ETDPC: A Multimodality Framework for Classifying Pages in Electronic Theses and Dissertations

Muntabir Hasan Choudhury[1], Lamia Salsabil[1], William A. Ingram[2], Edward A. Fox[2], Jian Wu[1]

[1]Old Dominion University, Norfolk, VA 23529
[2]Virginia Tech, Blacksburg, VA 24061
mchou001@odu.edu, lsals002@odu.edu, waingram@vt.edu, fox@vt.edu, jwu@cs.odu.edu

Abstract

Electronic theses and dissertations (ETDs) have been proposed, advocated, and generated for more than 25 years. Although ETDs are hosted by commercial or institutional digital library repositories, they are still an understudied type of scholarly big data, partially because they are usually longer than conference and journal papers. Segmenting ETDs will allow researchers to study sectional content. Readers can navigate to particular pages of interest, to discover and explore the content buried in these long documents. Most existing frameworks on document page classification are designed for classifying general documents, and perform poorly on ETDs. In this paper, we propose ETDPC. Its backbone is a two-stream multimodal model with a cross-attention network to classify ETD pages into 13 categories. To overcome the challenge of imbalanced labeled samples, we augmented data for minority categories and employed a hierarchical classifier. ETDPC outperforms the state-of-the-art models in all categories, achieving an F1 of 0.84 – 0.96 for 9 out of 13 categories. We also demonstrated its data efficiency. The code and data can be found on GitHub (https://github.com/lamps-lab/ETDMiner/tree/master/etd_segmentation).

Introduction

Electronic theses and dissertations (ETDs) are scholarly works of students pursuing higher education and successfully meeting the partial requirement of academic degrees. Since 1997, pioneered by Virginia Tech, many universities started requiring degree candidates to submit their ETDs, hosted by the university libraries or a centralized system such as ProQuest (recently acquired by Clarivate). ETDs have distinct features compared with conference papers and journal articles. They are book-length documents (i.e., typically 100 – 400 pages long), and the topics may shift across chapters. In addition, ETDs have unique metadata fields (e.g., advisor, discipline, department) compared with regular scholarly papers. However, most ETD repositories still have limited tools and services for discovering and accessing the content and knowledge in ETDs. One step toward better content and knowledge discovery is to segment the entire ETD by content types so information can be further mined using a customized content reader.

ETDs can be scanned or born-digital (e.g., LaTeX), with complex document structures. There are varied resolutions of scanned images, from typewritten and handwritten texts containing noise. There are few training samples available for classification tasks. To address these challenges, we previously (Ahuja, Devera, and Fox 2022) contributed datasets and methods to segment ETDs using a bottom-up strategy. The method automatically annotated major structural components but still does not perform well in detecting minority classes (e.g., date, degree, equation, algorithms) due to a lack of training samples. Hence we considered multi-modality state-of-the-art (SOTA) frameworks (Xu et al. 2021; Appalaraju et al. 2021) which were fine-tuned on downstream tasks such as document image classification. These frameworks had been evaluated on the RVL-CDIP dataset (Harley, Ufkes, and Derpanis 2015), consisting of scanned document images belonging to 16 classes (e.g., letter, form, email). However, even fine-tuned on ETDs, they do not generalize well to ETDs (e.g., achieving 9% accuracy), and retraining them is non-trivial because of the lack of data. Moreover, text-based classification ignores the information encoded in the layout. Therefore, we take a top-down approach by designing a new framework called ETD Page Classifier (ETDPC – see Figure 1), and apply it to ETD pages, relative to 13 categories (e.g., title-page, chapters, dedication), using multimodality with a cross-attention network. Our contributions are as follows.

- We proposed a two-stream multimodal classification model with cross-attention that uses a vision encoder (e.g., ResNet-50v2) and a text encoder (e.g., BERT with Talking-Heads Attention).
- We proposed a method to augment minority ETD pages leveraging paraphrasing techniques and image-based transformation.
- We built the ETD500 dataset with 92,371-page annotations of ETDs, plus PNGs, text, and bounding boxes.
- We quantitatively demonstrated our system's robustness in classifying ETDs using both original and pseudo-training samples.

