar, N. Drawel, G. Rjoub, and W. Pedrycz. 2023. A Comprehensive Survey on Applications of Transformers for Deep Learning Tasks. *Expert Systems with Applications* (2023), 122666.

[15] D. Kingma and J. Ba. 2014. Adam: A Method for Stochastic Optimization. *arXiv preprint arXiv:1412.6980* (2014), 1–15.

[16] V. Kukreja and Sakshi. 2022. Machine Learning Models for Mathematical Symbol Recognition: A Stem to Stern Literature Analysis. *Multimedia Tools and Applications* 81, 20 (2022), 28651–28687.

[17] T. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. 2017. Feature Pyramid Networks for Object Detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2117–2125.

[18] J. Long, Q. Hong, and L. Yang. 2023. An Encoder-Decoder Method with Position-Aware for Printed Mathematical Expression Recognition. In *Proceedings of International Conference on Document Analysis and Recognition*. Springer, 167–181.

[19] S. Madisetty, K. Maurya, A. Aizawa, and M. Desarkar. 2021. A Neural Approach for Detecting Inline Mathematical Expressions from Scientific Documents. *Expert Systems* 38, 4 (2021), e12576.

[20] P. Mali, P. Kukkadapu, M. Mahdavi, and R. Zanibbi. 2020. ScanSSD: Scanning Single Shot Detector for Mathematical Formulas in PDF Document Images. *arXiv preprint arXiv:2003.08005* (2020), 1–8.

[21] C. Malon, S. Uchida, and M. Suzuki. 2008. Mathematical Symbol Recognition with Support Vector Machines. *Pattern Recognition Letters* 29, 9 (2008), 1326–1332.

[22] National Center for Educational Statistics. 2022. The NAEP Mathematics Achievement Levels by Grade. https://nces.ed.gov/nationsreportcard/ mathematics/achieve.aspx.

[23] Nation's Report Card. 2019. NAEP Report Card: 2019 NAEP Mathematics Assessment. https://www.nationsreportcard.gov/highlights/mathematics/2019/g12/.

[24] D. Ogwok and E. Ehlers. 2020. Detecting, Contextualizing and Computing Basic Mathematical Equations from Noisy Images using Machine Learning. In *Proceedings of the 2020 3rd International Conference on Computational Intelligence and Intelligent Systems*. 8–14.

[25] W. Ohyama, M. Suzuki, and S. Uchida. 2019. Detecting Mathematical Expressions in Scientific Document Images Using a U-Net Trained on a Diverse Dataset. *IEEE Access* 7 (2019), 144030–144042.

[26] S. Peng, L. Gao, K. Yuan, and Z. Tang. 2021. Image to LaTeX with Graph Neural Network for Mathematical Formula Recognition. In *Proceedings of the 16th International Conference on Document Analysis and Recognition (ICDAR)*. Springer, 648–663.

[27] B. Phong, T. Hoang, and T. Le. 2020. A Hybrid Method for Mathematical Expression Detection in Scientific Document Images. *IEEE Access* 8 (2020), 83663–83684.

[28] Public School Review. 2023. Average Public School Math Proficiency. https://www.publicschoolreview.com/average-math-proficiency-stats/national-data.

[29] Z. Tian, C. Shen, H. Chen, and T. He. 2019. FCOS: Fully Convolutional One-Stage Object Detection. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 9627–9636.

[30] Z. Tian, C. Shen, H. Chen, and T. He. 2020. FCOS: A Simple and Strong Anchor-Free Object Detector. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 44, 4 (2020), 1922–1933.

[31] G. Tong, M. Dong, and Y. Song. 2023. A Real-Time and Effective Text Detection Method for Multi-Scale and Fuzzy Text. *Real-Time Image Processing* 20, 1 (2023), 2–15.

[32] J. Wang, Y. Sun, and S. Wang. 2019. Image to Latex with Densenet Encoder and Joint Attention. *Procedia Computer Science* 147 (2019), 374–380.

[33] Z. Wang and J. Liu. 2021. Translating Math Formula Images to LaTeX Sequences Using Deep Neural Networks with Sequence-Level Training. *Document Analysis and Recognition (IJDAR)* 24, 1-2 (2021), 63–75.

[34] W. Zhong, J. Yang, Y. Xie, and J. Lin. 2022. Evaluating Token-Level and Passage-Level Dense Retrieval Models for Math Information Retrieval. In *Findings of the Association for Computational Linguistics: EMNLP*. 1092–1102.

[35] Y. Zhong, Z. Deng, S. Guo, M. Scott, and W. Huang. 2020. Representation Sharing for Fast Object Detector Search and Beyond. In *Proceedings of the 16th European Conference on Computer Vision (ECCV)*. Springer, 471–487.

[36] M. Zhou, M. Cai, G. Li, and M. Li. 2022. An End-to-End Formula Recognition Method Integrated Attention Mechanism. *Mathematics* 11, 1 (2022), 177.