Related Work

In layout analysis for general documents, the existing frameworks usually adopt bottom-up approaches to identify docu-

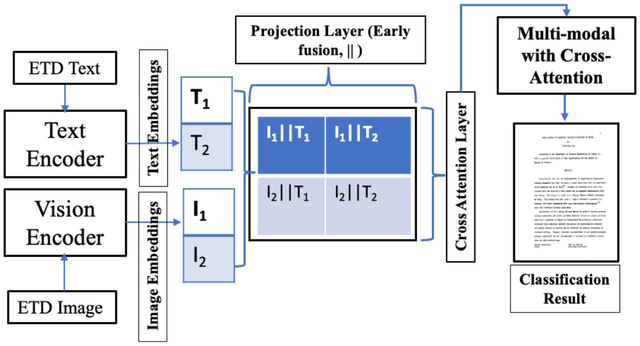


Figure 1: ETDPC – A Multimodality Framework (I – Image, and T – Text).

ment formats (forms, receipts, etc.). Several frameworks, including LayoutLM (Xu et al. 2020), LayoutLMv2 (Xu et al. 2021), and DocFormer (Appalaraju et al. 2021) used multimodality, introduced several pre-training tasks to the model, and then fine-tuned for downstream tasks.

These frameworks differ in their methodological approach and pre-training tasks. The authors in both LayoutLM (Xu et al. 2020) and LayoutLMv2 (Xu et al. 2021) used the joint multimodal, where vision and text features are concatenated into one long sequence and then fed through a transformer self-attention layer to learn the cross-modality interaction between visual and textual information. DocFormer (Appalaraju et al. 2021) instead proposed a discrete multimodal, focusing on sending visual and textual features through individual transformer layers. In each layer, visual and language features separately undergo self-attention with shared bounding box (bbox) information as a spatial feature.

Another multimodal architecture (Dauphinee, Patel, and Rashidi 2019) classifies using VGG16 and bag of words, respectively, for the visual and textual-based architectures. Their fusion technique has individual models running through their own architectures; later the class score vectors were concatenated. Finally, the resulting class score was classified by a meta-classifier using the XGBoost model.

RVL-CDIP (Harley, Ufkes, and Derpanis 2015) is a benchmark dataset of letters, forms, emails, files, resumes, etc., with 400,000 grayscale images in 16 classes. It is split into training, validation, and test sets. Dauphinee, Patel, and Rashidi (2019) trained their model using this dataset and reported 93.03% accuracy. Furthermore, DocFormer-base (Appalaraju et al. 2021) achieved an accuracy of 96.17%, which outperformed the 95.64% accuracy of LayouLMv2 (Xu et al. 2021).

Although these frameworks performed well on general document layout analysis, our experiments show that they do not work well for ETDs. Hence, a framework based on fine-tuning YOLOv7 has been proposed to segment ETDs based on visual features. The framework was evaluated on a new dataset called ETD-OD (Ahuja, Devera, and Fox 2022) that consists of over 25,000 page images from 200 ETDs with manually drawn bboxes around objects (e.g., title, author, paragraph, etc.). However, the lack of training samples led to low performance on minority categories such as date, algorithm, and equation.

To classify pages in legal documents, Wan et al. (2019) proposed a text-based architecture, which used chunk embeddings (i.e., splitting the documents into multiple chunks), which were then used to train Doc2Vec to extract features. It achieved an overall accuracy of 97.97%.

Our work attempts to overcome the limitations of existing frameworks by directly training a multimodal model with cross-attention using a dataset created by manual labeling and augmentation. We employ a hierarchical classification strategy to mitigate the sample imbalance problem.

Methodology

Conceptual Overview

In general, a multimodal workflow involves several unimodal neural networks to encode various input modalities independently. The extracted features are then combined using a fusion module. Finally, the fused features are fed into a classification network to make the prediction. There are three types of fusion (Baltrušaitis, Ahuja, and Morency 2017): a) **Early Fusion** – the features from each modality are combined at the start, and then the full model architecture is applied to the combined features; b) **Late Fusion** – the individual modalities run through their own architectures, and the features from each modality are combined at the end to make the prediction; and c) **Hybrid Fusion** – a combination of early and late fusion. In this paper, we consider a two-stream multimodal model with a cross-attention layer by leveraging the early fusion technique. Figure 1 shows each modality extracting individual embeddings, with a projection layer to unify the dimensions. Then, leveraging the early fusion, we concatenated each projection and combined them with the features from the cross-attention layer. Finally, the full model was applied to the combined features.

Visual Modality

We used ResNetv2 (He et al. 2016b), an improved version of ResNet (He et al. 2016a), with the propagation formulation of the connections between the neural layers. ResNETv2 has new residual units with pre-activation. Instead of putting batch normalization and ReLU after convolution, the authors put them prior to the convolution. ResNetv2 performed better than the original ResNet on the ImageNet-1k (Russakovsky et al. 2015) and CIFAR-10 / 100 datasets[1].

To use the ResNet50v2 model, we resized the ETD images to 224×224 pixels and fed them into the visual encoder. The output feature map was average-pooled to a fixed size with the width (W) and height (H). Later, it is flattened into a visual embedding sequence of length $W \times H$.

Textual Modality

For textual modality, we used BERT with the Talking-Heads Attention and the Gated GELU position-wise feed-forward networks (Shazeer et al. 2020), which is a transformer-based text encoder based on the original BERT (Devlin

[1]https://www.cs.toronto.edu/~kriz/cifar.html

et al. 2019) architecture by replacing the multi-head attention with the talking-heads attention and replacing the ordinary dense layer with a gated linear unit with a GELU activation. The authors compared it against the Text-to-Text Transfer Transformer (T5) (Raffel et al. 2020) model, which used multi-head attention. The results showed that talking-head-based models outperformed the multi-head attention on MNLI (Kim, Kang, and Kwak 2018) by at least 2% on F1.

To generate the embeddings, we first use AWS Textract[2] to extract text from the document images. We then used the pre-trained model of BERT with Talking-Heads Attention (large) (Shazeer et al. 2020) from TensorFlow Hub as a text encoder. Using its pre-processing module, we performed tokenization and extracted the following features: a) **input type IDs**, which identify which sequence a token belongs to when there is more than one sequence; b) **input mask**, which indicates whether a token should be attended to or not; and c) **input word IDs**, which are the indices corresponding to each token in the sentence. These features are then fed through a trainable embedding layer.

Multimodal with Cross-Attention

The vision and text encoders generate embeddings. However, the dimension of both embeddings needs to be unified for further early fusion. One general method that has been adopted to map the dimensions of these embeddings is applying a linear projection. Thus, we introduce a projection layer, where the embeddings from both modalities are combined. Our model takes one 256-D projection layer with 0.8 as a dropout rate (Srivastava et al. 2014). Later, the model fetches each embedding projection and performs an early fusion. To make our model focus on the most important pixels of an image that relate to their corresponding textual parts, we use the "cross-attention" (Chen, Fan, and Panda 2021). Cross-Attention (Wei et al. 2020) combines asymmetrically two separate embedding sequences of two different modalities (i.e., visual and text). Further, we concatenate the early fusion of the projection layer with the attention sequence. We finally passed it through the "softmax" layer for the classification.

Experiments

Dataset: We compiled ETD500 (Hasan Choudhury et al. 2021), which consists of 500 scanned ETDs published between 1945 and 1990. There are 350 STEM and 150 non-STEM majors from 468 doctoral, 27 master's, and 5 bachelor's degrees. The dataset contains a total of 92,371 pages, available in PNG format. For the page-level classification task, we manually annotated all pages of ETD500 using an annotation tool developed by Caragea et al. (2016), into 13 distinct category labels (Table 1). Later, OCR was performed on all pages using AWS Textract, a cloud-based service that detects and extracts texts from scanned documents. Textract converts PDF images into JSON containing text, ID, type (i.e., words or lines), bbox, and confidence score values.

[2]https://aws.amazon.com/textract/

Category	#Pages	#Aug	Pages Description
Chapters	71200	-	content within sections labeled as chapters
Appendices	9891	-	detailed content not included in chapters
References	3385	-	a list of biographical details of in-text citations
ToC	1114	-	the list of chapters and page numbers
TitlePage	911	-	a page containing a title and other metadata
Abstract	777	602	a narrative summary for the whole thesis
ListofFigures	586	651	pages that include a ListofFigures and page numbers
Ack	543	542	acknowledgment
ListofTables	477	584	pages that include ListofTables and page numbers
CV	124	1116	a curriculum vitae
Dedication	77	971	devoting the work to motivating or supportive persons
C. Abstract	66	1518	a chapter summary (occasional)
Other	3220	-	pages that do not fit into any of the other 12 categories
Total	92,371	5,984	-
Subtotal (non-chapters)	21,171	-	-

Table 1: ETD500 – page category labels (Category), total number of labeled pages (#Pages), the number of augmented pages (#Aug) (ToC = TableofContents, C. Abstract = ChapterAbstract).

Data Augmentation: Table 1 gives the number of labeled pages for each category. The labeled data is highly skewed towards Chapters, Appendices, References, and Other pages, making it inappropriate to be used directly for training a machine learning model. Our goal was to perform data augmentation and increase the sample sizes to at least 1,000 for the minority categories (i.e., with fewer samples (Table 1)) to mitigate the imbalance before training. We adopted the following strategies to generate pseudo-ETD training samples for the minority classes.

- We first paraphrase the text extracted by the OCR.
- Second, we convert the text into images.
- Finally, we perform an image-based transformation.

For **paraphrasing** the ETD text, we adopted Google's `PEGASUS` (Zhang et al. 2020)[3], a pre-trained model for text summarization. The base architecture of PEGASUS (Zhang

[3]https://huggingface.co/docs/transformers/model_doc/pegasus

Category	$\mathbf{P_a}$	$\mathbf{R_a}$	$\mathbf{F1_a}$	$\mathbf{P_b}$	$\mathbf{R_b}$	$\mathbf{F1_b}$	$\mathbf{P_c}$	$\mathbf{R_c}$	$\mathbf{F1_c}$
Chapters	0.87	0.98	0.92	–	–	–	–	–	–
Appendices	0.65	0.29	0.40	0.83	0.93	0.88 **(+0.48)**	0.83	0.93	0.88 **(+0.00)**
ReferenceList	0.92	0.92	0.92	0.94	0.94	0.94 **(+0.02)**	0.95	0.94	0.95 **(+0.01)**
TableofContent	0.81	0.75	0.78	0.80	0.86	0.83 **(+0.05)**	0.84	0.83	0.84 **(+0.01)**
TitlePage	0.87	0.88	0.88	0.88	0.94	0.91 **(+0.03)**	0.87	0.94	0.91 **(+0.00)**
Abstract	0.33	0.02	0.03	0.60	0.50	0.54 **(+0.51)**	0.59	0.56	0.74 **(+0.20)**
ListofFigures	0.60	0.57	0.58	0.66	0.64	0.65 **(+0.07)**	0.78	0.67	0.69 **(+0.04)**
Acknowledgment	0.82	0.81	0.82	0.85	0.84	0.84 **(+0.02)**	0.88	0.84	0.93 **(+0.09)**
ListofTables	0.58	0.35	0.44	0.65	0.44	0.52 **(+0.08)**	0.71	0.59	0.62 **(+0.07)**
CurriculumVitae	0.83	0.26	0.40	0.92	0.58	0.71 **(+0.31)**	0.86	1.00	0.94 **(+0.23)**
Dedication	1.00	0.27	0.43	1.00	0.55	0.71 **(+0.28)**	0.98	0.91	0.94 **(+0.23)**
ChapterAbstract	0.00	0.00	0.00	0.00	0.00	0.00 **(+0.00)**	1.0	0.95	0.96 **(+0.96)**
Other	0.61	0.20	0.30	0.75	0.55	0.63 **(+0.33)**	0.78	0.54	0.64 **(+0.01)**
macro F1	0.68	0.48	0.53	0.74	0.64	0.68 **(+0.15)**	0.83	0.81	0.83 **(+0.15)**

Table 2: Performance on ETD samples in the test set – a) performance of one-level classifier (i.e., *Case a*), where ETDPC is trained on ETD500; b) performance of two-level classifier (i.e., *Case b*), training first on chapter vs. non-chapter pages, and next on the remaining categories, including 21,171 manually labeled samples; and c) performance of the two-level classifier (*Case c*), trained on 21,171 manually labeled samples and 5,984 augmented samples. We demonstrated the F1 scores with remarkable improvements for each category.

et al. 2020) is a standard transformer encoder-decoder. In our experiment, we first use `PegasusTokenizer`, which is based on SentencePiece[4], to tokenize the input text while adding several parameters, including padding (i.e., pad to the longest sequence in the batch), max-length (i.e., pad to a maximum length specified with the argument max-length), and truncation (i.e., truncate to a maximum length specified with the argument max-length). To generate the paraphrase, we use the `PegasusForConditionalGeneration` model. After paraphrasing the text, we use a Python module called `textwrap`[5] to wrap the text with a width of 90, which represents the maximum length of wrapped lines. To **convert the text into images**, we change the fonts and size, adjust the textual position on a page, and finally draw the text on an image using Python's `pillow` library[6]. To perform **image-based transformation**, we use a library called `ImgAug`[7]. We adopted the following transformations from ScanBank (Kahu et al. 2021) to generate the final pseudo images.

- **Additive Gaussian noise** – A flatbed scanner works by reflecting the light from paper and creating an image of the paper based on the naturally reflected light. Hence, we use Additive Gaussian Noise to mimic this effect. The parameters of this noise are heuristically chosen using trial and error.
- **Salt-and-pepper noise** – Salt-and-pepper noise is often seen on images caused by sharp and sudden disturbances in the image signal. We heuristically chose 0.9 as the probability of replacing a pixel with noise.
- **Gaussian Blur** – Unlike natural images, digital images must be encoded with a specified resolution resulting in a pre-determined number of bytes and some loss of sharpness. Therefore, we apply Gaussian blurring to smooth the images using a Gaussian Kernel, $\sigma = 0.5$.
- **Linear Contrast** – Although today's scanners are built using modern technology, they cannot capture all colors of a natural object. To incorporate this scanning effect, we add Linear Contrast ($\alpha = 1$).

Fine-tuning Hyper-parameters & Training: We used our in-house high-performance computing cluster (HPC), which runs a deep learning-based container service with various versions of TensorFlow and PyTorch. We use a Tesla V100-SXM2 GPU to train our model and 12-core CPUs to perform other tasks, such as data augmentation. We heuristically fine-tuned the hyper-parameters. Our model with around 460M total parameters is trained on the original and augmented ETD pages. For the hyper-parameters, we choose the "Adam" optimizer with weight decay of 0.004, epsilon of 1e-07, clip value of 2.0, and learning rate of 0.001. In addition, we choose sparse categorical focal loss as the cost function. To avoid overfitting, we use the dropout rate of 0.8. We set up "early stopping" while monitoring the "validation loss" and the "model checkpoint" to save the best weight in each epoch. We finally trained the model with a batch size of 32 and 40 epochs, each taking around 1 hour.

[4]https://github.com/google/sentencepiece
[5]ttps://docs.pyton.org/3/library/textwrap.html
[6]https://pypi.org/project/Pillow/
[7]https://github.com/aleju/imgaug

Evaluation and Results

We split the manually labeled ETD pages into train, validation (25%), and test sets (15%) and consistently use the same test set for all evaluations.

Case a – One-Level Classifier: The ETDPC model, applied to predict all of the 13 categories, achieved an overall accuracy of 84% (Table 2). The classifier achieved lower F1 scores on the categories with relatively small sample sizes than the categories with relatively large sample sizes. For "Chapters" it achieved an F1 score of 0.92.

Case b – Two-Level Classifier: To mitigate the data imbalance problems, we train a two-level classifier. The first level classifies ETD pages into chapters and non-chapters. The second level classifier classifies non-chapter pages into Abstract, Acknowledgement, ListofFigures, ListofTables, Dedication, CV, and C. Abstract, etc., trained on 21,171 manually labeled pages.

Case c – Two-Level Classifier with Augmented Data: To achieve better performance, we used the original training data, consisting of 21,171 manually labeled with 5,984 augmented training samples (Table 1).

Baseline Models: For comparison, we fine-tuned DocFormer (Appalaraju et al. 2021), LayoutLMv2 (Xu et al. 2021), and VGG16 (Simonyan and Zisserman 2015) on the ETD500 dataset. LayoutLMv2 and DocFormer achieved low accuracy, below 30% on the test dataset. Surprisingly, VGG16, a simple model based only on visual features, performed slightly better than LayoutLMv2 and DocFormer (Table 3).

Results: Table 3 shows that ETDPC significantly outperformed the baseline models, LayoutLMv2 and DocFormer, with accuracy increases of 0.76 and 0.57, respectively. We also see a significant performance increase of 0.25 of the accuracy, and 0.44 of the F1 score, compared to VGG16.

We report three types of performance in Table 2: a) performance of the one-level classifier, b) performance of the two-level classifier, and c) performance of the two-level classifier trained with augmented samples. The total inference time to predict the output took 13 minutes. Table 2 indicates the one-level classifier performed well for classifying the "Chapters" category, achieving a 0.92 F1 score. The remaining categories achieve poor performance due to data imbalance. To mitigate data imbalance, we introduced a second-level classifier, which trained on 21,171 non-chapter manually labeled ETD pages. The two-level classifier achieves improved performance, boosting the F1 score, ranging from 0.02 to 0.48 ($\mathbf{F1_b}$ in Table 2), depending on the categories. However, we still observed poor performance for the minority categories (i.e., containing less than 1K training samples (Table 1)), including Abstract, ListofFigures, Acknowledgment, ListofTables, Curriculum Vitae, Dedication, and Chapter Abstract. Thus, we added augmented samples (Table 1) for these minority categories. Our result shows a significant increment in the performance, boosting the F1 scores up to 0.96, depending on the categories.

Models	#param	Accuracy	macro F1
LayoutLMv2-base[1]	200M	0.09	-
DocFormer[2]	174M	0.28	-
VGG16	121M	0.60	0.39
ETDPC **(ours)**	460M	**0.85**	**0.83**

[12] Due to the poor performance, only accuracy is reported.

Table 3: Comparison against baseline models. We report classification accuracy and macro F1 on the test set.

Table 2 shows that ListofFigures and ListofTables achieved relatively low F1 scores. Upon further investigation, we observed 36 samples of ListofTables were misclassified as ListofFigures, whereas 23 samples of ListofFigures were misclassified as ListofTables. After randomly inspecting the testing samples, we found that they are visually very similar except for the page headings. One way to improve the performance of these two categories is by integrating a heuristic-based method to capture lexical patterns, such as "Table" or "Figure", or the appearance of "List of Figures" or "List of Tables" in the headings.

Ablation Study

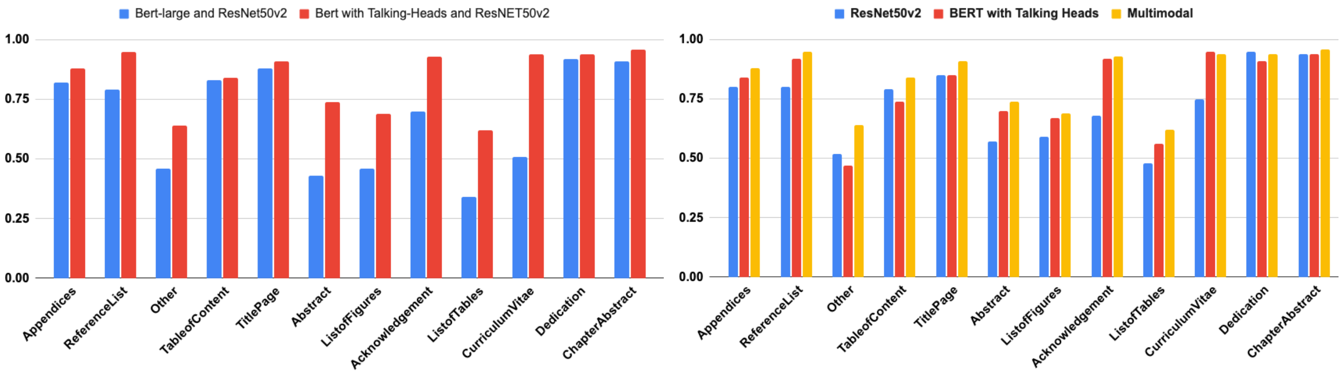
We performed an ablation analysis of our proposed model. We conducted two types of ablation studies: Experiment-1: Changing the text encoder in the multimodal model; Experiment-2: Using individual modalities. We describe each experiment in the following.

Experiment-1: Figure 2 (a) illustrates that when using the BERT with Talking-Heads Attention as a textual modality in our multi-modal model instead of BERT-large, ETDPC improved significantly, boosting the F1 score by 0.02 to 0.43, depending on the categories.

Experiment-2: For this experiment, we used the training samples in Case c. We used each individual modality for this experiment. Training with only visual modality took approximately 30 minutes. Training with only textual modality took approximately 14 hours. Training with our proposed multimodal with cross-attention mechanism took approximately 16 hours. Figure 2 (b) shows that our proposed multimodal model with both modalities outperformed each individual modality, achieving 0.84 to 0.96 F1 scores for 9 out of 13 categories. The model achieves relatively low F1 scores for four categories, including "ListofTables", "Other", "ListofFigures", and "Abstract" with F1 scores ranging from 0.62 to 0.74. However, the performance of the multimodal model for these categories still outperformed each individual modality, improving F1 scores ranging from 0.10 – 0.17 when compared to only visual modality and F1 scores ranging from 0.02 – 0.17 when compared to only textual modality.

Data Efficiency

From Table 2, the F1 scores of several minority categories, such as ListofTables and ListofFigures, are below 0.70. We



(a) Experiment 1

(b) Experiment 2

Figure 2: Ablation Study – a) Experiment 1 illustrates the performance increment when changing the original BERT model to BERT with Talking-Heads. b) Experiment 2 illustrates the performance of using the individual modalities vs. the multimodal model with cross-attention.

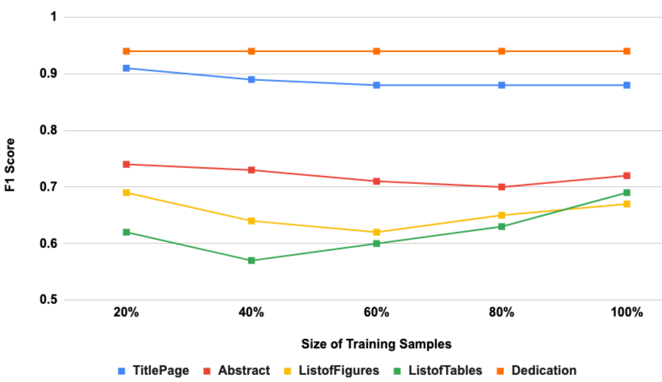


Figure 3: Diminishing returns in performance with increase in the number of training samples.

investigate whether the performance may improve when adding more training samples proportionally to these categories. Using the training dataset in Case c (i.e., 21,171 manually labeled and 5,984 augmented samples, a total of 27,155), we gradually increased the size of the training samples to the following minority categories: Abstract, Dedication, ListofFigures, ListofTables, and TitlePage, by 20% each time.

We show the data efficiency for five categories in Figure 3. The F1 score for the ListofTables category first decreased by ~ 0.05 for the ListofTables and then increased by almost 0.10 when the training size was 100%. The F1 score of the ListofFigures category reached the lowest when the training size is 60% and then increased by about 0.05 when the training size was 100%. The F1 score of the Abstract category decreased consistently until the training size is 80% and then increased marginally by about ~ 0.02. The performance of the Dedication remained constant. The F1 score of the TitlePage category first decreased by ~ 0.02, and then remained constant. Overall, the data efficiency analysis implies that the current datasizes are sufficient for most minority categories except for ListofTables and ListofFigures. In the future, we will add more training data and further investigate data augmentation.

Conclusion & Deployment Path

We developed ETDPC, a framework aiming to classify ETD pages into 13 categories using a two-stream multimodal model with a cross-attention by leveraging vision (e.g., ResNet50v2) and language (e.g., BERT with Talking-Heads) models. The proposed model outperforms SOTA document page classification models. Although our model uses ETD500, consisting of scanned ETDs, for born-digital ETDs, it is straightforward to convert them to images. Our model takes 0.06 seconds to process a single ETD on average, so it is scalable to millions of ETDs. Our model is also customizable. For example, we can easily adopt a multilingual language model to classify non-English ETDs.

Deployment Path The AI method we developed will be deployed following the steps below. First, we will integrate our model into a pipeline that extracts text using an open-source OCR package, such as docTR (Mindee 2021), which produces quality text and is more affordable than commercial OCR APIs, e.g., Amazon Textract. Second, we will deploy our framework on real-world data at the Old Dominion University Digital Commons hosting 3000+ ETDs and conduct an evaluation by manually inspecting the segmented ETDs from different years and academic disciplines. We will then mark and relabel incorrectly classified pages and fine-tune our pre-trained model using newly labeled pages. Finally, we will deploy the refined model on ETDs of the Old Dominion University and the Virginia Tech Libraries. Eventually, we will index ETD sections of different categories and make them available on the library websites.

Acknowledgments

Support was provided by the Institute of Museum and Library Services through grant LG-37-19-0078-198. We also thank Himarsha R. Jayanetti for the data collection.

References

Ahuja, A.; Devera, A.; and Fox, E. A. 2022. Parsing Electronic Theses and Dissertations Using Object Detection. In *Proceedings of the first Workshop on Information Extraction from Scientific Publications*, 121–130. Online: Association for Computational Linguistics.

Appalaraju, S.; Jasani, B.; Kota, B. U.; Xie, Y.; and Manmatha, R. 2021. DocFormer: End-to-End Transformer for Document Understanding. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 993–1003.

Baltrušaitis, T.; Ahuja, C.; and Morency, L.-P. 2017. Multimodal Machine Learning: A Survey and Taxonomy. arXiv:1705.09406.

Caragea, C.; Wu, J.; Gollapalli, S.; and Giles, C. 2016. Document Type Classification in Online Digital Libraries. *Proceedings of the AAAI Conference on Artificial Intelligence*, 30(2): 3997–4002.

Chen, C.-F. R.; Fan, Q.; and Panda, R. 2021. CrossViT: Cross-Attention Multi-Scale Vision Transformer for Image Classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, 357–366.

Dauphinee, T.; Patel, N.; and Rashidi, M. 2019. Modular Multimodal Architecture for Document Classification. *CoRR*, abs/1912.04376.

Devlin, J.; Chang, M.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In Burstein, J.; Doran, C.; and Solorio, T., eds., *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, 4171–4186. Association for Computational Linguistics.

Harley, A. W.; Ufkes, A.; and Derpanis, K. G. 2015. Evaluation of deep convolutional nets for document image classification and retrieval. In *2015 13th International Conference on Document Analysis and Recognition (ICDAR)*, 991–995.

Hasan Choudhury, M.; Jayanetti, H. R.; Wu, J.; Ingram, W. A.; and Fox, E. A. 2021. Automatic Metadata Extraction Incorporating Visual Features from Scanned Electronic Theses and Dissertations. In *2021 ACM/IEEE Joint Conference on Digital Libraries (JCDL)*, 230–233.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016a. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 770–778.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016b. Identity Mappings in Deep Residual Networks. In Leibe, B.; Matas, J.; Sebe, N.; and Welling, M., eds., *Computer Vision – ECCV 2016*, 630–645. Cham: Springer International Publishing.

Kahu, S.; Ingram, W. A.; Fox, E. A.; and Wu, J. 2021. ScanBank: A Benchmark Dataset for Figure Extraction from Scanned Electronic Theses and Dissertations. In *2021 ACM/IEEE Joint Conference on Digital Libraries (JCDL)*, 180–191. Los Alamitos, CA, USA: IEEE Computer Society.

Kim, S.; Kang, I.; and Kwak, N. 2018. Semantic Sentence Matching with Densely-connected Recurrent and Co-attentive Information. arXiv:1805.11360.

Mindee. 2021. docTR: Document Text Recognition. https://github.com/mindee/doctr. Accessed: December 6, 2023.

Raffel, C.; Shazeer, N.; Roberts, A.; Lee, K.; Narang, S.; Matena, M.; Zhou, Y.; Li, W.; and Liu, P. J. 2020. Exploring the Limits of Transfer Learning with a Unified Text-to-Text Transformer. *Journal of Machine Learning Research*, 21(140): 1–67.

Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M.; Berg, A. C.; and Fei-Fei, L. 2015. ImageNet Large Scale Visual Recognition Challenge. arXiv:1409.0575.

Shazeer, N.; Lan, Z.; Cheng, Y.; Ding, N.; and Hou, L. 2020. Talking-Heads Attention. arXiv:2003.02436.

Simonyan, K.; and Zisserman, A. 2015. Very Deep Convolutional Networks for Large-Scale Image Recognition. arXiv:1409.1556.

Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R. 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, 15(56): 1929–1958.

Wan, L.; Papageorgiou, G.; Seddon, M.; and Bernardoni, M. 2019. Long-length Legal Document Classification. *CoRR*, abs/1912.06905.

Wei, X.; Zhang, T.; Li, Y.; Zhang, Y.; and Wu, F. 2020. Multi-Modality Cross Attention Network for Image and Sentence Matching. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 10938–10947.

Xu, Y.; Li, M.; Cui, L.; Huang, S.; Wei, F.; and Zhou, M. 2020. LayoutLM: Pre-Training of Text and Layout for Document Image Understanding. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, 1192–1200. New York, NY, USA: Association for Computing Machinery. ISBN 9781450379984.

Xu, Y.; Xu, Y.; Lv, T.; Cui, L.; Wei, F.; Wang, G.; Lu, Y.; Florencio, D.; Zhang, C.; Che, W.; Zhang, M.; and Zhou, L. 2021. LayoutLMv2: Multi-modal Pre-training for Visually-rich Document Understanding. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, 2579–2591. Online: Association for Computational Linguistics.

Zhang, J.; Zhao, Y.; Saleh, M.; and Liu, P. J. 2020. PEGASUS: Pre-Training with Extracted Gap-Sentences for Abstractive Summarization. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org.